

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 notification to the users.

Filename VDS interface manual version 1.02 as of 06/10/2011

Contents:

Inhaltsverzeichnis

| | |
|--|----|
| 1. General Information..... | 4 |
| 2. Operating principle | 4 |
| 2.1 Hardware functions..... | 4 |
| 2.1.1 Front panel indicators..... | 4 |
| 2.1.2 Interlock input..... | 4 |
| 2.1.3 KillEnable / Kill / ClearKill..... | 4 |
| 2.1.4 Modus: Voltage regulation / Current trip..... | 4 |
| 2.2 Software functions..... | 5 |
| 2.2.1 Analogous values..... | 5 |
| 2.2.1.1 Voltage bounds / Voltage interlock maximum set / Voltage interlock minimum set | |
| Current bounds / Current interlock maximum set / Current interlock minimum set..... | 5 |
| 2.2.2 Digital values..... | 6 |
| 2.2.2.1 Status and event..... | 6 |
| 2.2.2.2 Event status and event mask..... | 6 |
| 2.2.3 Summarizing of channel characteristics into groups..... | 7 |
| 2.2.4 Reactions after events (Soft-Kill features)..... | 7 |
| 2.2.5 Autostart..... | 8 |
| 3. VME-Interface..... | 9 |
| 3.1 Memory space..... | 10 |
| 3.2 Details to the memory space..... | 13 |
| 3.2.1 Module registers..... | 13 |
| 3.2.2 Channel registers..... | 23 |
| 3.2.2.1 Fix Groups..... | 30 |
| 3.2.2.2 Variable Groups..... | 32 |
| 3.2.2.2.1 Set group..... | 32 |
| 3.2.2.2.2 Status group..... | 34 |
| 3.2.2.2.3 Monitoring group..... | 35 |
| 3.2.2.2.4 Timeout group..... | 37 |
| 3.3 Events and interrupts..... | 38 |
| 3.3.1 Events in channels..... | 38 |
| 3.3.2 Events in groups..... | 39 |
| 3.3.3 Events in characteristics of the whole module..... | 39 |
| 3.3.4 Event status of the module..... | 39 |
| 3.4 Special registers..... | 40 |
| 3.4.1 Setting of Basis Address..... | 40 |
| 3.4.2 Special Control Register..... | 40 |

Appendix RAM mapping

1. General Information

The VDS modules are High Voltage power supplies controlled exclusively via the VME bus system. The factory setup for the **VME base address is 0x4000** (byte-wise, respectively). This setup can be restored using the internal jumper "ADR" on the topside of the board (see section 3.4.1 Setting of Base Address).

2. Operating principle

The functionality of the module is achieved by a close interaction of hardware and software. Time-critical functions have a hardware implementation. An extensive list of further functions is controlled by firmware algorithms.

2.1 Hardware functions

2.1.1 Front panel indicators

2.1.2 Interlock input

The Interlock is an external input signal. It can be used for a fast shut-down of the high voltage generation, e.g. in critical system states. An activation of the signal will switch off the high voltage production immediately. The following inputs are available:

- global Interlock signal to shut down the whole module,
 - is implemented as a current loop (safety loop),
 - can be deactivated by a jumper;
- optional: single-channel INHIBIT via individual TTL-inputs ;
 - input open or at high level: channel is active
 - input connected to ground or at low level: channel is switched off

2.1.3 KillEnable / Kill / ClearKill

The flag SetKillEnable is a global control flag for the whole module. It defines the response of the module if the predefined voltage limit (Vmax) or the predefined current limit (Imax/Iset/Itrip) is exceeded. If KillEnable is active, the exceedance of Imax/Iset will trigger a Kill-signal in the respective channel . This signal will switch off the channel immediately without ramp.

The signal ClearKill is used to reset stored Kill-signals in all channels. Without this reset high voltage generation cannot be restarted in a channel that received a Kill-signal.

2.1.4 Voltage regulation and Current trip

Depending on the flag SetKillEnable as described above and the operating point a channel can be in one of the three following operation modes:

Voltage regulation (CV)

In the mode Voltage regulation the module works as a constant voltage source. For this mode it is required that the predefined current value Iset or Itrip is greater than the resulting output current.

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), the channel is switched off immediately.

2.2 Software functions

The qualities and functions described below are determined by the internal control of the module substantially. Main item is a microcontroller, which can measure or provide the analogous condition quantities over analogous I/O assemblies (ADC or DAC) and which determines the switching states of the hardware over digital I/O ports. The microcontroller controls and supervises the function of the voltage generation in the channels, the compliance with the limiting values, the occurrence of certain events. Furthermore the communication on the interface is controlled by the microcontroller. Details to this are described in section 0. Single module and channel characteristics are described in the following

2.2.1 Analogous values

Control items as well as status items come under this category

Analogous control items of the module

- voltage ramp speed
- restart time after recall set values
- voltage maximum set
- current maximum set
- ADC samples per second
- digital filter set

Analogous control items of a channel

- voltage set
- current trip set
- voltage bounds
- voltage interlock maximum set
- voltage interlock minimum set
- current bounds
- current interlock maximum set
- current interlock minimum set

Analogous status items of the module

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

Analogous status items of a channel

- voltage out
- current out
- voltage nominal
- current nominal

2.2.1.1 Voltage bounds / Voltage interlock maximum set / Voltage interlock minimum set Current bounds / Current interlock maximum set / Current interlock minimum set

This function of the module can be used for a largely autonomous business. With the help of the control variables VoltageBounds, Voltage interlock maximum set, Voltage interlock minimum set and CurrentBounds, Current interlock maximum set, Current interlock minimum set tubes are formed around the specification values VoltageSet and CurrentSet. If the measured condition sizes output voltage or output current is within these tubes, the condition is as interpreted well. If the condition values leave the specification area, a corresponding fault event is registered.

