

Precision VME High Voltage Power Supply

Operator's Manual VME Interface



Attention!

-It is not allowed to use the unit if the covers have been removed.

-We decline all responsibility for damages and injuries caused by an improper use of the module. It is highly recommended to read the operators manual before any kind of operation.

Note

The information in this manual is subject to change without notice. We take no responsibility whatsoever for any error in the document. We reserve the right to make changes in the product design without reservation and without notification to the users.

Filename VHS_interface_manual1.odt

Version 3.02

Date: 02/12/2016

Contents:

Inhaltsverzeichnis

1. Operation principel	4
1.1 Remote interface control	4
1.2 Operation modes	5
1.2.1.1 Voltage control(CV)	5
1.2.1.2 Current control(CC)	5
1.2.1.3 Current trip	5
1.3 Function KillEnable	5
2. Control and Status items	5
2.1 Controls	5
2.2 Status and events	6
2.3 Event status and event mask	6
3. Summarizing channel characteristics into groups	8
3.1 Set Group	8
3.2 Status Group	8
3.3 Monitor Group	8
3.4 Timeout Group	8
3.5 Responses on events (Soft-Kill features)	8
4. Autostart	9
5. VME-Interface	10
5.1 Memory space	11
5.2 Details to the memory space	14
5.2.1 Module registers	14
5.2.2 Channel registers	21
5.2.3 Groups	27
5.2.3.1 Fix Groups	27
5.2.3.2 Variable Groups	29
5.2.3.2.1 Set group	29
5.2.3.2.2 Status group	30
5.2.3.2.3 Monitoring group	31
5.2.3.2.4 Timeout group	32
5.3 Events and interrupts	33
5.3.1 Events in channels	33
5.3.2 Events in groups	34
5.3.3 Events in characteristics of the whole module	34
5.3.4 Event status of the module	34
5.4 Special registers	35
5.4.1 Setting of Basis Address	35
5.4.2 Special Control Register	35

1.1 Operation principle

1.1.1 Remote interface control

The Multi-Channel HV modules are controlled via a remote interface. The communication between an application and the module is performed by the transmission of data items. A data item is a unit to be submitted to and/or received from the module. It can represent a specific quantity or a union of single bits. The majority of the data items are standard for all Multi-Channel HV modules and are described in the interface manual in detail. Data items for optional functions are described in the interface options manual.

A general distinction can be made between data items to control individual HV channels and data items to control the HV modules with the sum of all contained channels.

The former group includes the following data items, which exist for every single HV channel:

- items to handle channel status, control and event's
- items to set the voltage or current, bounds, interlock maximum and minimum
- items to read the measured voltage and current
- items to read the nominal voltage and current

The following data items control the properties of the whole HV module. These items exist only once per module:

- items to handle module status, control and event's
- voltage ramp speed (is the same for all HV channels)
- current ramp speed
- restart time after recalling set values
- maximum set voltage
- maximum set current
- ADC samples per second
- digital filter setting

- power supply voltages
- temperature
- maximum voltage
- maximum current

1.1.2 Operation modes

There are three operation modes depending on the HV hardware and the module configuration.

1.1.2.1 Voltage control(CV)

In the mode Voltage control the module works as a constant voltage source. For this mode it is required that the value for current set (Iset) or current trip (Itrip) is greater than the resulting output current.

1.1.2.2 Current control(CC)

In the mode Current control the module works as a constant current source. For this mode it is required that the HV channel has implemented a current control and that the voltage set value Vset is greater than the resulting output voltage.

1.1.2.3 Current trip

This is a special case of the voltage regulation. The module usually provides a constant output voltage, where the value of the parameter Itrip defines a current limit. If this value is reached or exceeded (e.g. by arcs), in this mode the channel will be switched off immediately.

1.1.3 Function KillEnable

KillEnable is a global control signal that defines the behaviour of the module if a given voltage (Vmax) or current limit (Imax/Iset/Itrip) is exceeded.

If **KillEnable is active** the violation of one of the limits will trigger a Kill-signal in the respective channel. This signal will switch off the channel immediately without ramp.

If **KillEnable is inactive** and one of the limits Imax/Iset or Itrip is exceeded the following will happen:

HV hardware with current control - switch the channel from voltage control into current control.

HV hardware without current control – a trip in the channel hardware will switch off the high voltage generation. Then the module automatically starts to restore the HV via a voltage ramp to the set voltage. If the HV is held during the trip, e.g. by an external capacity load, the recovery of the HV starts from the voltage at the output. The auto-recovery of the voltage is performed only once in a time span of 10 minutes. If the channel trips a second time within the 10 minutes the HV will be switched off.

1.2 Control and Status items

1.2.1 Controls

Control items encapsulate a number of bits which allow to switch On or Off specific functions. There is a control item for the module (**ModulControl**) and one for each channel (**ChannelControl**). Control bits that are used to switch a function permanently are named “set...” (e.g. “setON” to switch

a channel On of Off). Bits that initiate the execution of a task just once are named “do...” (e.g. “doClear” to clear all events).

1.2.2 Status and events

Status items contain a register that encapsulates bits that indicate the current status of the module or channel. Status bits are named starting with “is...”. The status always displays only present conditions, if a condition has changed corresponding status bits will be updated.

Unlike the status, event items record previous conditions (e.g. exceeded limits, trips etc.). If an event is registered the corresponding event bit is set permanently to “1” and will keep the information until explicitly reset. Event bits are named starting with “E...”.

status Summary of actual condition of module, channel or group
 event Event, that characterizes a former or actual special condition of module, channel or group

1.2.3 Event status and event mask

To avoid the need for checking all event sources permanently for incoming events, the module provides a hierarchical chain for the combination of the events to a single status bit. The structure for the event processing allows a combination of events coming from the module status, the status of the channels and the group status. For each event status item a corresponding event mask item is provided. The event mask defines which event status bits contribute to the combined event status.

Event status Events that have been registered so far
 Event mask Filter to define which individual events contribute to the summarized event

Between event status items and the corresponding mask is a bit by bit correspondence. The bits in the mask are named starting with “ME...”. If the mask bit is set, the occurring of the respective event will activate the combined event. In turn these sum events are collected in an event status register and connected with an event mask register at this higher level.



The EventStatus and EventMask is always checked before the HV is switched on. If an event bit in the EventStatus is active and the corresponding bit in the EventMask is set, the HV generation cannot be activated. The EventStatus bits must be reset first by writing “1” on the corresponding bit positions.

Individual events in the channel event status are starting point of the event combination logic. First each event status bit for the channel is combined with the corresponding bit in the event mask using a logical AND. Then an event status bit for the channel is generated by combining all resulting bits with a logical OR. The full logical operation is given by

```
EventChannelStatus[n] = (Channel[n].EventVoltageLimit AND Channel[n].MaskEventVoltageLimit) OR
(Channel[n].EventCurrentLimit AND Channel[n].MaskEventCurrentLimit) OR
(Channel[n].EventCurrentTrip AND Channel[n].MaskEventCurrentTrip) OR
(Channel[n].EventExtInhibit AND Channel[n].MaskEventExtInhibit) OR
(Channel[n].EventVoltageBounds AND Channel[n].MaskEventVoltageBounds) OR
(Channel[n].EventCurrentBounds AND Channel[n].MaskEventCurrentBounds) OR
(Channel[n].EventControlledVoltage AND Channel[n].MaskEventControlledVoltage) OR
(Channel[n].EventControlledCurrent AND Channel[n].MaskEventControlledCurrent) OR
```

$$\begin{aligned} & (\text{Channel}[n].\text{EventEmergency AND Channel}[n].\text{MaskEventEmergency}) \text{ OR} \\ & (\text{Channel}[n].\text{EventEndOfRamp AND Channel}[n].\text{MaskEventEndOfRamp}) \text{ OR} \\ & (\text{Channel}[n].\text{EventOnToOff AND Channel}[n].\text{MaskEventOnToOff}) \text{ OR} \\ & (\text{Channel}[n].\text{EventInputError AND Channel}[n].\text{MaskEventInputError}) \end{aligned}$$

The result of the first step for all channels is stored in the register EventChannelStatus. In the next step all bits of the EventChannelStatus are combined to a single status bit, using the corresponding mask (EventChannelMask). The logical operation is given by

$$\begin{aligned} \text{EventChannelActive} = & (\text{EventChannelStatus}[0] \text{ AND EventChannelMask}[0]) \text{ OR} \\ & (\text{EventChannelStatus}[1] \text{ AND EventChannelMask}[1]) \text{ OR} \\ & \dots \\ & (\text{EventChannelStatus}[n] \text{ AND EventChannelMask}[n]) \end{aligned}$$

