0

MPOD
SNMP protocol manual

v1.0

December 11, 2019

Analog Flavor EURL au capital de 5000 Euro
SIREN 814 075 727

3 bis, rue des Engenières, 38360 Sassenage, France.
Telephone: +33 6 28 34 91 82
support@analogflavor.com
www.analogflavor.com

Table of Contents

CHAPTER 1 INTRODUCTION TO THE MPOD SNMP PROTOCOL

Wiener MPOD crates and all inserted modules and their channels are controlled over a network connection and the SNMP protocol. SNMP stands for Simple Network Management Protocol. It allows to communicate between the agent, which runs on the MPOD controller and one or several connected managers or clients.

The SNMP protocol is best exploited by using existing software libraries or modules and command-line tools. On overview over the available tools and libraries will be given below.

The MPOD is controlled by SNMP variables. Each variable is identified by their **object identifie**r or **OID**. The SNMP protocol defines requests to read and write variables:

- The GetRequest allows to read a single variable with it's OID.
- The GetNextRequest can be used to read a variables when only the previous OID is known. This allows to iterate over all SNMP variables.
- The GetBulkRequest is used to read several variables at once.
- The SetRequest is used to write a new value to a single variable.

Some of these variables like the crates fan speed are scalar variables. They contain a single value. Other variables, such as the channel voltages are vector variables. A value is defined for each channel.

The variable OIDs such as iso.3.6.1.4.1.19947.1.1.1.0 for the crates main switch are defined by a vector of integers. For all control interfaces based on the net-snmp library their use can be simplified by using a **management information base** file or **MIB** file. This file allows the net-snmp library to translate OIDs to human readable names. All following examples show how to read or write SNMP variables with or without MIB file.

This manual explains how to manage an MPOD crate, the inserted modules and their channels be reading and writing SNMP variables. Depending on the connected hardware and its version the available variables might differ. An overview over all available variables can be obtained by using the snmpwalk command line tool. The usage of this tool will be explained below. It is particularly useful for debugging control software and if asking for support.

To communicate with a Wiener MPOD it's IP address needs to be known. This IP address can be configured by using "MUSEcontrol" as explained below. The access to variables is restricted by using SNMP community names. This community name is passed as argument to all kind of requests. With default settings the community name "guru" allows full access to all SNMP variables. When using the community "public" only read-access is possible.

Additional information on Wiener MPOD crates and how to control them can be found in the following documents:

- The MPOD user manual is essential for all aspects of implementing control software. Please download the latest version from the Wiener web site.

- The user manuals of the inserted modules are required to fully understand how the modules and their channels can be controlled. Please download the latest version from the Wiener web site for LV modules and from the iseg web site for HV modules.
- Another important file is the Wiener MIB file called WIENER-CRATE-MIB.txt. The latest version can also be found on the Wiener web site.

1.1 Configuring the MPOD crate using MUSEcontrol

Before controlling an MPOD it must be configured using MUSEcontrol software over an USB connection.

- The IP address of the MPOD controller must be configured.
- The crate controller's "interlock" functionality is triggered by the crate controller's pins referred to as "interlock pins" in the MPOD user's manual. If the interlock gets active, all channels of all modules are ramped down. In order to use the controller's interlock functionality, the crate controller must be configured appropriately. It isn't possible to configure this functionality over SNMP. Instead the MUSEcontrol software must be used. To enable interlock, the "Ignore hardware interlock" check box must be unchecked. In the picture below, the flag is checked. As a consequence the interlock functionality is disabled. While the interlock is active, the channels' and the controller's "inhibit" flags are set. In addition to the controller's interlock, some modules might have "inhibit" pins on their front panels. The modules' inhibit functionality can be configured over SNMP.
- The channel property values for "Config Max Sense Voltage", "Config Max Terminal Voltage" and "Config Max Current" are read-only in the SNMP protocol. They can only be set using MUSEcontrol software. These values are only relevant for low-voltage modules.
- The values for the temperature limit for the channel's temperature supervision can only be set using MUSEcontrol software.
- The "Communication Timeout" value must be set using MUSEcontrol software. The behavior can be set in Easy LV | HV.

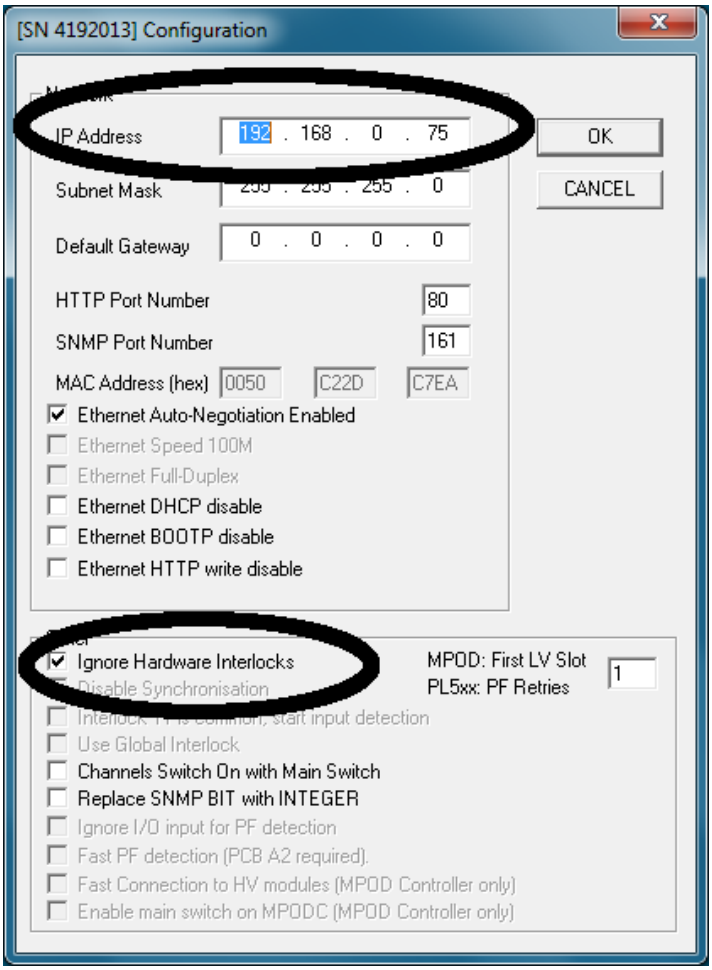

Illustration 1: MUSE control software is required to define the MPOD controller's IP and to enable or disable the crate controller's interlock. Here the flag is checked. As a consequence the interlock functionality is disabled.

1.2 List of available SNMP variables

The below table lists all SNMP variables that will be handled in this document. Depending on the connected hardware some variables might not be available. Some variables defined in Wiener MIB file are not handled here.

Some variables apply only to high voltage hardware. This is denoted by a "HV" in the "Applies to" column. Other variables only concern low voltage "LV". Some variables concern both but have different meanings for LV and HV. These variables are denoted by "LV (HV)". All other variables apply to LV and HV in a similar way.