2.2.2 Digital values

The digital control and state variables serve the setting or re-registration of single module or channel functions.

2.2.2.1 Status and event

You distinguish at the condition items in status and event. In status words the current status of the item is given. Depending on current condition the bits are set or reset by the controller. Unlike this a event is registered in event words without resetting it when the event has finished. A reset of stored events is made by a specific write on the event word.

| | |
|--------|--|
| 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 |

2.2.2.2 Event status and event mask

So that all event sources don't always have to be checked by events on arriving, the module has a hierarchical chain for the combination of the events to a single status bit which represents the short-term condition of the event hierarchy.

This structure for the event processing is built up uniformly for events from the module status, the status of the channels and the group status. An event status register and an event mask register exists respectively.

Event status Combination of the events arrived till now

Event mask Filter which checks the combination of individual events to sum events

A bit in the event mask is assigned to every event bit in the event status register. If the mask bit is set, the occurring of the accompanying event leads sum event to the activation. In turn these sum events are collected in an event status register and connected with an event mask register at this higher level.



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 individual event in the channels sources are starting point of the event logic. Every appearing event (status = 1) is stored in a bit of the event status register of the channel. Bits in a mask register are assigned to these event bits in the channel event status register. A logical AND condition (bit wise) between the event bit and the accompanying mask bit is achieved that a result arises only where the mask bit is set. A following logical OR of all these result bits yields the event status of the channel.

```
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].EventEmergency AND Channel[n].MaskEventEmergency) OR
                           (Channel[n].EventEndOfRamp AND Channel[n].MaskEventEndOfRamp) OR
                           (Channel[n].EventOnToOff AND Channel[n].MaskEventOnToOff ) OR
                           (Channel[n].EventInputError AND Channel[n].MaskEventInputError)
```

The condition of all event statuses of the channels is summarized in the register EventChannelStatus. For the choice or filtration of the channel events a mask register is also assigned (EventChannelMask) here. By means of the AND or ODER combination described in the channel the global signal AnyChannelEventActive of the channels is caused.

```
EventChannelActive = (EventChannelStatus[0] AND EventChannelMask[0]) OR
                           (EventChannelStatus[1] AND EventChannelMask[1]) OR
                           ...
                           (EventChannelStatus[n] AND EventChannelMask[n])
```

Besides the channel based events special conditions can be registered of qualities of the complete module as an event. The following scheme applies to these module events:

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

Parallel to these evaluation structures, events of the groups are supervised. Are described how later, different groups (monitor group, time out group) also can cause events. These stored group events are summarized in the status word EventGroupStatus. With the help of the mask register EventGroupMask the event-collecting signal of the groups EventGroupActive is formed from these group events.

```
EventGroupActive =      (EventGroupStatus[0] AND EventGroupMask[0]) OR
                           (EventGroupStatus[1] AND EventGroupMask[1]) OR
                           ...
                           (EventGroupStatus[32] AND EventGroupMask[32])
```

All summarized events are summarized to the bit IsEventActive of the register ModuleStatus:

```
IsEventActive =           EventChannelActive OR EventModuleActive OR EventGroupActive
```

2.2.3 Summarizing of channel characteristics into groups

The module shows a flexible group function. With the first one there is the possibility to set single specification values in all channels of the module with the help of Fix Groups. Furthermore Variable Groups can be defined. They allow to customize the logical structure of the module to the logical structure of the application. For these Variable Groups group types were pre-defined for whose application there isn't any restriction apart from the maximum number of groups (32). In particular got predefined:

- Set Group:
 - puts the condition of a channel characteristic for selected channels
 - no event generation
- Status Group:
 - represents the status (condition) of a channel characteristic for all channels
 - no event generation
- Monitor Group
 - monitors the condition of a channel characteristic for selected channels
 - event generation in condition change
 - reaction selectable (e.g. switch off)
- Timeout Group:
 - monitors the current trip in selected channels
 - It is prerequisite that the signal KillEnable is turned off
 - Event generation only after expiry of a predefined time within which the trip condition must be active
 - reaction selectable (e.g. switch off)

2.2.4 Reactions after events (Soft-Kill features)

In the event generating groups there is a choice between 4 reactions that have to be executed after the event is generated:

- switch off of the whole module, without ramp
 - high voltage of all channels of the module is switched off
- switch off of all members of the group, without ramp
 - high voltage of all channels of the group is switched off
- switch off of all members of the group, with ramp
 - high voltage of all channels of the group is ramped down
- no reaction
 - no change
 -

2.2.5 Autostart

The Autostart is a functionality which allows a recall/reload of stored values to the corresponding set values. A delayed switch On of the high voltages can be configured. The delay time for a delayed switch ON will be configured in RestartTimeAfterRecallSetValues.

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

- ChannelControl
- ChannelEventMask
- VoltageSet
- CurrentTrip
- VoltageBounds/VoltageIlkMaxSet
- CurrentBounds/CurrentIlkMaxSet
- VoltageIlkMinSet
- CurrentIlkMinSet
- VoltageMaxSet
- CurrentMaxSet

the module:

- ModuleControl
- ModuleEventMask
- ModuleEventChannelMask
- ModuleEventGroupMask
- VoltageRampSpeed
- CurrentRampSpeed
- RestartTimeAfterRecallSetValues
- ADCSSamplesPerSecond
- 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 made in chapter [3.2.1. Module registers](#), ModulStatus, ModulControl, ModuleEventStatus and RestartTimeAfterRecallSetValues.