A second branch in the event processing logic treats events generated by the status of the module. The following scheme applies to these module events:

$$\begin{aligned} \text{EventModuleActive} = & (\text{EventTemperatureNotGood AND MaskEventTemperatureNotGood}) \text{ OR} \\ & (\text{EventSupplyNotGood AND MaskEventSupplyNotGood}) \text{ OR} \\ & (\text{EventSafetyLoopNotGood AND MaskEventSafetyLoopNotGood}) \end{aligned}$$

A third branch combines events generated by groups (monitor group, timeout group, see chapter 3) Group events are stored in the status register EventGroupStatus. The mask EventGroupMask is used to generate the combined bit EventGroupActive with the following operation:

$$\begin{aligned} \text{EventGroupActive} = & (\text{EventGroupStatus}[0] \text{ AND EventGroupMask}[0]) \text{ OR} \\ & (\text{EventGroupStatus}[1] \text{ AND EventGroupMask}[1]) \text{ OR} \\ & \dots \\ & (\text{EventGroupStatus}[32] \text{ AND EventGroupMask}[32]) \end{aligned}$$

Finally the three branches are combined to the bit IsEventActive in the register ModuleStatus:

$$\text{IsEventActive} = \text{EventChannelActive OR EventModuleActive OR EventGroupActive}$$

1.3 Summarizing channel characteristics into groups

The module provides a highly flexible group functionality. A group is a combination of all or a selection of channels with the ability to control or monitor a specified quantity or characteristic of all included channels. There are two classes of groups "Fix Groups" and "Variable Groups". The former are predefined groups that allow to set single specification values in all channels.

The latter are configurable groups that allow to customize the logical structure of the module to the logical structure of the application. They allow an arbitrary assignment of channels and provide a wide range of functionality, structured in four predefined group types. Up to 32 Variable Groups can be defined. The predefined group types are:

1.3.1 Set Group

- sets a specified channel characteristic in all selected channels
- no event generation

1.3.2 Status Group

- represents the status (condition) of a channel characteristic for all channels
- no event generation

1.3.3 Monitor Group

- monitors the condition of a channel characteristic for selected channels
- event generation when the condition changes
- configurable response (e.g. switch off)

1.3.4 Timeout Group

- monitors the current trip in selected channels
- to employ this group the signal KillEnable must be turned off
- Event generation only after expiry of a predefined time within which the trip condition must be active
- configurable response (e.g. switch off)

1.3.5 Responses on events (Soft-Kill features)

Event generating groups can be configured to perform one out of four predefined responses if the event has been generated:

- shut down of the whole module without ramp
 - high voltage in all channels of the module is switched off
- switch off all channels that are members of the group without ramp
 - high voltage in all channels of the group is switched off
- switch off all channels that are members of the group with ramp
 - high voltage in all channels of the group is ramped down
- no response
 - no change

1.4 Autostart

The Autostart functionality allows a recall/reload of stored values to the corresponding set values. A delayed switch-on of the high voltage can be configured. The delay time is configured using the item `RestartTimeAfterRecallSetValues`.

The following set values can be stored permanently for the channels:

- ChannelControl
- ChannelEventMask
- VoltageSet
- CurrentSet/CurrentTrip
- VoltageBounds/VoltageIrkMaxSet
- CurrentBounds/CurrentIrkMaxSet
- VoltageIrkMinSet
- CurrentIrkMinSet
- VoltageMaxSet
- CurrentMaxSet

the module:

- ModuleControl
- ModuleEventMask
- ModuleEventChannelMask
- ModuleEventGroupMask
- VoltageRampSpeed
- CurrentRampSpeed
- RestartTimeAfterRecallSetValues
- ADCSamplesPerSecond
- DigitalFilter

Once a configuration of set values has been stored permanently, it can be “recalled/reloaded” anytime. For this purpose control and status bits are available in the `ModulControl`, `ModulStatus` and `ModulEventStatus`. The detailed explanation is given in chapter [4.2.1. Module registers](#), `ModulStatus`, `ModulControl`, `ModuleEventStatus` and `RestartTimeAfterRecallSetValues`.

2. VME-Interface

Access Mode:

Short supervisory access (AM=0x2D)
 Short non privileged access (AM=0x29)

Command execution time:

The command execution times are 1 µs typically.

Memory space:

The control of the module is working via a data exchange in the RAM memory of the VME module. This is working with a space of 1024 bytes.

The description of RAM addressing in this document is done in a byte addressing type. The RAM memory space begins at the base address (BA). This is a 16bit address, where the 10 LSB bits are 0. The 6 MSB bits can be set by the customer to insert the module's RAM into the VME space.

in bytes:

binary: BA = bbbbb00 00000000 (with b={0|1})
 hexadezimal: BA = xy00 (with x={0..F}, y={0,4,8,C}).

The MSB byte of the base address is stored in the non-volatile memory. It can be changed with help of a special write command (see special commands).

The factory setting is BA=0x4000 in Bytes

Partition of the memory (given in bytes):

BA+0x0000 .. BA+0x003f :	module data . 64 bytes	64 Bytes
BA+0x0060 .. BA+0x029f :	12 channel data blocks ea. 48 bytes	576 Bytes
BA+0x02a0 .. BA+0x02af :	2 fixed groups ea. 4 bytes	8 Bytes
BA+0x02c0 .. BA+0x033f :	32 variable groups (set, status, monitoring or timeout groups) data block ea. 4 bytes	128 Bytes
BA+0x03a0 .. BA+0x03ff :	control registers for special use	

The data exchange is working in standard formats 'Unsigned Long' (uint32), 'Unsigned Integer' (uint16) and Float, single precision (float). The access is with 16Bit words. There is no hardware check regarding non-valid data conditions (e.g. between writing of the first and second words of a floating value), also if an access conflict occurs in the dual ported RAM. Therefore it is necessary to find reasonable measures to save the correct data transfer. (e.g. read or write repetition).

Data formats:

The data format on the VME bus is Big Endian format, i.e. highest byte on lowest address. In contrast, Intel computers store the value byte-wise reversed in memory (Little Endian).

The following formats are used:

uint8 unsigned character (8 bit)
uint16 unsigned word (16 bit)
uint32 unsigned integer (32 bit)
float floating point according to IEEE-754 single precision format

To convert floating-point values to their hexadecimal representation and vice versa, the online calculator <http://babbage.cs.qc.edu/courses/cs341/IEEE-754/> can be used.

Example Channel 0 Vset = 1000 V:

Data-Bytes on the VME bus: 0x44 0x7A 0x00 0x00

Data-Bytes in little endian memory: 0x00 0x00 0x7A 0x44

Usually, the byte-swapping within a 16 bit word is done by the VME driver. As all VME accesses are 16 bit wide, only the data words have to be swapped, not the bytes.

- Writing word-wise:
 1. Write the high memory word to the lower VME address:
write 0x447A to address 0x4068
 2. Write the low memory word to the higher VME address:
write 0x0000 to address 0x406A
- Reading word-wise:
 1. Read the lower VME address and store the value in the high memory word
read from 0x406A, value = 0x0000
 2. Read the higher VME address and store the value in the low memory word
read from 0x4068, value = 0x447A

2.1 Memory space

Module data

Offset Bytes (rel. to BA)	Name	Data type	Access
0x0000	ModuleStatus	uint16	r
0x0002	ModuleControl	uint16	r/w
0x0004	ModuleEventStatus	uint16	r/w
0x0006	ModuleEventMask	uint16	r/w
0x0008	ModulEventChannelStatus	uint16	r/w
0x000A	ModulEventChannelMask	uint16	r/w
0x000C	ModuleEventGroupStatus	uint32	r/w
0x0010	ModuleEventGroupMask	uint32	r/w
0x0014	VoltageRampSpeed	float	r/w
0x0018	CurrentRampSpeed	float	r/w
0x001C	VoltageMax	float	r
0x0020	CurrentMax	float	r
0x0024	SupplyP5	float	r
0x0028	SupplyP12	float	r
0x002C	SupplyN12	float	r
0x0030	Temperature	float	r
0x0034	SerialNumber	uint32	r
0x0038	FirmwareRelease	uint8[4]	r
0x003C	PlacedChannels	uint16	r
0x003E	DeviceClass	uint16	r
0x0040	ModuleInterlockOut Registers		
...0x004A	(look at chapter Fehler: Referenz nicht gefunden)		
0x0050	RestartTimeAfterRecallSetValues	uint16	r/w
0x0058	ADCSamplesPerSecond	uint16	r/w
0x005a	DigitalFilter	uint16	r/w
0x005C	VendorId : const 'i','s','e','g' = 0x69736567	uint8[4]	r