The below table uses placeholders:

- <base>: the base OID that is common to all Wiener OIDs. It's value "iso.3.6.1.4.1.19947" must be prefixed to all concerned IODs.
- <channel>: this placeholder must be replaced by the integer values 1 … 2000, not all values are used. The indexes 1 … 100 are used for the channels of the module in slot 0, the indexes 101 … 200 are used for the channels of the module in slot 1 etc. More information can be found in the chapter on crate structure.
- <slot>: this placeholder must be replaced by the slot index of the module. This is an integer vale from 1 … 10, not all values are used. More information can be found in the chapter on crate structure.
- <community>: the integer community index takes values from 1 … 4.
- <aux>: the aux index is an integer value from 1 … 8.

System Variables	**OID**	**Applies to**	**Access**	**Type**
sysDescr	iso.3.6.1.2.1.1.1.0	LV HV	R	String
sysObjectID	iso.3.6.1.2.1.1.2.0	LV HV	R	OID
Crate Variables				
sysMainSwitch	<base>.1.1.1.0	LV HV	R/W	Integer
sysStatus	<base>.1.1.2.0	LV HV	R/W	Integer
sysHardwareReset	<base>.1.1.4.0	LV HV	R/W	Integer
snmpCommunityName	<base>.1.5.1.1.1.2.<community>	LV HV	R/W	String
psSerialNumber	<base>.1.6.2.0	LV HV	R	String
psOperatingTime	<base>.1.6.3.0	LV HV	R	Integer
psAuxiliaryNumber	<base>.1.6.4.0	LV HV	R	Integer
psAuxiliaryIndex	<base>.1.6.5.1.1.<aux>	LV HV	R	Integer
psAuxiliaryMeasurementVoltage	<base>.1.6.5.1.3.<aux>	LV HV	R	Float
psAuxiliaryMeasurementCurrent	<base>.1.6.5.1.4.<aux>	LV HV	R	Float
fanNominalSpeed	<base>.1.7.6.0	LV HV	R/W	Integer
Module Variables				
moduleNumber	<base>.1.3.5.0	LV HV	R	Integer
moduleDescription	<base>.1.3.6.1.2.<slot>	LV HV	R	String
moduleAuxiliaryMeasurementVoltage0	<base>.1.3.6.1.3.1.<slot>	HV	R	Float
moduleAuxiliaryMeasurementVoltage1	<base>.1.3.6.1.3.2.<slot>	HV	R	Float
moduleHardwareLimitVoltage	<base>.1.3.6.1.4.<slot>	HV	R	Float
moduleHardwareLimitCurrent	<base>.1.3.6.1.5.<slot>	HV	R	Float
moduleRampSpeedVoltage	<base>.1.3.6.1.6.<slot>	HV	R/W	Float
moduleRampSpeedCurrent	<base>.1.3.6.1.7.<slot>	HV	R/W	Float

moduleStatus	<base>.1.3.6.1.8.<slot>	HV	R	Bit String
moduleEventStatus	<base>.1.3.6.1.9.<slot>	HV	R	Bit String
moduleEventChannelStatus	<base>.1.3.6.1.10.<slot>	HV	R	Bit String
moduleDoClear	<base>.1.3.6.1.11.<slot>	HV	R/W	Integer
moduleAuxiliaryMeasurementTemperature0	<base>.1.3.6.1.12.1.<slot>	HV	R	Float
moduleAuxiliaryMeasurementTemperature1	<base>.1.3.6.1.12.2.<slot>	HV	R	Float
moduleAuxiliaryMeasurementTemperature2	<base>.1.3.6.1.12.3.<slot>	HV	R	Float
moduleAuxiliaryMeasurementTemperature3	<base>.1.3.6.1.12.4.<slot>	HV	R	Float
Channel Variables				
outputNumber	<base>.1.3.1.0	LV HV	R	Integer
outputName	<base>.1.3.2.1.2.<channel>	LV HV	R	String
outputGroup	<base>.1.3.2.1.3.<channel>	LV	R/W	Integer
outputStatus	<base>.1.3.2.1.4.<channel>	LV HV	R	Bit String
outputMeasurementSenseVoltage	<base>.1.3.2.1.5.<channel>	LV	R	Float
outputMeasurementTerminalVoltage	<base>.1.3.2.1.6.<channel>	LV HV	R	Float
outputMeasurementCurrent	<base>.1.3.2.1.7.<channel>	LV HV	R	Float
outputMeasurementTemperature	<base>.1.3.2.1.8.<channel>	LV	R	Integer
outputSwitch	<base>.1.3.2.1.9.<channel>	LV HV	R/W	Integer
outputVoltage	<base>.1.3.2.1.10.<channel>	LV HV	R/W	Float
outputCurrent	<base>.1.3.2.1.12.<channel>	LV HV	R/W	Float
outputVoltageRiseRate	<base>.1.3.2.1.13.<channel>	LV	R/W	Float
outputVoltageFallRate	<base>.1.3.2.1.14.<channel>	LV	R/W	Float
outputSupervisionBehavior	<base>.1.3.2.1.15.<channel>	LV HV	R/W	Integer
outputSupervisionMinSenseVoltage	<base>.1.3.2.1.16.<channel>	LV	R/W	Float
outputSupervisionMaxSenseVoltage	<base>.1.3.2.1.17.<channel>	LV	R/W	Float
outputSupervisionMaxTerminalVoltage	<base>.1.3.2.1.18.<channel>	LV	R/W	Float
outputSupervisionMaxCurrent	<base>.1.3.2.1.19.<channel>	LV	R/W	Float
outputConfigMaxSenseVoltage	<base>.1.3.2.1.21.<channel>	LV	R	Float
outputConfigMaxTerminalVoltage	<base>.1.3.2.1.22.<channel>	LV (HV)	R	Float
outputConfigMaxCurrent	<base>.1.3.2.1.23.<channel>	LV (HV)	R	Float
outputSupervisionMaxPower	<base>.1.3.2.1.24.<channel>	LV	R	Float
outputCurrentRiseRate	<base>.1.3.2.1.25.<channel>	LV	R/W	Float
outputCurrentFallRate	<base>.1.3.2.1.26.<channel>	LV	R/W	Float
outputTripTimeMaxCurrent	<base>.1.3.2.1.27.<channel>	HV	R/W	Integer
outputUserConfig	<base>.1.3.2.1.37.<channel>	LV	R/W	Integer
Group Variables				
groupsNumber	<base>.1.3.3.0	LV HV	R	Integer
groupsSwitch	<base>.1.3.4.1.9.<group>	LV HV	R/W	Integer

CHAPTER 2 HOW TO ACCESS THE SNMP VARIABLES

MPOD control is realized by reading and writing SNMP variables. Several libraries, modules and tools are available to perform these tasks. This chapter gives an overview over several available methods.

The examples in this documentation use net-snmp command line tools. They can easily be adapted for other access methods as shown in the “Additional Examples” chapter.

To control MPOD crates using the SNMP protocol it’s IP and the appropriate community name have to be known. All access methods using the net-snmp library or command line tools make usage of the Wiener MIB file that can be downloaded from the Wiener web site.

The following access methods are explained here:

- The net-snmp library provides the command line tools
  - snmpget
  - snmpset
  - snmpwalk
- Software written in C or C++ can make use of the following libraries:
  - the net-snmp library has a C API. It can be used on Windows, Linux and MacOS.
  - the WinSNMP library is only available on Windows.
- Python software can use the PySNMP module.
- Software based on TANGO-Controls can make use of a TANGO device server that converts TANGO commands to SNMP.