3. 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 2048 bytes.

The description of RAM addressing in this document is done in a byte addressing type for simple memory count. The access to any property of the module have to be done with two sequentially words. All properties such as VoltageSet or ModuleStatus are 32 bit information.



The VME decoder of the module accept a 32 bit information after the second word has been written. Other words can be transmitted in between.

The RAM memory space begins at the base address (BA). This is a 16bit address, where the 11 LSB bits are 0. The 5 MSB bits can be set by the customer to insert the module's RAM into the VME space.

in bytes:

| | | | | |
|--------------|----|---|--------------------|------------------------------|
| binary: | BA | = | bbbbbb000 00000000 | (with b={0 1}) |
| hexadezimal: | BA | = | xy00 | (with x={0..F}, y={0..8}). |

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=0x2000 (addressing words in VME short access) resp. BA=0x4000 in Bytes

Partition of the memory (given in bytes):

| | | |
|--------------------------|---|-----------|
| BA+0x0000 .. BA+0x009f : | module data . 160 bytes | 64 Bytes |
| BA+0x00a0 .. BA+0x00bf : | control registers for special use | |
| BA+0x00d0 .. BA+0x00e7 : | 8 fixed groups ea. 4 bytes | 32 Bytes |
| BA+0x03a0 .. BA+0x03ff : | 24 channel data blocks ea. 48 bytes | 576 Bytes |
| BA+0x02c0 .. BA+0x033f : | 32 variable groups (set, status, monitoring or timeout groups) data block ea. 8 bytes | 256 Bytes |

3.1 Memory space

Module data

| Offset Bytes (rel. to BA) | Name | Data type | Access |
|---------------------------|---|-----------|--------|
| 0x0000 | ModuleStatus | uint32 | r |
| 0x0004 | ModuleEventStatus | uint32 | r/w |
| 0x0008 | ModuleEventMask | uint32 | r/w |
| 0x000c | ModuleControl | uint32 | r/w |
| 0x0010 | ModulEventChannelStatus | uint32 | r/w |
| 0x0014 | ModulEventChannelMask | uint32 | r/w |
| 0x0018 | ModuleEventGroupStatus | uint32 | r/w |
| 0x001c | ModuleEventGroupMask | uint32 | r/w |
| 0x0020 | VoltageRampSpeed | float | r/w |
| 0x0024 | CurrentRampSpeed | float | r/w |
| 0x0028 | VoltageMax | float | r |
| 0x002c | CurrentMax | float | r |
| 0x0030 | SerialNumber | uint32 | r |
| 0x0034 | FirmwareRelease | uint8[4] | r |
| 0x0038 | PlacedChannels | uint32 | r |
| 0x003c | ChannelNumber_DeviceClass | uint16[2] | r |
| 0x0040 | SupplyP5 | float | r |
| 0x0044 | SupplyP12 | float | r |
| 0x0048 | SupplyN12 | float | r |
| 0x004c | Temperature | float | r |
| 0x0050 | RestartTimeAfterRecallSetValues | uint32 | r/w |
| 0x0054 | ADCSamplesPerSecond | uint32 | r/w |
| 0x0058 | DigitalFilter | uint32 | r/w |
| 0x005C | VendorId : const 'i','s','e','g' = 0x69736567 | uint8[4] | r |

Channels

| Offset Bytes (rel. to BA) | Name | |
|---------------------------|-------------------------|---------------------|
| 0x100 | ChAddr[0] | begin of channel 0 |
| 0x140 | ChAddr[1] | begin of channel 1 |
| 0x180 | ChAddr[2] | begin of channel 2 |
| 0x1c0 | ChAddr[3] | begin of channel 3 |
| 0x200 | ChAddr[4] | begin of channel 4 |
| 0x240 | ChAddr[5] | begin of channel 5 |
| 0x280 | ChAddr[6] | begin of channel 6 |
| 0x2c0 | ChAddr[7] | begin of channel 7 |
| 0x300 | ChAddr[8] ¹ | begin of channel 8 |
| | | |
| 0x0680 | ChAddr[22] ¹ | begin of channel 22 |
| 0x06c0 | ChAddr[23] ¹ | begin of channel 23 |

¹ only in module type VDS 18xx

Channel data

| Offset Bytes (rel. to ChAddr) | Name | Data type | Access |
|-------------------------------|---|-----------|--------|
| 0x00 | ChannelStatus | uint16[2] | r |
| 0x04 | ChannelEventStatus | uint16[2] | r/w |
| 0x08 | ChannelEventMask | uint16[2] | r/w |
| 0x0c | ChannelControl | uint16[2] | r/w |
| 0x10 | VoltageSet | float | r/w |
| 0x14 | CurrentSet / CurrentTrip ² | float | r/w |
| 0x18 | VoltageMeasure | float | r |
| 0x1c | CurrentMeasure | float | r |
| 0x20 | VoltageBounds / VoltageIlkMaxSet ³ | float | r/w |
| 0x24 | CurrentBounds / CurrentIlkMaxSet ⁴ | float | r/w |
| 0x28 | VoltageIlkMinSet | float | r/w |
| 0x2c | CurrentIlkMinSet | float | r/w |
| 0x30 | VoltageNominal / VoltageMaxSet ⁵ | float | r/(w) |
| 0x34 | CurrentNominal / CurrentMaxSet ⁵ | float | r/(w) |