Channels

Offset Bytes (rel. to BA)	Name	
0x0060	ChAddr[0]	begin of channel 0
0x0090	ChAddr[1]	begin of channel 1
0x00C0	ChAddr[2]	begin of channel 2
0x00F0	ChAddr[3]	begin of channel 3
0x0120	ChAddr[4] ¹	begin of channel 4
0x0150	ChAddr[5] ¹	begin of channel 5
0x0180	ChAddr[6] ¹	begin of channel 6
0x01B0	ChAddr[7] ¹	begin of channel 7
0x01E0	ChAddr[8] ¹	begin of channel 8
0x0210	ChAddr[9] ¹	begin of channel 9
0x0240	ChAddr[10] ¹	begin of channel 10
0x0270	ChAddr[11] ¹	begin of channel 11

¹ only in module type VHS Cxx

Channel data

Offset Bytes (rel. to ChAddr)	Name	Data type	Access
0	ChannelStatus	uint16	r
2	ChannelControl	uint16	r/w
4	ChannelEventStatus	uint16	r/w
6	ChannelEventMask	uint16	r/w
8	VoltageSet	float	r/w
12	CurrentSet / CurrentTrip ²	float	r/w
16	VoltageMeasure	float	r
20	CurrentMeasure	float	r
24	VoltageBounds / VoltageIkkMaxSet ³	float	r/w
28	CurrentBounds / CurrentIkkMaxSet ⁴	float	r/w
32	VoltageNominal / VoltageMaxSet ⁵	float	r/(w)
36	CurrentNominal / CurrentMaxSet ⁵	float	r/(w)
40	VoltageIkkMinSet	float	r/w
44	CurrentIkkMinSet	float	r/w

2 when KiiEnable=active

3 the addressed item are multiplexed by the ModuleControl bit setAVBND(0) – VoltageBounds, setAVBND(1) - VoltageIkkMaxSet

4 the addressed item are multiplexed by the ModuleControl bit setACBND(0) – CurrentBounds, setACBND(1) – CurrentIkkMaxSet

5 can be written in mode ModuleStatus IsStop = 1

Group data

Fixed Groups

Offset Bytes (rel. to BA)	Name	Data type	Access
0x02A0	SetVoltageAllChannels	float	r/w
0x02A4	SetCurrentAllChannels	float	r/w
0x02A8	SetVoltageBoundsAllChannels	float	r/w
0x02AC	SetCurrentBoundsAllChannels	float	r/w
0x02B0	SetEmergencyAllChannels	uint32	r/w
0x02B4	SetOnOffAllChannels	uint32	r/w
0x02B8	SetVoltageIkkMinSetAllChannels	float	r/w
0x02BA	SetCurrentIkkMinSetAllChannels	float	r/w

Variable Groups

Offset Bytes (rel. to BA)	Name	
0x02C0	GrAddr[0]	begin of group 0
0x02C4	GrAddr[1]	begin of group 1
0x02C8	GrAddr[2]	begin of group 2
0x02CC	GrAddr[3]	begin of group 3
0x02D0	GrAddr[4]	begin of group 4
0x02D4	GrAddr[5]	begin of group 5
0x02D8	GrAddr[6]	begin of group 6
0x02DC	GrAddr[7]	begin of group 7
0x02E0	GrAddr[8]	begin of group 8
...
0x033C	GrAddr[31]	begin of group 31

User defined nominal values (ModuleStatus IsStop(0))

Offset Bytes (rel. to BA)	Name	Data type	Access
0x0340	VoltageMaxSet channel 0	float	r
0x0344	CurrentMaxSet channel 0	float	r
0x0348	VoltageMaxSet channel 1	float	r
0x034C	CurrentMaxSet channel 1	float	r
0x0398	VoltageMaxSet channel 11	float	r
0x039C	CurrentMaxSet channel 11	float	r

If the module is not in mode STOP the values of VoltageMaxSet and CurrentMaxSet appear.

Hardware defined nominal values (ModuleControl SetStop(1), ModuleStatus IsStop(1))

Offset Bytes (rel. to BA)	Name	Data type	Access
0x0340	VoltageNominal channel 0	float	r
0x0344	CurrentNominal channel 0	float	r
0x0348	VoltageNominal channel 1	float	r
0x034C	CurrentNominal channel 1	float	r
0x0398	VoltageNominal channel 11	float	r
0x039C	CurrentNominal channel 11	float	r

If the module is in mode STOP the values of VoltageNominal and CurrentNominal appear.

Special Registers

Offset Bytes (rel. to BA)	Name	Data type	Access
0x03A0	NewBaseAddress	uint16	r/w
0x03A2	NewBaseAddressXor	uint16	r/w
0x03A4	OldBaseAddress	uint16	R
0x03A6	NewBaseAddressAccepted	uint16	R
0x03B0	SpecialControlStatus	uint16	R
0x03B2	SpecialControlCommand	uint16	r/w

2.2 Details to the memory space

2.2.1 Module registers

ModuleStatus

Offset Bytes (rel. to BA)	Name	Data type	Access
0x0000	ModuleStatus	uint16	r

Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
isKILE	isTMPG	isSPLYG	isMODG	isEVNTA	isSFLPG	isnRMP	isnSERR	isCCMPL	isSPMD	isIERR	ndSRVC	res	isSTOP	isILKO	isADJ

isKILE	IsKillEnable	Kill enable (1); Kill disable (0)
isTMPG	IsTemperatureGood	Module temperature good
isSPLYG	IsSupplyGood	Power supply good
isMODG	IsModuleGood	Module in state good
isEVNTA	IsEventActive	Any event is active and mask is set
isSFLPG	IsSafetyLoopGood	Safety loop closed
isnRMP	IsNoRamp	All channels stable, no ramp active .
isnSERR	IsNoSumError	Module without failure
isCCMPL	IsCommandComplete	All commands complete
isSPMD	IsSpecialMode	Module is in SpecialMode
isIERR	IsInputError	Input error in connection with a module access
ndSRVC	IsServiceNeeded	Module shows that a factory service is needed
isSTOP	IsStop	Modules is in state STOP, all high voltages are off
isILKO	IsInterlockOutput	InterlockOutput is active
isADJ	IsAdjustment	Activation of fine adjustment
Res	Reserved	

The status bits as there are IsTemperatureGood, IsSupplyGood, IsModuleGood, IsEventActive, IsSafetyLoopGood, IsNoRamp, IsNoSumError and IsServiceNeeded indicate the single status for the complete module.

The status bit IsCommandComplete indicates that all VME commands given to the module have been executed.

The condition bit IsEventActive is set, if at least one event is active in the channel, groups or module area and the corresponding masking bits are set.

The signal IsStop(1) shows that module is in mode STOP. In mode STOP it is possible to change the user defined nominal values VoltageMaxSet, CurrentMaxSet to a value lower or equal to the nominal values of hardware - VoltageNominal, CurrentNominal. When a user defined nominal value has been set, the module firmware will operate with it instead of the nominal value of hardware. In addition the Autostart function can be configured in this mode.

The signal IsAdjustment(1) shows that the high voltage is locked under fine adjustment. That means after a switch ON the high voltage will ramp to the value of set voltage followed by steps of adjustment until the measured value fits the set value and only bit wise correction of temperature drifts are necessary.

ModuleControl

Offset Bytes (rel. to BA)	Name	Data type	Access
0x0002	ModuleControl	uint16	r/w

Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
doSVSV	setKILE	res	setADJ	res	ILVL2	ILVL1	ILVL0	res	doCLEAR	res	res	setAON	setSTOP	doRCSV	setSPMD
0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0

doSVSV	DoSaveSetValues	DoSaveSetValues(1); no action (0)
setKILE	SetKillEnable	Kill enable SetKillEnable(1); Kill disable SetKillEnable(0)
setADJ	SetAdjustment	Activation of fine adjustment
ILVL[2..0]	IntLevel[2..0]	Code for VME-Interrupt-Level (1 to 7); Level 0 means: no VME Interrupt
doCLEAR	DoClear	Clears Kill (hardware) signals and all event signals of module and channels
setAON	SetActionOn	SetActionOn(1) activate a time delayed switch ON of the high voltages after a recall of the stored values when ChannelControl SetON(1)
setSTOP	SetStop	SetStop(1);
doRCSV	DoRecallSetValues	DoRecallSetValues(1); no action (0)
setSPMD	SetSpecialMode	Set into SpecialMode, for special tasks only Attention: Return from SpecialMode only with SpecialControlCommands e.g. EndSpecial
res	Reserved	

The signal SetAdjustment is used to enable an adjustment of the HV precisely in case of temperature drifts.

The signal SetKillEnable controls the reaction of the channels to extraordinary events, e.g. overcurrent. The signal is set module-wide, while the reaction (e.g. turn off the high voltage) is done in the correlating channel.