2.1 Using net-snmp command line tools

The net-snmp library can be downloaded as source code from http://www.net-snmp.org. Compiled versions are available for various platforms.

The examples of this document are based on the command line tools snmpget, snmpset and snmpwalk that are provided with net-snmp.

All 3 command line tools require similar arguments:

- Communicating with the Wiener MPOD over a network connection requires the MPOD’s IP address. In the following examples we will presume the IP address is 192.168.0.103.
- The access rights are granted by specifying a SNMP community with corresponding rights. Use the command-line option “-c guru” to use the SNMP community “guru”, which by grants full access rights if the community name hasn’t been changed.
- The version of the used SNMP protocol needs to be specified either as -v1 (version 1) or -v2c (version 2c).

- If the WIENER MIB file is used this must be specified on the command line. Add "-m +WIENER-CRATE-MIB" when invoking the 3 command line tools.
- If the WIENER-CRATE-MIB.txt file is not found, it's path can be specified by defining the "MIBDIRS" variable accordingly. Alternatively the option "-M <paths>" can be added to the command line options. The Wiener MIB refers to other MIB files. The argument <paths> must contain the list of paths to the Wiener MIB file and all MIB files it refers to. The paths must be separated by a colon.

2.1.1 Snmpwalk

The snmpwalk command is very useful to understand which variables are currently available and for debugging a control software. The command iterates over all variables matching <variable name> and prints their name and value. The arguments <variable name or OID> and <variable OID> can be a partial.

Syntax:

> snmpwalk -v2c -m +WIENER-CRATE-MIB 192.168.0.103 -c guru <variable name or OID>

> snmpwalk -v2c 192.168.0.103 -c guru <variable OID>

Remark: if the crate is switched off, module and crate variables aren't available.

Example: all available SNMP variables can be printed by replacing <variable name> by a dot

> snmpwalk -v2c -m +WIENER-CRATE-MIB 192.168.0.103 -c guru .

If the MIB file is found the list of variables is returned in human readable form. Otherwise OIDs are printed. The command

> snmpwalk -v2c 192.168.0.103 -c guru .

returns all variables as OID. The Wiener MIB file is not used for translating the OIDs. The commands

> snmpwalk -v2c -m +WIENER-CRATE-MIB 192.168.0.103 -c guru moduleStatus

> snmpwalk -v2c 192.168.0.103 -c guru iso.3.6.1.4.1.19947.1.3.6.1.8

return only the subset of variables defining the module status. If 3 modules are present and the crate is switched on, 3 values will be returned.

Remark: check the "Troubleshooting" section to find out how snmpwalk can be used efficiently.

Remark: the output of an snmpwalk command can help a lot when requesting support on your hardware.

2.1.2 Snmpget

The snmpget command allows to retrieve the value of a single SNMP variable. The syntax is similar to the snmpwalk command, but the arguments <variable name or OID> and <variable OID> must be precise.

Syntax:

> snmpget -v2c -m +WIENER-CRATE-MIB 192.168.0.103 -c guru <variable name or OID>

> snmpget -v2c 192.168.0.103 -c guru <variable OID>

Example: the commands

> snmpget -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103 sysMainSwitch.0

> snmpget -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.1.1.0

> snmpget -v2c 192.168.0.103 -c guru iso.3.6.1.4.1.19947.1.1.1.0

return the status of the main switch. The integer value 0 indicates that the crate is switched off, 1 indicates that is is switched on.

2.1.3 Snmpset

The snmpset command allows to set the value of a single SNMP variable. The syntax is similar to the snmpset command. The 2 additional arguments specify variable type and the new value.

Syntax:

> snmpset -v2c -m +WIENER-CRATE-MIB 192.168.0.103 -c guru <variable name or OID> <type> <value>

> snmpset -v2c 192.168.0.103 -c guru <variable OID> <type> <value>

the placeholder <type> must be replaced by “i” for an integer, by “s” for a string value and by “F” for a floating point value.

Example: the commands

> snmpset -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103 sysMainSwitch.0 i 1

> snmpset -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.1.1.0 i 1

> snmpset -v2c 192.168.0.103 -c guru iso.3.6.1.4.1.19947.1.1.1.0 i 1

switch the crate on.

2.2 Using the net-snmp library from C/C++

The net-snmp library can be downloaded from http://www.net-snmp.org. It can be linked and used from C and C++ programs. Several tutorials and demo examples on the net-snmp website illustrate the usage of the library. Synchronous and asynchronous communication is possible.

The OID or SNMP variable name that is used by snmpset or snmpget must be passed to the library function “get_node()”. SNMP variable names can only be used if the WIENER-CRATE-MIB.txt file has been loaded previously.

To simplify the usage of net-snmp with the Wiener MPOD we provide a small open source library “WienerMpodLvHvSnmp”. The source code and the provided examples can be used as an example of how to make use of net-snmp for controlling a Wiener MPOD in a synchronous way.

2.3 Using the WienerMpodLvHvSnmp library from C/C++

The WienerMpodLvHvSnmp library can be downloaded as source code or compiled shared library from our website. The license allows all kind usage in commercial or non-commercial applications.

The library is not threadsafe. As the synchronous interface is used, function execution might take long if errors such as communication problems occur. For integrating applications it is very important to handle this appropriately. The integrating software should avoid to make costly calls to the library if the Wiener MPOD was disconnected or other errors occur.

The library consists of the C++ class “WienerMpodLvHvSnmp” and C access functions. The C++ class and it’s access methods are documented in the “WienerMpodLvHvSnmp.h” header file.

The C functions are declared in the “wiener_mpod_lv_hv_snmp.h” header file. They use a C++ object casted to a void* pointer and the C++ methods. There are several types of C functions:

- The function mpod_snmp_create() that creates an object pointer that is used by all other functions. Call mpod_snmp_delete() to delete that object.
- Functions to establish the connection to an MPOD crate and retrieve crate status and error messages. These functions are documented in the header file.
- Access functions that are named after SNMP variables as follows:
  - “mpod_snmp_CrateGet...” for reading crate variables. A single argument, the object pointer is required.
  - “mpod_snmp_CrateSet...” for writing crate variables. 2 arguments, the object pointer and the new value, are required.
  - “mpod_snmp_ModuleGet...” for reading module variables. 2 arguments, the object pointer and the slot index of the module, are required.
  - “mpod_snmp_ModuleSet...” for writing module variables. 3 arguments, the object pointer, the slot index of the module and the new value, are required.
  - “mpod_snmp_ChannelGet...” for reading channel or output variables. 3 arguments, the object pointer, the slot index of the module and the channel index, are required.
  - “mpod_snmp_ChannelSet...” for writing channel or output variables. 4 arguments, the object pointer, the slot index of the module, the channel index and the new value, are required.
  - “mpod_snmp_GroupGet...” for reading group variables. 2 arguments, the object pointer and the group index, are required.
  - “mpod_snmp_GroupSet...” for writing group variables. 3 arguments, the object pointer, the group and the new value, are required.

  The rest of the name corresponds to the name of the SNMP variable. For example the SNMP variable “outputVoltage“ is read using the function “mpod_snmp_ChannelGetVoltage” and written using “mpod_snmp_ChannelSetVoltage”.