2 when KilEnable=active

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

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

5 can be written in mode ModuleStatus IsStop = 1

Group data

Fixed Groups

| Offset Bytes (rel. to BA) | Name | Data type | Access |
|---------------------------|--|-----------|--------|
| 0x00d0 | SetVoltageAllChannels | float | r/w |
| 0x00d4 | SetCurrentAllChannels | float | r/w |
| 0x00d8 | SetVoltageBoundsAllChannels | float | r/w |
| 0x00dc | SetCurrentBoundsAllChannels | float | r/w |
| 0x00e0 | SetEmergencyAllChannels | uint32 | r/w |
| 0x00e4 | SetOnOffAllChannels | uint32 | r/w |
| 0x00e8 | SetVoltageIlkMinSetAllChannels | float | r/w |
| 0x00ec | SetCurrentIlkMinSetAllChannels | float | r/w |

Variable Groups

| Offset Bytes (rel. to BA) | Name | |
|---------------------------|------------|-------------------|
| 0x0700 | GrAddr[0] | begin of group 0 |
| 0x0708 | GrAddr[1] | begin of group 1 |
| 0x0710 | GrAddr[2] | begin of group 2 |
| 0x0718 | GrAddr[3] | begin of group 3 |
| 0x0720 | GrAddr[4] | begin of group 4 |
| 0x0728 | GrAddr[5] | begin of group 5 |
| 0x0730 | GrAddr[6] | begin of group 6 |
| 0x0738 | GrAddr[7] | begin of group 7 |
| 0x0740 | GrAddr[8] | begin of group 8 |
| ... | ... | ... |
| 0x07f8 | GrAddr[31] | begin of group 31 |

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

| Offset Bytes (rel. to BA) | Name | Data type | Access |
|---------------------------|---------------------------|-----------|--------|
| 0x0700 | VoltageNominal channel 0 | float | r |
| 0x0704 | CurrentNominal channel 0 | float | r |
| 0x0708 | VoltageNominal channel 1 | float | r |
| 0x070C | CurrentNominal channel 1 | float | r |
| | | | |
| 0x07b8 | VoltageNominal channel 23 | float | r |
| 0x07bC | CurrentNominal channel 23 | 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 |
|---------------------------|------------------------|-----------|--------|
| 0x00a0 | NewBaseAddress | uint16 | r/w |
| 0x00a2 | NewBaseAddressXor | uint16 | r/w |
| 0x00a4 | OldBaseAddress | uint16 | R |
| 0x00a6 | OldBaseAddressXor | uint16 | R |
| 0x00a8 | NewBaseAddressAccepted | uint16 | R |
| 0x00aa | NewBaseAddressAccepted | uint16 | R |
| 0x00ac | SpecialControlStatus | uint32 | R |
| 0x00b0 | SpecialControlCommand | uint32 | r/w |

3.2 Details to the memory space

3.2.1 Module registers

ModuleStatus

| Offset Bytes (rel. to BA) | Name | Data type | Access |
|------------------------------|--------------|-----------|--------|
| 0x0000 | ModuleStatus | uint32 | r |

ModuleStatus2 (reservation)

| Bit15 | Bit14 | Bit13 | Bit12 | Bit11 | Bit10 | Bit9 | Bit8 | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|
| res | res | res | res | res | res | res | res | res | res | res | res | res | res | res | res |

ModuleStatus1

| 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.

ModuleEventStatus

ModuleEventStatus

| Offset Bytes (rel. to BA) | Name | Data type | Access |
|------------------------------|-------------------|-----------|--------|
| 0x0004 | ModuleEventStatus | uint32 | r/w |

ModuleEventStatus2 (reservation)

| Bit15 | Bit14 | Bit13 | Bit12 | Bit11 | Bit10 | Bit9 | Bit8 | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|
| res | res | res | res | res | res | res | res | res | res | res | res | res | res | res | res |

ModuleEventStatus1

| 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 |
|------------------------------|-----------------|-----------|--------|
| 0x0008 | ModuleEventMask | uint32 | r/w |

ModuleEventMask1

| 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.

ModuleControl

| Offset Bytes (rel. to BA) | Name | Data type | Access |
|------------------------------|---------------|-----------|--------|
| 0x000c | ModuleControl | uint3216 | r/w |

ModuleControl2

| Bit15 | Bit14 | Bit13 | Bit12 | Bit11 | Bit10 | Bit9 | Bit8 | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|
| res | res | res | res | res | res | res | res | res | res | res | res | res | res | res | res |

ModuleControl1

| 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 | doRCV | 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 be 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.

ModuleEventChannelStatus

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

ModuleEventChannelStatus2

| Bit15 | Bit14 | Bit13 | Bit12 | Bit11 | Bit10 | Bit9 | Bit8 | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|
| res | res | res | res | res | res | res | res | CH23 | CH22 | CH21 | CH20 | CH19 | CH18 | CH17 | CH16 |

ModuleEventChannelStatus1

| Bit15 | Bit14 | Bit13 | Bit12 | Bit11 | Bit10 | Bit9 | Bit8 | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|
| CH15 | CH14 | CH13 | CH12 | 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.

$$CH_n = \text{EventStatus}[n] \& \text{EventMask}[n]$$

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 |
|------------------------------|------------------------|-----------|--------|
| 0x0014 | ModuleEventChannelMask | uint32 | r/w |

ModuleEventChannelMask2

| Bit15 | Bit14 | Bit13 | Bit12 | Bit11 | Bit10 | Bit9 | Bit8 | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|
| res | res | res | res | res | res | res | res | CH23 | CH22 | CH21 | CH20 | CH19 | CH18 | CH17 | CH16 |