The signals SetStop, SetActionOn, DoSaveSetValues and DoRecallSetValues will be used to realize the Autostart functionality which allows a store and recall/reload of stored values. A time delay of switch ON high voltages is configurable.

SetStop(1) The high voltage of all channels will be decreased with the VoltageRamp and switched OFF. The module firmware goes in the state IsStop(1), ModuleStatus when all channels are OFF.

doSaveSetValues(1) –when setStop(1) only
will start a task to store the set values permanently, listed in chapter 3.2.5 Autostart, when the module is in state IsStop(1). When the task is finished the bit is reset to zero.

SetStop(0) A software restart will be executed whereas the stored set values are reloaded from flash memory. Depending from the bit SetActionOn a delayed switch ON of high voltage will realized.

DoRecallSetValues(1)
execute a recall of the stored set values. The high voltages will be switched on after the value RestartTimeAfterRecallSetValues when a delayed switch ON has been configured SetActionOn(1).

SetActionOn(1)
A recall of the stored values with time delayed switch ON of the high voltages will cause the bit set ERSTA of ModuleEventStatus.

Short overview about reaction in dependency of KillEnable:

	Vout >= Voltage limit	Iout >= Current limit	Iout >= Iset
SetKillEnable=1 (ON)	Kill =1; Vout -> 0; Vset=0;	Kill=1; Vout -> 0; Vset=0;	Vout -> 0, Vset=0
SetKillEnable=0 (OFF)	Vout = Voltage limit	Iout = Current limit	Iout = Iset

The signal SetAdjustment switches on the fine justification of the high voltage, around temperature drifts compensate by the DAC. It is activated after reset.

ModuleEventStatus

Offset Bytes (rel. to BA)	Name	Data type	Access
0x0004	ModuleEventStatus	uint16	r/w

Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
res	ETMPngd	ESPLYngd	res	res	ESFLPngd	res	res	res	res	EIERR	ESRVC	res	res	ERSTA	res

ETMPngd	EventTemperatureNotGood	Event: Temperature is above 55°C
ESPLYngd	EventSupplyNotGood	Event: at least one of the supplies is not good
ESFLPngd	EventSafetyLoopNotGood	Event: Safety loop is open
EIERR	EventInputError	Event: Input error in connection with a module access
ESRVC	EventServiceNeeded	Event: Module needs a factory service
ERSTA	EventRestart	Event: Restart of HV after the RestartTimerAfterRecallSetValues
res	Reserved	

These bits are set when the condition occurs. They can be reset individually by writing ones. If the triggering event is still active, a reset isn't possible.

ModuleEventMask

Offset Bytes (rel. to BA)	Name	Data type	Access
0x0006	ModuleEventMask	uint16	r/w

Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
res	METMPngd	MESPLYngd	res	res	MESFLPngd	res	res	res	res	MEIERR	res	res	res	MERSTA	res

METMPngd	MaskEventTemperatureNotGood	MEventMask: Temperature is above 55°C
MESPLYngd	MaskEventSupplyNotGood	MEventMask: at least one of the supplies is not good
MESFLPngd	MaskEventSafetyLoopNotGood	MEventMask: Safety loop (SL) is open
MEIERR	MaskEventInputError	MEventMask: Input error in connection with a module access
MERSTA	MaskEventRestart	MEventMask: Restart of HV after the RestartTimeAfterRecallSetValues
res	Reserved	

This register decides whether a pending event leads to the sum event flag of the module or not. If the a bit of the mask is set and the corresponding event in the ModuleEventStatus is active the bit IsEventActive in the ModuleStatus register is set.

ModuleEventChannelStatus

Offset Bytes (rel. to BA)	Name	Data type	Access
0x0008	ModuleEventChannelStatus	uint16	r/w

Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
res	res	res	res	CH11	CH10	CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0

The n-th bit of the register is set, if an event is active in the n-th channel and the associated bit in the EventMask register of the n-th channel is set too.

$$CHn = EventStatus[n] \& EventMask[n]$$

The bits can be reset individually by writing ones. If the triggering event is still active, a reset isn't possible.

ModuleEventChannelMask

Offset Bytes (rel. to BA)	Name	Data type	Access
0x000A	ModuleEventChannelMask	uint16	r/w

Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
res	res	res	res	CH11	CH10	CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0

This register decides whether a pending event leads to the sum event flag of the module or not. If the n-th bit of the mask is set and the n-th channel has an active event in the ModuleEventChannelStatus the bit isEventActive in the ModuleStatus register is set.

ModuleEventGroupStatus

Offset Bytes (rel. to BA)	Name	Data type	Access
0x000C	ModuleEventGroupStatus	uint32	r/w

Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
GR31	GR30	GR29	GR28	GR27	GR26	GR25	GR24	GR23	GR22	GR21	GR20	GR19	GR18	GR17	GR16

Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
GR15	GR14	GR13	GR12	GR11	GR10	GR9	GR8	GR7	GR6	GR5	GR4	GR3	GR2	GR1	GR0

The n-th bit of this double word register is set, if an event is active in the n-th group.

ModuleEventGroupMask

Offset Bytes (rel. to BA)	Name	Data type	Access
0x0010	ModuleEventGroupMask	uint32	r/w

Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
GR31	GR30	GR29	GR28	GR27	GR26	GR25	GR24	GR23	GR22	GR21	GR20	GR19	GR18	GR17	GR16

Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
GR15	GR14	GR13	GR12	GR11	GR10	GR9	GR8	GR7	GR6	GR5	GR4	GR3	GR2	GR1	GR0

This register decides whether a pending event leads to the sum event flag of the module or not. If the n-th bit of the mask is set and the n-th group has an active event in the ModuleEventGroupStatus the bit isEventActive in the ModuleStatus register is set.

VoltageRampSpeed

Offset Bytes (rel. to BA)	Name	Data type	Access
0x0014	VoltageRampSpeed	float	r/w

The speed of the voltage ramp in percent of the nominal voltage of the channel. The upper limit is 20%. The lower limit is equivalent to 1mV/s.

CurrentRampSpeed (option)

Offset Bytes (rel. to BA)	Name	Data type	Access
0x0018	CurrentRampSpeed	float	r/w

not realized in VHS x0x

VoltageMax

Offset Bytes (rel. to BA)	Name	Data type	Access
0x001C	VoltageMax	float	r

VoltageMax is the actual value of the trim potentiometer of the front panel, given in per cent. In conjunction with the nominal voltage VoltageNominal of a channel one can calculate the actual maximal output voltage of the channel.

$$\text{VoltageLimit} = \text{VoltageNomial} * \text{VoltageMax}$$

This voltage value VoltageLimit is the reference for setting the status bit IsVoltageLimitExceeded.

CurrentMax

Offset Bytes (rel. to BA)	Name	Data type	Access
0x0020	CurrentMax	float	r

CurrentMax is the current value of the trim potentiometer of the front panel, given in per cent. In conjunction with the nominal current CurrNom of a channel one can calculate the actual maximal output current of the channel.

$$\text{CurrentLimit} = \text{CurrentNomial} * \text{CurrentMax}$$

This current value CurrentLimit is the reference for setting the status bit IsCurrentLimitExceeded.

SupplyP5

Offset Bytes (rel. to BA)	Name	Data type	Access
0x0024	SupplyP5	float	r

The actual value of the +5 line of the power supply, given in V.

SupplyP12

Offset Bytes (rel. to BA)	Name	Data type	Access
0x0028	SupplyP12	float	r

The actual value of the +12 line of the power supply, given in V.

SupplyN12

Offset Bytes (rel. to BA)	Name	Data type	Access
0x002C	SupplyN12	float	r

The actual value of the -12 line of the power supply, given in V.

Temperature

Offset Bytes (rel. to BA)	Name	Data type	Access
0x0030	Temperature	float	r

The actual temperature of the board, given in °C.

SerialNumber

Offset Bytes (rel. to BA)	Name	Data type	Access
0x0034	SerialNumber	uint32	r

The Serial number of the module as long integer value.

FirmwareRelease

Offset Bytes (rel. to BA)	Name	Data type	Access
0x0038	FirmwareRelease	uint8[4]	r

The firmware release as a sequence of four unsigned short integer values.

PlacedChannels

Offset Bytes (rel. to BA)	Name	Data type	Access
0x003C	PlacedChannels	uint16	r

For each existent channel the corresponding bit is set in this word.

For example, a fully equipped 4 channel module VHS 40x has PlacedChannels = 0x000f, a fully equipped 12 channel module VHS C0x has PlacedChannels = 0x0fff .

DeviceClass

Offset Bytes (rel. to BA)	Name	Data type	Access
0x003E	DeviceClass	uint16	r

This is a constant value to divide device families in iseg firmware and applications.

For VHS x0x this value is 20 (0x14).