The usage of the functions is illustrated in the provided examples.

2.4 Using PySNMP from Python

…

2.5 Using the WinSNMP library from C/C++ on Windows

.

.

2.6 Using EPICS Controls

The EPICS control system is a free and open source control system (https://epics-controls.org). It is used to operate devices such as particle accelerators, telescopes and other large experiments.

MPOD crates can be controlled by using a software support module such as the NSCL EPICS snmp device driver. This driver can be downloaded from https://groups.nscl.msu.edu/controls/. It can also be found from the EPICS website → Resources and Support → Modules → Soft Support.

This driver requires an appropriate .db file to convert EPICS device server requests to SNMP. An example db file for the device driver can be generated from Easy LV|HV for an existing MPOD connection. A more generic set of files (several db files and a sub file) can be downloaded from our website or provided on request.

The generated files might have to be adjusted in a text editor.

The names of the generated db records are similar to the names of the SNMP variables. Therefore all aspects of this documentation can easily be carried to EPICS.

2.7 Using TANGO Controls

The TANGO control system is a free and open source control system (http://www.tango-controls.org). Three device classes allow to control MPOD crates and the included modules over SNMP:

- **HVisegMPODCtrl**: this device class establishes a SNMP connection to an MPOD crate and allows to read and write SNMP variables in a format that is identical to the format used by net-snmp.
- **HVisegEHS80**: this device class uses HvisegMPODCtrl. It simplifies the basic control commands required to control an iseg EHS module.
- **WienerMpodLvHvCtrl**: this device class defines low level commands for writing and reading SNMP variables and a higher level command set that simplifies the access to net-snmp variables. An overview over this device class will be given here.

- **WienerMpodHvModule**: this device class connects to a WienerMpodLvHvCtrl device. It exposes the SNMP variables of a single module and it's channels as attributes. This device class takes into account the specifities of a HV module. It is inappropriate for a LV module.

2.7.1 The WienerMpodLvHvCtrl device class

In order to use the WienerMpodLvHvCtrl device class, the source code with examples has to be downloaded and compiled. Next a device making use of this device class has to be defined in the TANGO database. Now the device server can be started and accessed from TANGO controls for example by running the Python examples provided in the source code package.

- The source code can be downloaded from the following web site http://www.tango-controls.org/developers/dsc/families/?family=PowerSupply. The device class WienerMpodLvHvCtrl will provide a link to the source code repository where the source code and the Python-TANGO examples can be downloaded. The "trunk" directory contains the latest version.
- The README.txt file included in the device class repository provides some hints for compiling the device class. The TANGO library with headers and the net-snmp library with headers are required to compile the source code. On debian the package "libsnmp-dev" can be installed for the net-snmp dependency.
- For the Python examples a device "mpod_test/mpod/1" must have been created. The Python example .../python_usage_examples/0001-ds-configuration.py gives some hints how this device can be created from Python. The device property "IpAddr" needs to be adjusted to the IP address of the MPOD crate.
- Now start the device server by launching the newly compiled executable "WienerMpodLvHvCtrl" with the name given to the device server in the device configuration. In the provided examples the full command is:

  > WienerMpodLvHvCtrl mpod_test

- Now the device "mpod_test/mpod/1" is accessible from TANGO.
- The documentation in ".../doc_html/index.html" contains a documentation of the available device server commands.
- The example .../python_usage_examples/0003-ds-low-level-snmp.py shows how to use the low-level commands such as "SnmpGetValueDouble". These commands start with "Snmp...". They require either an OID or an equivalent SNMP variable name as argument. The SNMP variable name only works if a WIENER-CRATE-MIB.txt was installed with the net-snmp installation. To avoid this restriction and work with human readable names nevertheless, the higher level commands might be used.
- The higher level commands are named after SNMP variables as follows:
  - "CrateGet..." for reading crate variables. No argument is required.
  - "CrateSet..." for writing crate variables. 1 argument, the new value, is required.
  - "ModuleGet..." for reading module variables. 1 argument, the slot index of the module, is required.

  - “ModuleSet...” for writing module variables. 2 arguments, the slot index of the module and the new value, are required.
  - “ChannelGet...” for reading channel or output variables. 2 arguments, the slot index of the module and the channel index, are required.
  - “ChannelSet...” for writing channel or output variables. 3 arguments, the slot index of the module, the channel index and the new value, are required.
  - “GroupGet...” for reading group variables. 1 argument, the group index, is required.
  - “GroupSet...” for writing group variables. 2 arguments, the group and the new value, are required.

  The rest of the name corresponds to the name of the SNMP variable. For example the SNMP variable “outputVoltage“ is read using the device server command “ChannelGetVoltage” and written using “ChannelSetVoltage”. The examples 0005-ds-high-level-crate.py, 006-ds-high-level-module.py and 0007-ds-high-level-channel.py list all available high-level device server commands.

- The example .../python_usage_examples/0004-ds-snmpwalk.py shows how to use the low-level commands to iterate over all SNMP variables and print their values.
- The commands “CrateGetOccupiedSlots”, “ModuleGetNumberOfChannels” and “ModuleGetIsHvModule” have no equivalent SNMP variable. They have been introduced for convenience.
- The example 0010-ds-analyze-status-flags.py shows how to analyze the status flags returned for the crate, the included modules and low- and high-voltage channels.
- The example 0011-ds-low-voltage-channels.py is a more complete example. It traverses all low-voltage channels and switches the on and off.
- The example 0012-ds-high-voltage-channels.py does the same thing for high-voltage channels.

CHAPTER 3 THE CRATE STRUCTURE

An MPOD might host several modules. These modules contain one or several high- or low-voltage channels. In order to address all modules and channels efficiently the crate structure needs to be known. Some of the SNMP variables allow to retrieve information on the crate structure and the number of modules and channels. These variables can only be read. The information is only available once the crate has been switched on.

3.1 Number of modules and module numbering

If the crate is switched on the commands

> snmpget -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103 moduleNumber.0

> snmpget -v2c 192.168.0.103 -c guru iso.3.6.1.4.1.19947.1.3.5.0

return the number of modules inserted into the crate. A maximum of 10 modules are supported.

All further module variables are available as vectors. That means that variables as the module descriptions can be retrieved for each of the inserted modules.

As a consequence retrieving module information requires the slot index of the module. The occupied slots are not necessarily slot 0, 1, etc. Therefore the slot indexes must be determined by attempting to read a variable for increasing slot indices and by remembering the slot indices returning valid data:

> snmpget -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103 moduleDescription.<module name>

> snmpget -v2c 192.168.0.103 -c guru iso.3.6.1.4.1.199471.3.6.1.2.<slot index>

In these commands the placeholders <module name> must be replaced by “ma0” … “ma9” and <slot index> must be replaced by 1 … 10 to address the module in slot 0 … 9.

If only slot 2 is occupied, the above commands will return valid data only for <module name> = “ma2” and <slot index> = 3.

The valid <module names> and <slot indices> must be memorized as they are important for channel numbering.