ModuleEventChannelMask1

| Bit15 | Bit14 | Bit13 | Bit12 | Bit11 | Bit10 | Bit9 | Bit8 | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|
| CH15 | CH14 | CH13 | CH12 | 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 |
|------------------------------|------------------------|-----------|--------|
| 0x0018 | ModuleEventGroupStatus | uint32 | r/w |

ModuleEventGroupStatus2

| | | | | | | | | | | | | | | | |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 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 |

ModuleEventGroupStatus1

| | | | | | | | | | | | | | | | |
|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|
| 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 |
|------------------------------|----------------------|-----------|--------|
| 0x001c | ModuleEventGroupMask | uint32 | r/w |

ModuleEventGroupMask2

| | | | | | | | | | | | | | | | |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 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 |

ModuleEventGroupMask1

| | | | | | | | | | | | | | | | |
|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|
| 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 |
|------------------------------|------------------|-----------|--------|
| 0x0020 | 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 (reservation)

| Offset Bytes (rel. to BA) | Name | Data type | Access |
|------------------------------|------------------|-----------|--------|
| 0x0024 | CurrentRampSpeed | float | r/w |

not realized in VDS x0x

VoltageMax

| Offset Bytes (rel. to BA) | Name | Data type | Access |
|------------------------------|------------|-----------|--------|
| 0x0028 | 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.

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

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

CurrentMax

| Offset Bytes (rel. to BA) | Name | Data type | Access |
|------------------------------|------------|-----------|--------|
| 0x002c | 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.

SerialNumber

| Offset Bytes (rel. to BA) | Name | Data type | Access |
|------------------------------|--------------|-----------|--------|
| 0x0030 | SerialNumber | uint32 | r |

The Serial number of the module as long integer value.

FirmwareRelease

| Offset Bytes (rel. to BA) | Name | Data type | Access |
|------------------------------|-----------------|-----------|--------|
| 0x0034 | 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 |
|------------------------------|----------------|-----------|--------|
| 0x0038 | PlacedChannels | uint32 | r |

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

For example, a fully equipped 8 channel module VDS 08x has PlacedChannels = 0x00000f and a fully equipped 24 channel module VDS 18x has PlacedChannels = 0x00ffff.

ChannelNumber_DeviceClass

| Offset Bytes (rel. to BA) | Name | Data type | Access |
|------------------------------|---------------------------|-----------|--------|
| 0x003C | ChannelNumber DeviceClass | uint16[2] | r |

ChannelNumber: 8 or 24

It exists two kinds of VDS modules. One kind in single width for a number of 8 HV channels and a second one in double width for a number of 24 HV channels.

DeviceClass 29

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

For VDS x0x this value is 29 (0x1d).

SupplyP5

| Offset Bytes (rel. to BA) | Name | Data type | Access |
|------------------------------|----------|-----------|--------|
| 0x0040 | 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 |
|------------------------------|-----------|-----------|--------|
| 0x0044 | 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 |
|------------------------------|-----------|-----------|--------|
| 0x0048 | 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 |
|------------------------------|-------------|-----------|--------|
| 0x004c | Temperature | float | r |

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

RestartTimeAfterReloadSetValues

| Offset Bytes (rel. to BA) | Name | Data type | Access |
|------------------------------|---------------------------------|-----------|--------|
| 0x0050 | RestartTimeAfterRecallSetValues | uint32 | 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 |
|------------------------------|---------------------|-----------|--------|
| 0x0054 | ADCSamplesPerSecond | uint32 | 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*1/SPS multiplied with the number of channels for device.

Factory settings: 500 SPS

DigitalFilter

| Offset Bytes (rel. to BA) | Name | Data type | Access |
|------------------------------|----------------------|-----------|--------|
| 0x0058 | DigitalFilter | uint32 | 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"}.

3.2.2 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 |
|-------------------------------|---------------|-----------|--------|
| 0x00 | ChannelStatus | uint32 | R |

ChannelStatus2

| Bit15 | Bit14 | Bit13 | Bit12 | Bit11 | Bit10 | Bit9 | Bit8 | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|
| res | res | res | res | res | res | res | res | res | res | res | res | res | res | res | res |

ChannelStatus1

| | | | | | | | | | | | | | | | |
|--------|--------|--------|--------|---------|---------|------|------|------|------|--------|--------|------|--------|------|------|
| 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:

```

if (| Vmeas – Vset | > Vbounds)           isVoltageBoundsExceeded =1;
if (| Imeas – Iset | > Ibounds)           isCurrentBoundsExceeded =1;

```

ChannelEventStatus

| Offset Bytes (rel. to ChAddr) | Name | Data type | Access |
|-------------------------------|--------------------|-----------|--------|
| 0x04 | ChannelEventStatus | uint32 | r/w |

ChannelEventStatus2

| Bit15 | Bit14 | Bit13 | Bit12 | Bit11 | Bit10 | Bit9 | Bit8 | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|
| res | res | res | res | res | res | res | res | res | res | res | res | res | res | res | res |

ChannelEventStatus1

| 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 |
|-------------------------------|------------------|-----------|--------|
| 0x08 | ChannelEventMask | uint32 | r/w |

ChannelEventMask2

| Bit15 | Bit14 | Bit13 | Bit12 | Bit11 | Bit10 | Bit9 | Bit8 | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|
| res | res | res | res | res | res | res | res | res | res | res | res | res | res | res | res |

ChannelEventMask1

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

| | | |
|----------|----------------------------|--|
| MEVLM | 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 | |

ChannelControl

| Offset Bytes (rel. to ChAddr) | Name | Data type | Access |
|-------------------------------|----------------|-----------|--------|
| 0x0c | ChannelControl | uint16 | r/w |

ChannelControl2

| Bit15 | Bit14 | Bit13 | Bit12 | Bit11 | Bit10 | Bit9 | Bit8 | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|
| res | res | res | res | res | res | res | res | res | res | res | res | res | res | res | res |

ChannelControl1

| 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, VoltageMinIlkSet, CurrentBounds and CurrentMinIlkSet 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)

VoltageIlkMaxSet ≤ VoltageMeasure ≤ VoltageIlkMaxSet

No event!