RestartTimeAfterReloadSetValues

Offset Bytes (rel. to BA)	Name	Data type	Access
0x0050	RestartTimeAfterRecallSetValues	uint16	r/w

This is value for a delay until restart the HV - activation of the stored setON of the corresponding channels – after the control command doRecallSetValues has been sent.

RestartTimeAfterRecallSetValues unit [ms]

ADC SamplesPerSecond SPS

Offset Bytes (rel. to BA)	Name	Data type	Access
0x0058	ADCSamplesPerSecond	uint16	r/w

Adjusts the number of averages of the programmable ADC filter of the HV module. Possible values are 500, 100, 60, 50, 25, 10 and 5 SPS. Notch should be set with 60 SPS using a 110V line with 60Hz and 50 SPS using a 230V line with 50Hz in order to improve the common-mode rejection of these frequencies. However a SPS value of the ADC will increase the main loop time by $4 \cdot 1/\text{SPS}$ multiplied with the number of channels for device.

Factory settings: 500 SPS

DigitalFilter

Offset Bytes (rel. to BA)	Name	Data type	Access
0x005A	DigitalFilter	uint16	r/w

The digital filter in the firmware of the processor reduces the white noise of the analog values of channel VoltageMeasure, channel CurrentMeasure. The digital filtering gives the possibility to get a higher precision and to react fast on changes of the measured values. The filter is not used during a voltage ramp. The filter is restarted after a significant change of the signal. The value DigitalFilter represents the number of filter steps.

Possible steps are: 1, 16, 64, 256, 512 and 1024

Factory settings: 64

VendorId

Offset Bytes (rel. to BA)	Name	Data type	Access
0x005C	VendorId	Uint8[4]	r

This is a constant value to identify the vendor / manufacturer. The value is {0x69;0x73;0x65;0x67}, or in ASCII {"I","S","E","G"}.

2.2.2 Channel registers

The channel Status and Control information will allow to monitor and control output voltage, output current, control and status information of each channel. These detailed information can be collected in groups and several channel can be set and/ or controlled with help of group commands).

ChannelStatus

Offset Bytes (rel. to ChAddr)	Name	Data type	Access
0	ChannelStatus	uint16	R

Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
isVLIM	isCLIM	isTRIP	isEINH	isVBNDs	isCBNDs	res	res	isCV	isCC	isEMCY	isRAMP	isON	isIERR	res	res

isVLIM	IsVoltageLimitExceeded	Hardware- voltage limit has been exceeded (when KillEnable=0)
isCLIM	IsCurrentLimitExceeded	Hardware- current limit has been exceeded (when KillEnable=0)
isTRIP	IsTripSet	Trip is set when Iset has been exceeded (when KillEnable=1)
isEINH	IsExtInhibit	External Inhibit
isVBNDs	IsVoltageBoundsExceeded	Voltage out of bounds
isCBNDs	IsCurrentBoundsExceeded	Current out of bounds
isCV	IsControlledVoltage	Voltage control active
isCC	IsControlledCurrent	Current control active
isEMCY	IsEmergency	Emergency off without ramp
isON	IsOn	On
isRAMP	IsRamping	Ramp is running
isIERR	IsInputError	Input error
res	Reserved	

The channel status register describes the actual status. Depending on the status of the module the bits will be set or reset.

The bit IsInputError is set if the given parameter isn't plausible or it exceeds the module parameters (e.g. if the command Vset=4000V is given to a module with NominalVoltage=3000V). The bit IsInputError isn't set if the given values are temporarily not possible (e.g. Vset=2800 at a module with NominalVoltage=3000V, but HardwareLimitVoltage=2500V). A certain signature which kind of input error it is does not yet happen.

The status bits isVoltageBoundsExceeded resp. isCurrentBoundsExceeded are set:

$$\begin{aligned} \text{if } (|V_{\text{meas}} - V_{\text{set}}| > V_{\text{bounds}}) & \quad \text{isVoltageBoundsExceeded} = 1; \\ \text{if } (|I_{\text{meas}} - I_{\text{set}}| > I_{\text{bounds}}) & \quad \text{isCurrentBoundsExceeded} = 1; \end{aligned}$$

ChannelControl

Offset Bytes (rel. to ChAddr)	Name	Data type	Access
2	ChannelControl	uint16	r/w

Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
res	res	res	res	setAVBND	setACBND	res	res	res	res	setEMCY	res	setON	res	res	res

setEMCY	SetEmergency	Set "Emergency": shut off the channel without ramp, clear the Vset value
setON	SetOn	Set On, if 1; set Off if 0: ramp the output to Vset or to Zero
setAVBND	SetAsymmetricVoltageBounds	Set setAVBND, if 1 - set asymmetric voltage bounds; if 0 - set symmetric voltage bounds
setACBND	SetAsymmetricCurrentBounds	Set setACBND, if 1 - set asymmetric current bounds; if 0 - set symmetric current bounds
res	Reserved	

The signals SetOn and SetEmergency control basic functions of the channel. The signal SetOn is switching ON the HV of the channel and is a precondition for giving voltage to the output. As far as a VoltageSet has been set and no event has occurred and is not registered yet (in minimum, bit 5 and bit 10 to 15 of ChannelEventStatus register must be 0), a start of a HV ramp will be synchronized (a ramp is a software controlled, time proportionally increase / decrease of the output voltage).

There are methods to observe the high voltage via the measured values of voltage and current in stable state outside of a ramp. For this purpose the set values VoltageBounds, VoltageMinIkSet, CurrentBounds and CurrentMinIkSet are used to define a tolerance bounds for the measurement values. When the measured values crossing the defined bounds an event will be generated.

The ChannelControl bits setAVBND and setACBND define whether the tolerance bounds are asymmetric setA[V/C]BND(1) to the set value as an absolute value or symmetric setA[V/C]BND(0) as a relative value to the set value.

setAVBND(1)

VoltageIkMaxSet \leq VoltageMeasure \leq VoltageIkMaxSet No event!

VoltageIkMaxSet > VoltageMeasure or
VoltageMeasure > VoltageIkMaxSet IsVoltageBoundsExceeded(1), ModuleStatus
EventVoltageBounds(1), ModuleEventStatus

setAVBND(0)

VoltageSet-VoltageBounds \leq VoltageMeasure \leq VoltageSet+VoltageBounds No event!

VoltageSet-VoltageBounds > VoltageMeasure or
VoltageMeasure > VoltageSet+VoltageBounds IsVoltageBoundsExceeded(1), ModuleStatus
EventVoltageBounds(1), ModuleEventStatus

setACBND(1)

CurrentIkMaxSet \leq CurrentMeasure \leq CurrentIkMaxSet No event!

CurrentIkMaxSet > CurrentMeasure or
CurrentMeasure > CurrentIkMaxSet IsCurrentBoundsExceeded(1), ModuleStatus
EventCurrentBounds(1), ModuleEventStatus

setACBND(0)

CurrentSet-CurrentBounds \leq CurrentMeasure \leq CurrentSet+CurrentBounds No event!

CurrentSet-CurrentBounds > CurrentMeasure or
CurrentMeasure > CurrentSet+CurrentBounds IsCurrentBoundsExceeded(1), ModuleStatus
EventCurrentBounds(1), ModuleEventStatus

A special feature is the correct changeover from symmetric to asymmetric bounds or from asymmetric to symmetric bounds:

setA[V/C]BND(0) to setA[V/C]BND(1)

Voltage: VoltageIkMaxSet=VoltageSet+VoltageBounds
VoltageIkMinSet=VoltageSet-VoltageBounds
Current: when ChannelStatus isCC(1)
CurrentIkMaxSet=CurrentSet+CurrentBounds
CurrentIkMinSet=CurrentSet-CurrentBounds
ChannelStatus isCC(0), ChannelStatus isON(1), ChannelStatus isRAMP(0)
CurrentIkMaxSet=CurrentMeasure+CurrentBounds
CurrentIkMinSet=CurrentMeasure-CurrentBounds

setA[V/C]BND(1) to setA[V/C]BND(0)

Voltage: VoltageBounds=(VoltageIkMaxSet- VoltageIkMinSet)/2
Current: CurrentBounds=(CurrentIkMaxSet-CurrentIkMinSet)/2

ChannelEventStatus

Offset Bytes (rel. to ChAddr)	Name	Data type	Access
4	ChannelEventStatus	uint16	r/w

Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EVLIM	ECLIM	ECTRP	EEINH	EVBNDs	ECBNDs	res	res	ECV	ECC	EEMCY	EEOR	Eon2Off	EIER	res	res

EVLIM	EventVoltageLimit	Event: Hardware- voltage limit has been exceeded
ECLIM	EventCurrentLimit	Event: Hardware- current limit has been exceeded
ETRIP	EventTrip	Event: Trip is set when Iset has been exceeded (when KillEnable=1)
EEINH	EventExtInhibit	Event external Inhibit
EVBNDs	EventVoltageBounds	Event: Voltage out of bounds
ECBNDs	EventCurrentBounds	Event: Current out of bounds
ECV	EventControlledVoltage	Event: Voltage control
ECC	EventControlledCurrent	Event: Current control
EEMCY	EventEmergency	Event: Emergency
EEOR	EventEndOfRamp	Event: End of ramp
EOn2Off	EventOnToOff	Event: Change from state "On" to "Off" without ramp ¹
EIER	EventInputError	Event: Input Error
res	Reserved	

An event bit is permanently set if the status bit is 1 or changes to 1. Different to the status bit an event bit isn't reset automatically. A reset has to be done by customer by writing an 1 to this event bit.