3.2 Number of channels and channel numbering

If the crate is switched on the commands

> snmpget -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103 outputNumber.0

> snmpget -v2c 192.168.0.103 -c guru iso.3.6.1.4.1.19947.1.3.1.0

return the number of outputs or channels available for the entire crate. Crate numbering allows to address up to 2000 channels. When reading a channel variable such as the outputStatus

> snmpget -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103 outputStatus.<channel name>

> snmpget -v2c 192.168.0.103 -c guru iso.3.6.1.4.1.199471.1.3.2.1.3.<channel index>

the appropriate <channel name> and <channel index> have to be specified. For the channels of the module in slot 0 the channel names “U0” … “U99” and the channel indexes 1 … 100 are used. If the module only has 8 channels, only the channel names “U0” … “U7” and the channel indexes 1 … 8 will return valid data.

The module in slot 1 uses the channel names “U100” … “U199” and the channel indexes 101 … 200 etc.

3.3 Number of groups and group numbering

Groups can be used for 2 purposes. A group index from 0 … 63 can be assigned to LV channels. The channel’s output supervision behavior can be set to a mode, where all channels of the same group are switched off when a supervision condition is violated.

All HV channels are assigned to the group index 64. This assignment can’t be changed. The groutsSwitch variable can be used with group index 64 to enable or disable the “fine adjust” and “kill” function of the HV modules.

The number of groups depends on how the LV channels have been assigned to groups. It can be retrieved by the following commands:

```
> snmpget -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103 groupsNumber.0
```

```
> snmpget -v2c 192.168.0.103 -c guru iso.3.6.1.4.1.19947.1.3.3.0
```

3.4 Number of communities and community numbering

The number of available SNMP community variables depends on the community used to query the SNMP variables. A maximum of 4 communities with the indexes 1 … 4 is available for a user with “guru” access rights.

3.5 Number of auxiliary voltage sources and their numbering

Auxiliary voltage sources are used internally to power the HV modules with 24V. The number of auxiliary voltage sources can be retrieved using the psAuxiliaryNumber variable:

```
> snmpget -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103 psAuxiliaryNumber.0
```

```
> snmpget -v2c 192.168.0.103 -c guru iso.3.6.1.4.1.19947.1.6.4.0
```

Once the number of auxiliary voltages is known, the variables psAuxiliaryMeasurementVoltage and psAuxiliaryMeasurementCurrent can be read to verify the internal power supply of the HV modules.

If the above commands return the integer value 8, the auxiliary voltage sources are numbered 1 … 8 and named u0 … u7.

As a consequence the integers 1 … 8 can be used as <aux index> and the names u0 … u7 can be used as <aux name> when reading the SNMP variables psAuxiliaryMeasurementVoltage and psAuxiliaryMeasurementCurrent.

CHAPTER 4 STANDARD SETTINGS

The following variables allow to set the standard settings for an MPOD crate, it's modules and channels. Managing these variables is sufficient to operate a crate but no advanced features are used.

4.1 Crate Standard Settings

The most important crate variables allow to switch on a crate and to perform a hardware reset.

4.1.1 Switching a crate on or off (crate variable sysMainSwitch)

The variable "sysMainSwitch" can be read by using the net-snmp command line tool snmpget:

> snmpget -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103 sysMainSwitch.0

> snmpget -v2c 192.168.0.103 -c guru iso.3.6.1.4.1.19947.1.1.1.0

The snmpget command returns the integer value 1 if the crate is switched on and 0 if it switched off.

The MPOD crate is switched on or off by writing a 1 or a 0 to the variable "sysMainSwitch".

> snmpset -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103 sysMainSwitch.0 i 0

> snmpset -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.1.1.0 i 1

4.1.2 Performing a hardware reset on a crate (crate variable sysHardwareReset)

The variable "sysHardwareReset" can be written by using the net-snmp command line tool snmpset. The commands

> snmpset -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103 sysHardwareReset.0 i 1

> snmpset -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.1.4.0 i 1

will perform a hardware reset on the crate.

4.2 Module Standard Settings

Module variables are only relevant for HV power supply modules. For basic operation it is sufficient to clear HV module events flags that might have been set by the hardware.

When reading or writing the following variables, the placeholder <module name> has to be replaced by "ma0" … "ma9" and <slot index> has to be replaced by the integers 1 … 10.

The module name "ma3" and the slot index 4 both refer to the module in slot 3. For more details please refer to the "crate structure" chapter.

4.2.1 Clearing module events (module variable moduleDoClear)

The variable “moduleDoClear” can be written using the net-snmp command line tool snmpset. The commands

> snmpset -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103 moduleDoClear.<module name> i 1

> snmpset -v2c 192.168.0.103 -c guru iso.3.6.1.4.1.19947.1.3.6.1.11.<slot index> i 1

will clear all module events. The placeholders <module name> and <slot index> must be replaced as defined above.

Important: module events depend on channel events. Therefore channel events should have been cleared before clearing module events. Channel events are cleared by writing the integer value 10 to the variable “outputSwitch” of the corresponding channel.

4.3 Channel Standard Settings

The standard channel setting allow to switch a channel on or off and define output voltages and currents.

The placeholder <channel name> have to be replaced by “u0” … “u1999” and <channel index> has to be replaced by the integers 1 … 2000.

The channel name “u307” and the channel index 308 both refer to the channel 7 of the module in slot 3. For more details please refer to the “crate structure” chapter.

4.3.1 Switching a channel on or off (module variable outputSwitch)

The variable “outputSwitch” can be written by using the net-snmp command line tool snmpset:

> snmpset -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103 outputSwitch.<channel name> i <value>

> snmpset -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.3.2.1.9.<channel index> i <value>

The placeholders <module name> and <slot> must be replaced as defined above. Different actions are performed for different values for the placeholder <value>:

0	The channel will be switched off.
1	The channel will be switched on.
2	The “emergency off” status is cleared (HV channels only).
3	An “emergency off” signal is sent to the channel (HV channels only).
10	All channel events are cleared (HV channels only).

4.3.2 Reading and writing the output voltage Vset for a channel (module variable outputVoltage)

The variable “outputVoltage” can be read by using the net-snmp command line tool snmpget. The commands

> snmpget -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103 outputVoltage.<channel name>

> snmpget -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.3.2.1.10.<channel index>

will return the output voltage defined for the channels defined by <channel name> and <channel index>. The voltages can be set to 2V by the following commands:

> snmpset -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103 outputVoltage.<channel name> F 2

> snmpset -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.3.2.1.10.<channel index> F 2

4.3.3 Reading and writing the output current Iset for a channel (module variable outputCurrent)

The variable “outputCurrent” can be read by using the net-snmp command line tool snmpget. The commands

> snmpget -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103 outputCurrent.<channel name>

> snmpget -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.3.2.1.12.<channel index>

will return the output current defined for the channels defined by <channel name> and <channel index>. The currents can be set to 1mA by the following commands:

> snmpset -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103 outputCurrent.<channel name> F 0.001

> snmpset -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.3.2.1.12.<channel index> F 0.001

CHAPTER 5 STATUS FLAGS

The following variables allow to retrieve information about the current status of an MPOD crate, it's modules and channels.

5.1 Retrieving crate status flags (module variable sysStatus)

The variable "sysStatus" can be read by using the net-snmp command line tool snmpget:

```
> snmpget -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103 sysStatus.0
```

```
> snmpget -v2c 192.168.0.103 -c guru iso.3.6.1.4.1.19947.1.1.2.0
```

The table below illustrates the meaning of the individual bits.