VoltageIlkMaxSet > VoltageMeasure or
VoltageMeasure > VoltageIlkMaxSet

IsVoltageBoundsExceeded(1), ModuleStatus
EventVoltageBounds(1), ModuleEventStatus

setAVBND(0)

VoltageSet-VoltageBounds ≤ VoltageMeasure ≤ VoltageSet+VoltageBounds No event!

VoltageSet-VoltageBounds > VoltageMeasure or
VoltageMeasure > VoltageSet+VoltageBounds

IsVoltageBoundsExceeded(1), ModuleStatus
EventVoltageBounds(1), ModuleEventStatus

setACBND(1)

CurrentIlkMaxSet ≤ CurrentMeasure ≤ CurrentIlkMaxSet

No event!

CurrentIlkMaxSet > CurrentMeasure or
CurrentMeasure > CurrentIlkMaxSet

IsCurrentBoundsExceeded(1), ModuleStatus
EventCurrentBounds(1), ModuleEventStatus

setACBND(0)

CurrentSet-CurrentBounds ≤ CurrentMeasure ≤ 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: VoltageIlkMaxSet=VoltageSet+VoltageBounds

 VoltageIlkMinSet=VoltageSet-VoltageBounds

Current: when ChannelStatus isCC(1)

 CurrentIlkMaxSet=CurrentSet+CurrentBounds

 CurrentIlkMinSet=CurrentSet-CurrentBounds

ChannelStatus isCC(0), ChannelStatus isON(1), ChannelStatus isRAMP(0)

 CurrentIlkMaxSet=CurrentMeasure+CurrentBounds

 CurrentIlkMinSet=CurrentMeasure-CurrentBounds

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

Voltage: VoltageBounds=(VoltageIlkMaxSet- VoltageIlkMinSet)/2

Current: CurrentBounds=(CurrentIlkMaxSet-CurrentIlkMinSet)/2

VoltageSet

| Offset Bytes (rel. to ChAddr) | Name | Data type | Access |
|-------------------------------|------------|-----------|--------------|
| 0x10 | VoltageSet | Vset | float r/w |

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.

CurrentTrip

| Offset Bytes (rel. to ChAddr) | Name | Data type | Access |
|-------------------------------|-------------|------------|--------------|
| 0x14 | CurrentTrip | Iset/Itrip | float r/w |

The value of CurrentTrip is used for a current compare by software. 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 |
|-------------------------------|----------------|-----------|--------------|
| 0x18 | 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 |
|-------------------------------|----------------|-----------|--------------|
| 0x1c | CurrentMeasure | Imeas | float r/w |

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

VoltageBounds

VoltageIlkMaxSet

| Offset Bytes (rel. to ChAddr) | Name | Data type | Access |
|-------------------------------|------------------|-------------|--------|
| 0x20 | VoltageBounds | Vbounds | float |
| 0x20 | VoltageIlkMaxSet | VIIlkMaxSet | float |

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:

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

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

By the help of VoltageIlkMaxSet (VIIlkMaxSet) and VoltageIlkMinSet (VIIlkMinSet) there is defined a region around VoltageSet (Vset), where the actual values are interpreted as good. This region is defined as follows:

$$VIIlkMinSet \leq V_{meas} \leq VIIlkMaxSet$$

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

CurrentIlkMaxSet

| Offset Bytes (rel. to ChAddr) | Name | Data type | Access |
|-------------------------------|------------------|------------|--------|
| 0x24 | CurrentIlkMaxSet | IllkMaxSet | float |

CurrentIlkMaxSet:

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

$$IllkMinSet \leq I_{meas} \leq IllkMaxSet$$

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

VoltageIlkMinSet

| Offset Bytes (rel. to ChAddr) | Name | Data type | Access |
|-------------------------------|------------------|-------------|--------|
| 0x28 | VoltageIlkMinSet | VIIlkMinSet | float |

see VoltageIlkMaxSet above

CurrentIlkMinSet

| Offset Bytes (rel. to ChAddr) | Name | Data type | Access |
|-------------------------------|------------------|------------|--------|
| 0x2c | CurrentIlkMinSet | IllkMinSet | float |

see CurrentIlkMaxSet above

VoltageNominal / VoltageMaxSet

| Offset Bytes (rel. to ChAddr) | Name | Data type | Access |
|-------------------------------|--------------------------------|-----------|----------------|
| 0x30 | 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 |
|-------------------------------|--------------------------------|-----------|----------------|
| 0x34 | 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)

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

3.2.2.1 Fix Groups

The functions and characteristics of the groups are fix defined.

SetVoltageAllChannels

| Offset Bytes (rel. to BA) | Name | Data type | Access |
|---------------------------|-----------------------|-----------|--------|
| 0x00d0 | 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 |
|---------------------------|-----------------------|-----------|--------|
| 0x00d4 | 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 |
|---------------------------|-----------------------------|-----------|--------|
| 0x00d8 | 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.