ChannelEventMask

Offset Bytes (rel. to ChAddr)	Name	Data type	Access
6	ChannelEventMask	uint16	r/w

Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MEVLIM	MECLIM	MECTRP	MEEINH	MEVBNDs	MECBNDs	res	res	MECV	MECC	MEEMCY	MEEOR	MEOn2Off	MEIERR	res	res

MEVLIM	MaskEventVoltageLimit	EventMask: Hardware- voltage limit has been exceeded
MECLIM	MaskEventCurrentLimit	EventMask: Hardware- current limit has been exceeded
METRIP	MaskEventTrip	EventMask: Voltage limit or Current limit or Iset has been exceeded (when KillEnable=1)
MEEINH	MaskEventExtInhibit	EventMask: External Inhibit
MEVBNDs	MaskEventVoltageBounds	EventMask: Voltage out of bounds
MECBNDs	MaskEventCurrentBounds	EventMask: Current out of bounds
MECV	MaskEventControlledVoltage	EventMask: Voltage control
MECC	MaskEventControlledCurrent	EventMask: Current control
MEEMCY	MaskEventEmergency	EventMask: Emergency off
MEEOR	MaskEventEndOfRamp	EventMask: End of ramp
MEOn2Off	MaskEventOnToOff	EventMask: Change from state on to off without ramp
MEIER	MaskEventInputError	EventMask: Input Error
res	Reserved	

VoltageSet

Offset Bytes (rel. to ChAddr)	Name	Data type	Access
8	VoltageSet	Vset	float

The value of VoltageSet (Vset) is the preset for voltage regulation. Valid values are between 0 and the actual hardware limit value. This actual hardware limit value computes as follows:

$$\text{voltage limit} = \text{VoltageNominal} * \text{VoltageMax}$$

When writing values between the actual hardware limit and the nominal value, then the module reduces these values to the value of the actual hardware limit. When writing values above the nominal data or smaller than 0 an input error is indicated by setting the bit IsInputError.

CurrentSet / CurrentTrip

Offset Bytes (rel. to ChAddr)	Name		Data type	Access
12	CurrentSet / CurrentTrip	Iset/Itrip	float	r/w

The value of CurrentSet is the preset for current regulation. Valid values are between 0 and the actual hardware limit value. This actual hardware limit value computes as follows:

$$\text{current limit of channel } x = \text{CurrentNominal} * \text{CurrentMax}$$

When writing values between the actual hardware limit and the nominal value, then the module reduces these values to the value of the actual hardware limit. When writing values above the nominal data or smaller than 0 an input error is indicated by setting the bit IsInputError.

In case of KillEnable=1 there no current regulation in the module active. Then the item CurrentSet (Iset) is replaced by CurrentTrip (Itrip). When exceeding this value a current trip event is registered ad the voltage output is set to 0V.

VoltageMeasure

Offset Bytes (rel. to ChAddr)	Name		Data type	Access
16	VoltageMeasure	Vmeas	float	r/w

VoltageMeasure (Vmeas) is the actual measured value of voltage, in V.

CurrentMeasure

Offset Bytes (rel. to ChAddr)	Name		Data type	Access
20	CurrentMeasure	Imeas	float	r/w

CurrentMeasure (Imeas) is the actual measured value of current, in A.

VoltageBounds VoltageIikMaxSet

Offset Bytes (rel. to ChAddr)	Name		Data type	Access
24	VoltageBounds	Vbounds	float	r/w
24	VoltageIikMaxSet	VIikMaxSet	float	r/w

VoltageBounds:

By the help of VoltageBounds (Vbounds) there is defined a region around VoltageSet (Vset), where the actual values are interpreted as good. This region is defined as follows:

$$| V_{\text{meas}} - V_{\text{set}} | \leq V_{\text{bounds}}$$

If this area is left, a corresponding event is registered.

VoltageIikMaxSet:

By the help of VoltageIikMaxSet (VIikMaxSet) and VoltageIikMinSet (VIikMinSet) there is defined a region around VoltageSet (Vset), where the actual values are interpreted as good. This region is defined as follows:

$$VIikMinSet \leq V_{\text{meas}} \leq VIikMaxSet$$

If this area is left, a corresponding event is registered.

**CurrentBounds
CurrentIlkMaxSet**

Offset Bytes (rel. to ChAddr)	Name		Data type	Access
28	CurrentBounds	lbounds	float	r/w
28	CurrentIlkMaxSet	IlkMaxSet	float	r/w

CurrentBounds:

By the help of CurrentBounds (lbounds) there is defined a region around CurrentSet (Iset), where the actual values are interpreted as good. This regions is defined as follows:

$$| I_{meas} - I_{set} | \leq I_{bounds}$$

If this area is left, a corresponding event is registered.

CurrentIlkMaxSet:

By the help of CurrentIlkMaxSet (IlkMaxSet) and CurrentIlkMinSet (IlkMinSet) there is defined a region around CurrentSet (Iset), where the actual current are interpreted as good. This region is defined as follows:

$$I_{lkMinSet} \leq I_{meas} \leq I_{lkMaxSet}$$

If this area is left, a corresponding event is registered.

VoltageNominal / VoltageMaxSet

Offset Bytes (rel. to ChAddr)	Name		Data type	Access
32	VoltageNominal / VoltageMaxSet	Vnom	float	r/(w)

This is the maximal possible output voltage of the channel. Normally this is the fixed value of the HV channel hardware (given by the technical specifications of the module). If the user writes a lower VoltageMaxSet, this value appears here. VoltageMaxSet is writeable in mode ModuleStatus IsStop = 1 in the range (0 < VoltageMaxSet ≤ VoltageNominal)

CurrentNominal / CurrentMaxSet

Offset Bytes (rel. to ChAddr)	Name		Data type	Access
36	CurrentNominal / CurrentMaxSet	Inom	float	r/(w)

This is the maximal possible output current of the channel. Normally this is the fixed value of the HV channel hardware (given by the technical specifications of the module). If the user writes a lower CurrentMaxSet, this value appears here. CurrentMaxSet is writeable in mode ModuleStatus IsStop = 1 in the range (0 < CurrentMaxSet ≤ CurrentNominal)

VoltagellkMinSet

Offset Bytes (rel. to ChAddr)	Name		Data type	Access
40	VoltagellkMinSet	VllkMinSet	float	r/w

see VoltagellkMaxSet above

CurrentllkMinSet

Offset Bytes (rel. to ChAddr)	Name		Data type	Access
44	CurrentllkMinSet	llkMinSet	float	r/w

see CurrentllkMaxSet above

2.2.3 Groups

The Multi Channel VME module offers an extended and flexible range of group functions. There are both well defined Fix Groups and free configurable variable groups.

Each definition of a group consists of 2 words (4 bytes).

In the Fix Groups these 2 words hold the value of a floating point value or a logical information. In Variable Groups is one word an identifier for the group. The other word holds the information about the group members (which channel is a member of the group) or it gives an overview over a characteristic in all channels.

Caution!

In order to avoid a malfunction both words of a group have to be written, even in case just one has been changed.

Four different groups have been established:

- Set group
- Status group
- Monitoring group
- Timeout group

2.2.3.1 Fix Groups

The functions and characteristics of the groups are fix defined.

SetVoltageAllChannels

Offset Bytes (rel. to BA)	Name	Data type	Access
0x02A0	SetVoltageAllChannels	float	r/w

The value of the set voltage in V for all channels will be submitted to the group as a floating point number in the 4 bytes. This value is accepted, if the corresponding channel characteristics permit it. Otherwise it's ignored.

SetCurrentAllChannels

Offset Bytes (rel. to BA)	Name	Data type	Access
0x02A4	SetCurrentAllChannels	float	r/w

The value of the set current in A for all channels will be submitted to the group as a floating point number in the 4 bytes. This value is accepted, if the corresponding channel characteristics permit it. Otherwise it's ignored.