Important: The bits are ordered as in the table, the first bit or most significant bit is bit 0, the last bit or least significant bit is bit 15:

Bit	0	1	2	3	4	5	6	7
Crate Flag	ON	INHIBIT	LOCAL	INPUT ERROR	CHANNEL ERROR	FAN	-	-

Bit	8	9	10	11	12	13	14	15
Crate Flag	INCOMPAT	RESET	DERATING	SUPPLY	DERATING 2	SUPPLY 2	-	-

Flag Name	**Description**
ON	Crate status flag "Main On". The crate is switched on.
INHIBIT	Crate status flag "Main Inhibit". An external (hardware-) interlock of the complete system is active.
LOCAL	Crate status flag "Local Control Only". Only local control is possible (CAN BUS write access denied).
INPUT ERROR	Crate status flag "Input Failure". An input failure such as a power fail occurred.
CHANNEL ERROR	Crate status flag "Channel Error". An channel error occurred. More details are available from the channel status flags.

FAN	Crate status flag "Fan Tray Failure". A fan tray failure occurred.
INCOMPAT	Crate status flag "Plug And Play Incompatible". A wrong power supply and rack have been connected.
RESET	Crate status flag "Bus Reset". The system bus (e.g. VME or CPCI) reset signal is active.
DERATING	Crate status flag "Supply Derating". The first system power supply has the DEG signal active.
SUPPLY	Crate status flag "Supply Failure". The first system power supply has the FAL signal active.
DERATING 2	Crate status flag "Supply Derating 2". The second system power supply has the DEG signal active.
SUPPLY 2	Crate status flag "Supply Failure 2". The second system power supply has the FAL signal active.

Example: the above snmpget command

```
> snmpget -v 2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103 sysStatus.0
```

returns

```
WIENER-CRATE-MIB::sysStatus.0 = BITS: 80 mainOn(0)
```

The hexadecimal result 80 can be written in binary as follows:

80_{hex} = 1000 0000 0000 0000_{bin}

this means that bit 0 is set. The status flag “ON” is active. If the command returns the hexadecimal string 40 10, the binary representation

40 10_{hex} = 0100 0000 0001 0000_{bin}

shows that bit 1 “INHIBIT” and bit 11 “SUPPLY” are set.

5.2 Retrieving module status flags (module variable moduleStatus)

As most module variables, “moduleStatus” only applies to **high voltage modules**. This variable can be read for all modules at once by using the net-snmp command line tool snmpwalk:

```
> snmpwalk -v2c -m +WIENER-CRATE-MIB 192.168.0.103 -c guru moduleStatus
```

```
> snmpwalk -v2c 192.168.0.103 -c guru iso.3.6.1.4.1.19947.1.3.6.1.8
```

The net-snmp command line tool snmpget can can be used to read the module status for a single module. The module name “ma0” and the module index 1 denote the module in slot 0. This module must be present for the following commands:

```
> snmpget -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103 moduleStatus.ma0
```



```
> snmpget -v2c 192.168.0.103 -c guru iso.3.6.1.4.1.19947.1.3.6.1.8.1
```

Module status flags are transient. They disappear as soon as the condition triggering the status flag is no longer true.

Important: The bits are ordered as in the table, the first bit or most significant bit is bit 0, the last bit or least significant bit is bit 15.

Warning: The bar over flag names indicates that the flag is inverted. That means a logical 0 indicates that the flag is active.

Important: Please note that if “kill” is enabled, the voltage ramp for the module is limited.

Bit	0	1	2	3	4	5	6	7
HV Flag	ADJ	-	LIVE INS	HV	SERVICE	V LIMIT	IN ERROR	-

Bit	8	9	10	11	12	13	14	15
HV Flag	$\overline{\text{SUM ERROR}}$	$\overline{\text{RAMP}}$	$\overline{\text{SAFETY}}$	EVENT	$\overline{\text{MOD ERROR}}$	$\overline{\text{SUPPLY}}$	$\overline{\text{MAX TEMP}}$	KILL

Flag Name	**Description**
ADJ	Module status flag "Fine Adjustment Active". This flag is set if the module is in fine adjustment mode. An additional compensation loop in the firmware adjusts Vmeas to the user set value for Vset. Fine adjustment takes some time to be effective after having enabled the option.
LIVE INS	Module status flag "Live Insertion". This flag is set if a hot plug is prepared for the module. Live insertion is not available for all power supplies.
HV	Module status flag "High Voltage On". This flag is set if at least one channel delivers a high voltage.
SERVICE	Module status flag "Maintenance Required". This flag is set if the module has to be returned for maintenance.
V LIMIT	Module status flag "Hardware Voltage Limit Exceeded". This flag is set if the hardware voltage limit isn't in the correct range. This flag is not available for all hardware. It is only informed by HV distributor modules with current mirror.
IN ERROR	Module status flag "Input Error". This flag will appear for example if a set value is out of range, has a bad polarity sign or for another communication problem. Channel input errors are also reported here.

SUM ERROR	Module status flag "Sum Error". This flag is set if a critical event occurred in at least one channel. The concerned channel status flags are the following: I LIMIT, V LIMIT, TRIP, INHIBIT, V BOUND, I BOUND. This flag is inverted. A logical 0 indicates that this flag is active.
RAMP	Module status flag "Ramping". This flag is set if at least one channel is ramping up or down. This flag is inverted. A logical 0 indicates that this flag is active.
SAFETY	Module status flag "Safety Loop Open". This flag is set if the safety loop is open. The SL connector on the front panel is potential free and requires an external 5-20 mA current to be closed. The internal optocoupler has a voltage drop of approx. 3 V. The safety loop needs to be activated by removing the jumper on the module's bottom side. This flag is inverted. A logical 0 indicates that this flag is active.
EVENT	Module status flag "Event Active". This flag is set if a module event is active and the masks have been set accordingly. The module events that will trigger this flag are: SAFETY, ...
MOD ERROR	Module status flag "Module Error". This flag is set if one of the following module flags is active : SUM ERROR, MAX TEMP, SUPPLY, SAFETY. This flag is inverted. A logical 0 indicates that this flag is active.
SUPPLY	Module status flag "Power Supply Failure". This flag is set if the module's power supply fails. This flag is inverted. A logical 0 indicates that this flag is active.
MAX TEMP	Module status flag "Temperature High". This flag is set if the temperature of the module is too high. This flag is inverted. A logical 0 indicates that this flag is active.
KILL	Module status flag "Kill Enabled". This flag is set if the "Kill" option has been enabled for the module. If "Kill" is enabled, several channel status flags will lead to a shutdown of the channel. These status flags are TRIP, I LIMIT, V LIMIT, I BOUND, V BOUND, ARC.