SetCurrentIlkMaxSetAllChannels

| Offset Bytes (rel. to BA) | Name | Data type | Access |
|---------------------------|--------------------------------|-----------|--------|
| 0x00dc | SetCurrentIlkMaxSetAllChannels | float | r/w |

The value of the SetCurrentIlkMinSetAllChannels 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 |
|---------------------------|-------------------------|-----------|--------|
| 0x00e0 | 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 |
|---------------------------|---------------------|-----------|--------|
| 0x00e4 | 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 |

SetVoltageIlkMinSetAllChannels

| Offset Bytes (rel. to BA) | Name | Data type | Access |
|---------------------------|--------------------------------|-----------|--------|
| 0x00e8 | SetVoltageIlkMinSetAllChannels | float | r/w |

The value of the SetVoltageIlkMaxSetAllChannels 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.

SetCurrentIlkMinSetAllChannels

| Offset Bytes (rel. to BA) | Name | Data type | Access |
|---------------------------|--------------------------------|-----------|--------|
| 0x00ec | SetCurrentIlkMinSetAllChannels | float | r/w |

The value of the SetCurrentIlkMinSetAllChannels 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.

3.2.2.2 Variable Groups

3.2.2.2.1 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[2] | r/w |
| 4 | | TypeSet2 | | | | | | | | | | | | uint16 | r/w |
| 6 | | TypeSet1 | | | | | | | | | | | | uint16 | r/w |

MemberList2:

| | | | | | | | | | | | | | | | |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Bit31 | Bit30 | Bit29 | Bit28 | Bit27 | Bit26 | Bit25 | Bit24 | Bit23 | Bit22 | Bit21 | Bit20 | Bit19 | Bit18 | Bit17 | Bit16 |
| res | CH23 | CH22 | CH21 | CH20 | CH19 | CH18 | CH17 | CH16 |

MemberList1:

| | | | | | | | | | | | | | | | |
|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|
| Bit15 | Bit14 | Bit13 | Bit12 | Bit11 | Bit10 | Bit9 | Bit8 | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
| CH15 | CH14 | CH13 | CH12 | CH11 | CH10 | CH9 | CH8 | CH7 | CH6 | CH5 | CH4 | CH3 | CH2 | CH1 | CHO |

TypeSet2:

| | | | | | | | | | | | | | | | |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Bit31 | Bit30 | Bit29 | Bit28 | Bit27 | Bit26 | Bit25 | Bit24 | Bit23 | Bit22 | Bit21 | Bit20 | Bit19 | Bit18 | Bit17 | Bit16 |
| res | MCH4 | MCH3 | MCH2 | MCH1 | MCH0 |

TypeSet1:

| | | | | | | | | | | | | | | | |
|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|
| 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 | res | res | res | res |

| | | | |
|-------|-------|--------------|-------------------------------|
| 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 | SetVIIkMinSet | Copy VIIkMinSet 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 (in preparation) |

| | | | | | | |
|------|------|------|------|------|-------|------------------------------------|
| MCH4 | MCH3 | MCH2 | MCH1 | MCH0 | Value | |
| 0 | 0 | 0 | 0 | 0 | 0 | 1: Channel 0 is MasterChannel MCH |
| 0 | 0 | 0 | 0 | 1 | 1 | 1: Channel 1 is MasterChannel MCH |
| ... | ... | ... | ... | ... | ... | ... |
| 0 | 0 | 0 | 1 | 1 | 3 | 1: Channel 3 ist MasterChannel MCH |

3.2.2.2.2 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[2] | r/w |
| 4 | TypeStatus2 | uint16 | r/w |
| 6 | TypeStatus1 | uint16 | r/w |

ChannelStatusList2:

| | | | | | | | | | | | | | | | |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Bit31 | Bit30 | Bit29 | Bit28 | Bit27 | Bit26 | Bit25 | Bit24 | Bit23 | Bit22 | Bit21 | Bit20 | Bit19 | Bit18 | Bit17 | Bit16 |
| res | CH23 | CH22 | CH21 | CH20 | CH19 | CH18 | CH17 | CH16 |

ChannelStatusList1:

| | | | | | | | | | | | | | | | |
|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|
| Bit15 | Bit14 | Bit13 | Bit12 | Bit11 | Bit10 | Bit9 | Bit8 | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
| CH15 | CH14 | CH13 | CH12 | CH11 | CH10 | CH9 | CH8 | CH7 | CH6 | CH5 | CH4 | CH3 | CH2 | CH1 | CH0 |

TypeStatus2:

| | | | | | | | | | | | | | | | |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Bit31 | Bit30 | Bit29 | Bit28 | Bit27 | Bit26 | Bit25 | Bit24 | Bit23 | Bit22 | Bit21 | Bit20 | Bit19 | Bit18 | Bit17 | Bit16 |
| res |

TypeStatus1:

| | | | | | | | | | | | | | | | |
|-------|-------|-------|-------|-------|-------|------|------|-------|-------|-------|-------|------|------|------|------|
| 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) |

3.2.2.2.3 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 | uint32 | r/w |
| 4 | TypeMonitoring2 | uint16 | r/w |
| 6 | TypeMonitoring1 | uint16 | r/w |

MemberList2:

| Bit31 | Bit30 | Bit29 | Bit28 | Bit27 | Bit26 | Bit25 | Bit24 | Bit23 | Bit22 | Bit21 | Bit20 | Bit19 | Bit18 | Bit17 | Bit16 |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| res | CH23 | CH22 | CH21 | CH20 | CH19 | CH18 | CH17 | CH16 |