SetVoltageBoundsAllChannels

Offset Bytes (rel. to BA)	Name	Data type	Access
0x02A8	SetVoltageBoundsAllChannels	float	r/w

The value of the voltage bounds in V for all channels will be submitted to the group as a floating point number in the 4 bytes. This value is accepted, if the corresponding channel characteristics permit it. Otherwise it's ignored.

SetCurrentBoundsAllChannels

Offset Bytes (rel. to BA)	Name	Data type	Access
0x02AC	SetCurrentBoundsAllChannels	float	r/w

The value of the current bounds in A for all channels will be submitted to the group as a floating point number in the 4 bytes. This value is accepted, if the corresponding channel characteristics permit it. Otherwise it's ignored.

SetEmergencyAllChannels

Offset Bytes (rel. to BA)	Name	Data type	Access
0x02B0	SetEmergencyAllChannels	uint32	r/w

Is worth without coding. Writing any information to this group triggers an alarm switching off in all channels of the module.

SetOnOffAllChannels

Offset Bytes (rel. to BA)	Name	Data type	Access
0x02B4	SetOnOffAllChannels	uint32	r/w

The data word holds the function of the command:

- data = 1: Switch on all channels of the module
- data = 0: Switch off all channels of the module

SetVoltageIikMinSetAllChannels

Offset Bytes (rel. to BA)	Name	Data type	Access
0x02B8	SetVoltageIikMinSetAllChannels	float	r/w

The value of the SetVoltageIikMaxSetAllChannels in V for all channels will be submitted to the group as a floating point number in the 4 bytes. This value is accepted, if the corresponding channel characteristics permit it. Otherwise it's ignored.

SetCurrentIikMinSetAllChannels

Offset Bytes (rel. to BA)	Name	Data type	Access
0x02BC	SetCurrentIikMinSetAllChannels	float	r/w

The value of the SetCurrentIikMinSetAllChannels in A for all channels will be submitted to the group as a floating point number in the 4 bytes. This value is accepted, if the corresponding channel characteristics permit it. Otherwise it's ignored.

2.2.3.2 Variable Groups

2.2.3.2.1 Set group

Set groups will be used in order to set channels to a same value, which happen to carry the identical channel value. Therefore within the group will be defined:

- Member of the group
 - o Each member will be activated in the member list
- Type of the group
 - o constant: SetGroupType
- Channel characteristics
 - o Coding of characteristics , which are to be set commonly
- Control mode
 - o Divides between a one-time setting of the slave channel property and a permanently copying of the Master channel's property to the slave channels
- Master channel
 - o Number of the channel, which characteristics will be transferred to the other channels.
 - o Is just necessary for Set groups which set a value.
If functions have to be initialized e.g. start of ramp then there is no Master channel

SetGroup

Offset Bytes (rel. to GrAddr)	Name	Data type	Access
0	MemberList	uint16	r/w
2	TypeSet	uint16	r/w

MemberList:

Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
res	res	Res	res	CH11	CH10	CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0

TypeSet:

Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TYPE1	TYPE0	Res	res	res	res	res	MOD0	SET3	SET2	SET1	SET0	MCH3	MCH2	MCH1	MCH0

TYPE1	TYPE0	Value	
0	0	SetGroupType	Group is defined as Set group

MOD0	Value	
0	0	The group function is done one time
1	1	The group function is done permanently

SET3	SET2	SET1	SET0	Value	
0	0	0	1	SetVset	Copy Vset from MCH to all members
0	0	1	0	SetIset	Copy Iset from MCH to all members
0	1	0	0	SetVbnds	Copy Vbounds from MCH to all members
0	1	0	1	SetIbnds	Copy Ibounds from MCH to all members
0	1	1	0	SetVIkMinSet	Copy VIkMinSet from MCH to all members
0	1	1	1	SetIIIkMinSet	Copy IIIkMinSet from MCH to all members
1	0	1	0	SetOn	Switch ON/OFF all members depending on setON in MCH
1	0	1	1	SetEmrgCutOff	Switch OFF all members (Emergency OFF)
1	1	1	1	Cloning	Set all properties of members like MCH properties (<i>in preparation</i>)

MCH3	MCH2	MCH1	MCH0	Value	
0	0	0	0	0	1: Channel 0 is MasterChannel MCH
0	0	0	1	1	1: Channel 1 is MasterChannel MCH
...
0	0	1	1	3	1: Channel 3 ist MasterChannel MCH

2.2.3.2.2 Status group

Status groups are used to report the status of a single characteristic of all channels simultaneously. No action is foreseen. Therefore within the group has to be defined :

- type of the group
 - o constant: StatusGroupType
- channel characteristics
 - o coding of characteristics , which is to be reported

StatusGroup

Offset Bytes (rel. to GrAddr)	Name	Data type	Access
0	ChannelStatusList	uint16	r/w
2	TypeStatus	uint16	r/w

ChannelStatusList:

Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Res	res	res	res	CHST11	CHST10	CHST9	CHST8	CHST7	CHST6	CHST5	CHST4	CHST3	CHST2	CHST1	CHST0

TypeStatus:

Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TYPE1	TYPE0	res	res	res	res	res	res	STAT3	STAT2	STAT1	STAT0	Res	res	res	Res

TYPE1	TYPE0	Value	
0	1	StatusGroupType	Group will be defined as Status group

STAT3	STAT2	STAT1	STAT0	Value	
0	0	1	1	ChkIsOn	check channel Status.isON (is on)
0	1	0	0	ChkIsRamping	check channel Status.isRAMP (is ramping)
0	1	1	0	ChkIsControlledCurrent	check channel Status.isCC (is current control)
0	1	1	1	ChkIsControlledVoltage	check channel Status.isCV (is voltage control)
1	0	1	0	ChkIsCurrentBounds	check channel Status.isCBNDs (is current bounds)
1	0	1	1	ChkIsVoltageBounds	check channel Status.isVBNDs (is voltage bounds)
1	1	0	0	ChkIsExternalInhibit	check channel Status.isEINH (is external inhibit)
1	1	0	1	ChkIsTrip	check channel Status.isTRIP (is trip)
1	1	1	0	ChkIsCurrentLimit	check channel Status.isCLIM (is current limit exceeded)
1	1	1	1	ChkIsVoltageLimit	check channel Status.isVLIM (is voltage limit exceeded)

2.2.3.2.3 Monitoring group

Monitoring groups are used to observe a single characteristic of selected channels simultaneously and in case of need take action. Therefore the group has to be defined :

- members of the group
 - o each member will be activated in the member list
- type of the group
 - o constant: MonitoringGroupType
- channel characteristics
 - o coding of characteristics , which is to be monitored
- control mode
 - o coding of the control function, i.e. which kind of change in the group-image shall cause a signal.
- activity
 - o define , which activity has to happen after the event

MonitoringGroup

Offset Bytes (rel. to GrAddr)	Name	Data type	Access
0	MemberList	uint16	r/w
2	TypeMonitoring	uint16	r/w

MemberList

Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
res	Res	res	res	CH11	CH10	CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0

TypeMonitoring:

Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TYPE1	TYPE0	ACT1	ACT0	res	res	res	MOD0	MON3	MON2	MON1	MON0	res	res	Res	res

TYPE1	TYPE0	Value	
1	0	MonitoringGroupType	Group will be defined as Monitoring group

MON3	MON2	MON1	MON0	Value	
0	0	1	1	MonitorIsOn	monitor channel Status.isON (is on)
0	1	0	0	MonitorIsRamping	monitor channel Status.isRAMP (is ramping)
0	1	1	0	MonitorIsControlledCurrent	monitor channel Status.isCC (is current control)
0	1	1	1	MonitorIsControlledVoltage	monitor channel Status.isCV (is voltage control)
1	0	1	0	MonitorIsCurrentBounds	monitor channel Status.isCBNDs (is current bounds)
1	0	1	1	MonitorIsVoltageBounds	monitor channel Status.isVBNDs (is voltage bounds)
1	1	0	0	MonitorIsExternalInhibit	monitor channel Status.isEINH (is external inhibit)
1	1	0	1	MonitorIsTrip	monitor channel Status.isTRIP (is trip)
1	1	1	0	MonitorIsCurrentLimit	monitor channel Status.isCLIM (is current limit exceeded)
1	1	1	1	MonitorIsVoltageLimit	monitor channel Status.isVLIM (is voltage limit exceeded)

MOD0	Value	
0	0	event will happen if at least one Channel == 0
1	1	event will happen if at least one Channel == 1

ACT1	ACT0	Value	
0	0	0	No special action ; EventGroupStatus[grp] will be set
0	1	1	Ramp down of group; EventGroupStatus[grp] will be set
1	0	2	Switch OFF of group without ramp; EventGroupStatus[grp] will be set
1	1	3	Switch OFF of module without ramp; EventGroupStatus[grp] will be set

2.2.3.2.4 Timeout group

Timeout groups are necessary to keep the timing for the time controlled Trip function and to define the action which has to happen after a Trip.