Example: the above snmpget command

```
> snmpget -v 2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103 moduleStatus.ma0
```

returns

WIENER-CRATE-MIB::moduleStatus.ma3 = BITS: 00 EE moduleIsNoSumError(8) moduleIsNoRamp(9) moduleSafetyLoopIsGood(10) moduleIsGood(12) moduleSupplyIsGood(13) moduleTemperatureIsGood(14)

The hexadecimal result 00 EE can be written in binary as follows:

$00\ EE_{hex} = 0000\ 0000\ 1110\ 1110_{bin}$

this means that the bits 8, 9, 10, 12, 13 and 14 are set. This corresponds to the flags $\overline{\text{SUM ERROR}}$, $\overline{\text{RAMP}}$, $\overline{\text{SAFETY}}$, $\overline{\text{MOD ERROR}}$, $\overline{\text{SUPPLY}}$ and $\overline{\text{MAX TEMP}}$. As all these flags are inverted. No status event flag is active.

5.3 Retrieving module event flags (module variable moduleEventStatus)

As most module variables, “moduleEventStatus” only applies to **high voltage modules**. This variable can be read for all modules at once by using the net-snmp command line tool snmpwalk:

> snmpwalk -v2c -m +WIENER-CRATE-MIB 192.168.0.103 -c guru moduleEventStatus

> snmpwalk -v2c 192.168.0.103 -c guru iso.3.6.1.4.1.19947.1.3.6.1.9

The net-snmp command line tool snmpget can can be used to read the module event status for a single module. The module name “ma0” and the module index 1 denote the module in slot 0. This module must be present for the following commands:

> snmpget -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103 moduleEventStatus.ma0

> snmpget -v2c 192.168.0.103 -c guru iso.3.6.1.4.1.19947.1.3.6.1.9.1

Module event status flags are permanent. They are triggered as the module status flag but they don’t disappear automatically. Instead they must be cleared by the user by writing the SNMP variable moduleDoClear.

Important: The bits are ordered as in the table, the first bit or most significant bit is bit 0, the last bit or least significant bit is bit 15.

Warning: The flags inverted in the module status aren’t inverted here. That means for all bits a logical 1 indicates that the flag is active.

Bit	0	1	2	3	4	5	6	7
HV Flag	ADJ	-	LIVE INS	HV	SERVICE	V LIMIT	IN ERROR	-

Bit	8	9	10	11	12	13	14	15
HV Flag	SUM ERROR	RAMP	SAFETY	EVENT	MOD ERROR	SUPPLY	MAX TEMP	KILL

The meaning of the individual bits is identical to the module's status flags.

Example: the above snmpget command

```
> snmpget -v 2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103 moduleEventStatus.ma0
```

returns

```
WIENER-CRATE-MIB::moduleEventStatus.ma0 = BITS: 20 moduleEventLiveInsertion(2)
```

The hexadecimal result 20 can be written in binary as follows:

20_{hex} = 0010 0000 0000 0000_{bin}

this means that the bit 2 is set. This corresponds to the flag LIVE INS.

5.4 Retrieving channel status flags (channel variable outputStatus)

The read-only variable "outputStatus" returns a value of type bit string. The individual bits of the returned value inform on the current state of the channel. This variable can be read for all channels at once by using the net-snmp command line tool snmpwalk:

```
> snmpwalk -v2c -m +WIENER-CRATE-MIB 192.168.0.103 -c guru outputStatus
```

```
> snmpwalk -v2c 192.168.0.103 -c guru iso.3.6.1.4.1.19947.1.3.2.1.4
```

The net-snmp command line tool snmpget can can be used to read the channel status for a single channel. The channel name u100 or channel index 101 must exist:

```
> snmpget -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103 outputStatus.u100
```

```
> snmpget -v2c 192.168.0.103 -c guru iso.3.6.1.4.1.19947.1.3.2.1.4.101
```

As illustrated in the examples below the bit ordering needs to be respected when analyzing the status flags. Only few bits are common to low- and high-voltage channels. Therefore the returned values must be interpreted differently for low- and high-voltage channels.

5.4.1 Interpreting channel status flags for low-voltage channels

The below table illustrates the meaning of each bit for low-voltage channels.

Important: The bits are ordered as in the table, the first bit or most significant bit is bit 0, the last bit or least significant bit is bit 15.

Bit	0	1	2	3	4	5	6	7
LV Flag	ON	INHIBIT	MIN SENSE V	MAX SENSE V	MAX TERM V	MAX I	MAX TEMP	MAX PLOAD

Bit	8	9	10	11	12	13	14	15
LV Flag	-	TIME	CC	RAMP	RAMP	-	-	-

		OUT		UP	DOWN			

Flag Name	**Description**
ON	The channel has been switched on.
INHIBIT	External Inhibit Detected. An external inhibit signal is detected either on the interlock pin of the controller or on the inhibit pins of the module.
MIN SENSE V	Sense Voltage Below Minimum. Supervision limit hurt: The measured sense voltage Vmeas is too low.
MAX SENSE V	Max Sense Voltage Exceeded. Supervision limit hurt: The measured sense voltage Vmeas is too high.
MAX TERM V	Max Terminal Voltage Exceeded. Supervision limit hurt: The measured terminal voltage Vterm is too high.
MAX I	Max Current Exceeded. Supervision limit hurt: The measured current is too high.
MAX TEMP	Maximum Temperature Exceeded. Supervision limit hurt: The heat sink temperature is too high.
MAX PLOAD	Maximum Load Power Exceeded. Supervision limit hurt: The output power on the load is too high.
TIME OUT	Timeout Occurred. A timeout occurred in the communication between the output channel and the crate controller.
CC	Constant Current Mode. This flag is set if the current defined for Iset is reached. The channel operates in current control mode.
RAMP UP	Ramping Up. The channel is ramped up.
RAMP DOWN	Ramping Down. The channel is ramped down.

Example: the above snmpget command

```
> snmpget -v 2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103 outputStatus.u100
```

returns

```
WIENER-CRATE-MIB::outputStatus.u100 = BITS: 80 10 outputOn(0) outputRampUp(11)
```

The hexadecimal result 80 10 can be written in binary as follows:

$80\ 10_{hex} = 1000\ 0000\ 0001\ 0000_{bin}$

this means that the bits 1 and 11 are set. The flags “ON” and “RAMP UP” are active.

5.4.2 Interpreting channel status flags for high-voltage channels

The below table illustrates the meaning of each bit for high-voltage channels.

Important: The bits are ordered as in the table, the first bit or most significant bit is bit 0, the last bit or least significant bit is bit 15.

Important: Please note that if “kill” is enabled, the voltage ramp for the module is limited.