MemberList1:

| Bit15 | Bit14 | Bit13 | Bit12 | Bit11 | Bit10 | Bit9 | Bit8 | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|
| CH15 | CH14 | CH13 | CH12 | CH11 | CH10 | CH9 | CH8 | CH7 | CH6 | CH5 | CH4 | CH3 | CH2 | CH1 | CH0 |

TypeMonitoring2:

| Bit31 | Bit30 | Bit29 | Bit28 | Bit27 | Bit26 | Bit25 | Bit24 | Bit23 | Bit22 | Bit21 | Bit20 | Bit19 | Bit18 | Bit17 | Bit16 |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| res |

TypeMonitoring1:

| 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 |

3.2.2.2.4 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[2] | r/w |
| 4 | TypeTimeOut2 | uint16 | r/w |
| 2 | TypeTimeOut1 | uint16 | r/w |

MemberList1:

| Bit31 | Bit30 | Bit29 | Bit28 | Bit27 | Bit26 | Bit25 | Bit24 | Bit23 | Bit22 | Bit21 | Bit20 | Bit19 | Bit18 | Bit17 | Bit16 |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| res | CH23 | CH22 | CH21 | CH20 | CH19 | CH18 | CH17 | CH16 |

MemberList2:

| Bit15 | Bit14 | Bit13 | Bit12 | Bit11 | Bit10 | Bit9 | Bit8 | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|
| CH15 | CH14 | CH13 | CH12 | CH11 | CH10 | CH9 | CH8 | CH7 | CH6 | CH5 | CH4 | CH3 | CH2 | CH1 | CH0 |

TypeTimeOut1:

| Bit31 | Bit30 | Bit29 | Bit28 | Bit27 | Bit26 | Bit25 | Bit24 | Bit23 | Bit22 | Bit21 | Bit20 | Bit19 | Bit18 | Bit17 | Bit16 |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| TOT15 | TOT14 | TOT13 | TOT12 | TOT11 | TOT10 | TOT9 | TOT8 | TOT7 | TOT6 | TOT5 | TOT4 | TOT3 | TOT2 | TOT1 | TOT0 |

TOT[15..0]: Binary coded Timeout-time in ms (0..65535ms)

TypeTimeOut2:

| Bit15 | Bit14 | Bit13 | Bit12 | Bit11 | Bit10 | Bit9 | Bit8 | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|
| TYPE1 | TYPE0 | ACT1 | ACT0 | res | res | res | res | res | res | res | res | res | res | res | res |

| TYPE1 | TYPE0 | Value | |
|-------|-------|------------------|--|
| 1 | 1 | TypeOutGroupType | 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 |

3.3 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.

3.3.1 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] =

$$\begin{aligned}
 & (\text{EventVoltageLimit}[n] \text{ AND } \text{MaskEventVoltageLimit}[n]) \text{ OR} \\
 & (\text{EventCurrentLimit}[n] \text{ AND } \text{MaskEventCurrentLimit}[n]) \text{ OR} \\
 & (\text{EventTrip}[n] \text{ AND } \text{MaskEventTrip}[n]) \text{ OR} \\
 & (\text{EventExtInhibit}[n] \text{ AND } \text{MaskEventExtInhibit}[n]) \text{ OR} \\
 & (\text{EventVoltageBounds}[n] \text{ AND } \text{MaskEventVoltageBounds}[n]) \text{ OR} \\
 & (\text{EventCurrentBounds}[n] \text{ AND } \text{MaskEventCurrentBounds}[n]) \text{ OR} \\
 & (\text{EventControlledVoltage}[n] \text{ AND } \text{MaskEventControlledVoltage}[n]) \text{ OR} \\
 & (\text{EventEmergency}[n] \text{ AND } \text{MaskEventEmergency}[n]) \text{ OR} \\
 & (\text{EventEndOfRamp}[n] \text{ AND } \text{MaskEventEndOfRamp}[n]) \text{ OR} \\
 & (\text{EventOnToOff}[n] \text{ AND } \text{MaskEventOnToOff}[n]) \text{ OR} \\
 & (\text{EventInputError}[n] \text{ AND } \text{MaskEventInputError}[n])
 \end{aligned}$$

where is:

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

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

MaskEventVoltageLimit[n]: bit MaskEventVoltageLimit of register ChannelEventMask of thr 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])

3.3.2 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])

3.3.3 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 MaskEventRestart) OR

3.3.4 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

3.4 Special registers

3.4.1 Setting of Basis Address

| Offset Bytes (rel. to BA) | Name | Data type | Access |
|---------------------------|------------------------|-----------|--------|
| 0x00A0 | NewBaseAddress | uint16 | r/w |
| 0x00A2 | NewBaseAddressXor | uint16 | r/w |
| 0x00A4 | OldBaseAddress | uint16 | r |
| 0x00A8 | 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 2kByte 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 = bbbbb000 00000000 (with b={0|1})
hexadezimal: BA = xy00 (with x={0..F}, y={0..8}).

The default value (factory setting and setting when started with jumper “ADR” on the topside of the board has been set) is BA=0x4000 (in bytes) resp. BA=0x2000 in words.

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 2kByte 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.

3.4.2 Special Control Register

| Offset Bytes (rel. to BA) | Name | Data type | Access |
|---------------------------|-----------------------|-----------|--------|
| 0x00ac | SpecialControlStatus | uint32 | r |
| 0x00b0 | SpecialControlCommand | uint32 | r/w |
| 0x00b4 | Length Address | uint32 | r/w |
| 0x00b8 | RDATA | uint32 | r |
| 0x00bc | WDATA | uint32 | w |

The registers "SpecialControlStatus", "SpecialControlCommand" and following are reserved for maintenance and service purposes such as live insertion. Their usage is explained in a separate manual.