Therefore in the group will be defined:

- members of group
 - o each member will be activated in a word MemberList
- type of the group
 - o constant: TimeOutGroupType
- activity
 - o define , which activity has to happen after time controlled Trip
- timeout
 - o coding of Timeout-time as 12 Bit Integer

TimeOutGroup:

Offset Bytes (rel. to GrAddr)	Name	Data type	Access
0	MemberList	uint16	r/w
2	TypeTimeOut	uint16	r/w

MemberList:

Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
res	res	res	res	CH11	CH10	CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0

TypeTimeOut:

Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TYPE1	TYPE0	ACT1	ACT0	TOT11	TOT10	TOT9	TOT8	TOT7	TOT6	TOT5	TOT4	TOT3	TOT2	TOT1	TOT0

TYPE1	TYPE0	Value	
1	1	TimeOutGroupType	Group will be defined as Timeout group .

ACT1	ACT0	Value	
0	0	0	No special action ; EventGroupStatus[grp] will be set
0	1	1	Ramp down of group EventGroupStatus[grp] will be set
1	0	2	Switch OFF of group without ramp; EventGroupStatus[grp] will be set
1	1	3	Switch OFF of module without ramp; EventGroupStatus[grp] will be set

TOT[11..0]: Binary coded Timeout-time in ms (0..4096ms)

2.3 Events and interrupts

Remark: The activation of interrupts at the VME bus is not realized yet. The event handling is realized

The module provides an extended event collecting and interrupt logic. This is necessary to monitor extraordinary events and forward them to the host.

Events can be generated by:

- occurrence of special conditions in the module status (e.g. safety loop open, temperature too high)
- occurrence of special conditions in a channel (e.g. over-voltage, over-current, current-trip)
- occurrence of events in channel status (e.g. end of a ramp)
- occurrence of events in a monitoring group
- occurrence of events in a timeout group

The occurrence of such single events will be stored in the EventStatus registers:

- ModuleEventStatus
- ChannelEventStatus
- ModuleEventGroupStatus

Since every appearing event doesn't have inevitably to lead to a report to the host, the EventMask registers exist parallel to the EventStatus registers. These decide whether an occurred event leads to a report to the host or not. If the event shall be reported, the responsible bit must be set in the mask register.



A check of EventStatus and EventMask is made before the HV will be switched on. When bits are set in the EventStatus and the corresponding bits are set in the EventMask the HV cannot be switched on again before the EventStatus bits are reset by writing "1" on the corresponding bit positions.

The report to the host can be made by queries of the bit "IsEventActive" in the ModuleStatus register. This bit is set if an event has occurred and the setting of the event mask enables the passing. Independent of the being of the reason for an event, these remain stored further in the accompanying event status register.

The reset of the individual events is done by a re-write of a 1 to the event bit in the accompanying EventStatus register. It's possible to reset more than one event at the same time. If there is still the reason for the event, the reset is prevented or a new set of an event is immediately carried out.

2.3.1 Events in channels

Main origin of the event logic are the single event sources in the channels. The occurrence of an event is stored in the register ChannelEventStatus of the channel. The accompanying register ChannelEventMask decides if the event is to be reported. An event is reported if the accompanying bit in the mask register is set. To generate a global information about the existence of any event to be reported a sum signal is made. All these sum signals of all channels are stored in the status register ModuleEventChannelStatus

ModuleEventChannelStatus [n] =

(EventVoltageLimit[n] AND MaskEventVoltageLimit[n]) OR
(EventCurrentLimit[n] AND MaskEventCurrentLimit[n]) OR
(EventTrip[n] AND MaskEventTrip[n]) OR
(EventExtInhibit[n] AND MaskEventExtInhibit[n]) OR
(EventVoltageBounds[n] AND MaskEventVoltageBounds[n]) OR
(EventCurrentBounds[n] AND MaskEventCurrentBounds[n]) OR
(EventControlledVoltage[n] AND MaskEventControlledVoltage[n]) OR
(EventControlledCurrent[n] AND MaskEventControlledCurrent[n]) OR
(EventEmergency[n] AND MaskEventEmergency[n]) OR
(EventEndOfRamp[n] AND MaskEventEndOfRamp[n]) OR
(EventOnToOff[n] AND MaskEventOnToOff [n]) OR

(EventInputError[n] AND MaskEventInputError[n])

where is:

ModuleEventChannelStatus[n]: ch-th bit of the register ModuleEventChannelStatus

EventVoltageLimit[n]: bit EventVoltageLimit of register ChannelEventStatus of the ch-th channel

MaskEventVoltageLimit[n]: bit MaskEventVoltageLimit of register ChannelEventMask of the ch-th channel

The selection of channels is done by the register ModuleEventChannelMask. Only those channels can report an event that have a set bit in this mask register. The sum event of all channel events is the (internal) signal EventChannelActive:

EventChannelActive = (ModuleEventChannelStatus[0] AND ModuleEventChannelMask[0]) OR
 (ModuleEventChannelStatus[1] AND ModuleEventChannelMask[1]) OR
 ...
 (ModuleEventChannelStatus[n] AND ModuleEventChannelMask[n])

2.3.2 Events in groups

Like written before groups are also able to generate Events. These events will be collected in the status word ModuleEventGroupStatus. This status word is 32 bits wide. It consists of the status registers ModuleEventGroupStatusHigh and ModuleEventGroupStatusLow, each 16bit wide. With help of the accompanying mask register ModuleEventGroupMask the events are filtered and the (internal) signal of the groups EventGroupActive will be generated.

EventGroupActive = (ModuleEventGroupStatus[0] AND ModuleEventGroupMask[0]) OR
 (ModuleEventGroupStatus[1] AND ModuleEventGroupMask[1]) OR
 ...
 (ModuleEventGroupStatus[23] AND ModuleEventGroupMask[24])

2.3.3 Events in characteristics of the whole module

These events are events of single characteristics of the module. An event is stored in the register EventModuleStatus. This register also has a mask register for filtering. The sum signal of this type of events is the (internal) signal EventModuleActive.

EventModuleActive = (EventTemperatureNotGood AND MaskEventTemperatureNotGood) OR
 (EventSupplyNotGood AND MaskEventSupplyNotGood) OR
 (EventSafetyLoopNotGood AND MaskEventSafetyLoopNotGood) OR
 (EventRestart AND MaskEventResart) OR

2.3.4 Event status of the module

The event status of the module is summarized out of the event status of the channels, of the groups and of the module single characteristics. This sum signal IsEventActive is part of the register ModuleStatus:

IsEventActive = EventChannelActive OR
 EventGroupActive OR
 EventModuleActive

2.4 Special registers

2.4.1 Setting of Basis Address

Offset Bytes (rel. to BA)	Name	Data type	Access
0x03A0	NewBaseAddress	uint16	r/w
0x03A2	NewBaseAddressXor	uint16	r/w
0x03A4	OldBaseAddress	uint16	r
0x03A6	NewBaseAddressAccepted	uint16	r

As shown in the preliminary remarks to section 4, the module is bound into the VME address room by defining the Basis Address BA. This address is the begin of a 1kByte wide memory segment. the address BA is free in the bits A15 to A10, the bits A9 to A1 are fixed to 0.

binary: BA = bbbbb00 00000000 (with b={0|1})
 hexadezimal: BA = xy00 (with x={0..F}, y={0,4,8,C}).

The default value (factory setting and setting when started with jumper "ADR" on the topside of the board has been set) is BA=0x4000.

New address setting is done using four registers:

In register "NewBaseAddress" the new base address (byte counting) is to write. In register "NewBaseAddressXor" the complementary value of "NewBaseAddress" is to write.

NewBaseAddressXor = NewBaseAddress XOR 0xFFFF

When both values are written, and the condition is fulfilled, the new address is accepted. If the new address doesn't point to the beginning of a 1kByte segment, it is corrected to the beginning of the next smaller segment.

After that, the value is stored into EEPROM.

This new Base Address is used after the next reset (e.g. after PowerOn, SYSRESET or a special command). Until this the old address is valid.



When the jumper "ADR" is set the Base Address of the module will reset to the default address 0x4000 after a power up. This function can be used when there is no communication for instance the Base Address is unknown.
 When the jumper is not set the stored address inside of the module will be used as Base Address.

2.4.2 Special Control Register

Offset Bytes (rel. to BA)	Name	Data type	Access
0x03B0	SpecialControlStatus	uint16	r
0x03B2	SpecialControlCommand	uint16	r/w

Both these registers "SpecialControlStatus" and "SpecialControlCommand" are used for maintenance and service purposes. Their usage is explained in a separate manual.

1.