Bit	0	1	2	3	4	5	6	7
HV Flag	ON	INHIBIT	-	-	V LIMIT	TRIP	-	-

Bit	8	9	10	11	12	13	14	15
HV Flag	-	-	CC	RAMP UP	RAMP DOWN	KILL	EMCY	ADJ

Bit	16	17	18	19	20	21	22	23
HV Flag	CV	LCR	V BOUND	I LIMIT	-	-	-	-

Flag Name	**Description**
ON	The channel has been switched on.
INHIBIT	External Inhibit Detected. An external inhibit signal is detected on the inhibit pin of the module. The channel will be shut down according to the "External Inhibit Action" in the channel properties. This flag is also set if the crate controller’s interlock is active.
V LIMIT	Voltage Limit Exceeded. This flag is set if the voltage exceeds the value defined for the hardware voltage limit (Vmax potentiometer). If the "Kill" option has been enabled for the module, the channel will be shut down.
TRIP	"Current Trip Occurred". This flag is set if Imeas exceeds Iset and the "Kill" option is enabled for the module or a delayed trip action has been defined. If a delayed trip is used the flag will only be set after Imeas exceeded Iset for a user-defined delay time. If the "Kill" option isn't enabled for the module and no delayed trip action has been defined,

	the module will operate in current control mode when the output current reaches Iset.
CC	Constant Current Mode. The module option "kill" has been disabled and no delayed trip action has been defined: this flag is set if the current defined for Iset is reached. The channel operates in current control mode. The module option "kill" has been enabled or a delayed trip action has been defined: this flag is never set. A current trip will occur instead.
RAMP UP	Ramping Up. The channel is ramped up.
RAMP DOWN	Ramping Down. The channel is ramped down.
KILL	Kill Enabled. This flag is set if the "Kill" option has been enabled for the module. If "Kill" is enabled, several channel status flags will lead to a shutdown of the channel. These status flags are TRIP, I LIMIT, V LIMIT, I BOUND, V BOUND.
EMCY	Emergency Off. The emergency off is triggered by the user. The channel is shut down without ramp. The emergency has to be cleared before the channel can be switched on again.
ADJ	Fine Adjustment Active. This flag is set if the module is in fine adjustment mode. An additional compensation loop in the firmware adjusts Vmeas to the user set value for Vset. Fine adjustment takes some time to be effective after having enabled the option.
CV	Constant Voltage Mode. This flag is set if the voltage defined for Vset is reached. The channel operates in voltage control mode. This flag is also active when the channel is ramped up or down.
LCR	Low Current Measurement Range. This flag is set if the low current range is used for current measurements. This increases the precision of Imeas. The low current measurement range is only available for high precision power supplies.
V BOUND	Vbound Exceeded. This flag is set if | Vmeas - Vset | > Vbound. If the "Kill" option has been enabled for the module, the channel will be shut down.
I LIMIT	Current Limit Exceeded. This flag is set if the current exceeds the value defined for the hardware current limit (Imax potentiometer). If the "Kill" option has been enabled for the module, the channel will be shut down.

Example: the above snmpget command

```
> snmpget -v 2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103 outputStatus.u100
```

returns

WIENER-CRATE-MIB::outputStatus.u100 = BITS: 80 01 40 outputOn(0) outputAdjusting(15) outputLowCurrentRange(17)

The hexadecimal result 80 01 40 can be written in binary as follows:

$80\ 01\ 40_{hex}$ = $1000\ 0000\ 0000\ 0001\ 0100\ 0000_{bin}$

this means that the bits 1, 15 and 17 are set. The flags “ON” and “ADJ” and “LCR” are active.

CHAPTER 6 CONFIGURATION

The following variable allow to configure an MPOD crate, it's modules and channels.

6.1 Crate Configuration

6.1.1 Changing the snmp community name (crate variable snmpCommunityName)

The variable "snmpCommunityName" can be read and written by using the net-snmp command line tools snmpget and snmpset. The commands

> snmpget -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103 snmpCommunityName.<community>

> snmpget -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.5.1.1.1.2.<community>

will return the community name for different access right levels. The commands

> snmpset -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103 snmpCommunityName.<community> s <new community name>

> snmpset -v2c -c guru 192.168.0.103 so.3.6.1.4.1.19947.1.5.1.1.1.2.<community> s <new community name>

Will change the community names to <new community name>. The placeholder <community> takes the values 1 .. 4.

Example: change the guru community name to "superuser":

> snmpset -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103 snmpCommunityName.4 s superuser

To read the new value "superuser" has to be specified in the snmpget command:

> snmpget -v2c -m +WIENER-CRATE-MIB -c superuser 192.168.0.103 snmpCommunityName.4

6.1.2 Reading the power supply's serial number (crate variable psSerialNumber)

Each MPOD crate can be identified by reading it's power supply's serial number. The variable "psSerialNumber" can be read by using the net-snmp command line tool snmpget. The commands

> snmpget -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103 psSerialNumber.0

> snmpget -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.6.2.0

will return the serial number as string.

6.1.3 Reading the power supply's operating time (crate variable psOperatingTime)

The variable "psSerialNumber" can be read by using the net-snmp command line tool snmpget. It returns the number of seconds the power supply has been switched on. The commands

> snmpget -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103 psOperatingTime.0

> snmpget -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.6.3.0

will return the operating time as integer value.

READ WRITE ? IS IT RESET EACH TIM WE SWITCH THE CRATE OFF ?

6.1.4 Reading and writing the fan speed (crate variable fanNominalSpeed)

The variable “fanNominalSpeed” can be read and written by using the net-snmp command line tools snmpget and snmpset. The commands

> snmpget -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103 fanNominalSpeed.0

> snmpget -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.7.6.0

will return the nominal fan speed as integer. The fan speed can be set by the commands

> snmpset -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103 fanNominalSpeed.0 i <fan speed>

> snmpset -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.7.6.0 i <fan speed>

The value for <fan speed> can be 0 or 1200 .. 3600.

6.2 Module Configuration

Only HV modules can be configured. The configurable items are the voltage and current ramp. The voltage and current limits of the module can be adjusted using the Vmax and Imax potentiometers. The corresponding SNMP variables are read-only.

6.2.1 Reading and writing voltage ramp speed (module variable moduleRampSpeedVoltage)

The variable “moduleRampSpeedVoltage” can be read and written by using the net-snmp command line tools snmpget and snmpset. The commands

> snmpget -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103
moduleRampSpeedVoltage.<module name>

> snmpget -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.3.6.1.6.<slot index>

will return the voltage ramp speed as floating point number. The voltage ramp speed is given as percentage of the nominal voltage of the HV channels. It can be set by the commands

> snmpset -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103
moduleRampSpeedVoltage.<module name> F <ramp speed>

> snmpset -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.3.6.1.6.<slot index> F <ramp speed>

The maximum value for <ramp speed> depends on the concerned HV module. Most modules accept ramp speed values up to 10% but for some modules this value can’t exceed 1%.

Important: Please note that the voltage ramp for HV module is always limited to 1% if “kill” has been enabled or if a delayed trip has been set up.

The placeholder <module name> has to be replaced by “ma0” … “ma9”. <slot index> has to be replaced by the integer values 1 … 10. See the chapter on crate structure for more details.

6.2.2 Reading and writing current ramp speed (module variable moduleRampSpeedCurrent)

The variable “moduleRampSpeedCurrent” can be read and written by using the net-snmp command line tools snmpget and snmpset. The commands

> snmpget -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103
moduleRampSpeedCurrent.<module name>

> snmpget -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.3.6.1.7.<slot index>

will return the current ramp speed as floating point number. The current ramp speed is given as percentage of the nominal current of the HV channels. It can be set by the commands

> snmpset -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103
moduleRampSpeedCurrent.<module name> F <ramp speed>

> snmpset -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.3.6.1.7.<slot index> F <ramp speed>

The value for <ramp speed> depends on the concerned HV module. The placeholder <module name> has to be replaced by “ma0” … “ma9”. <slot index> has to be replaced by the integer values 1 … 10. See the chapter on crate structure for more details.

6.2.3 Reading the hardware voltage limit (module variable moduleHardwareLimitVoltage)

The read-only variable “moduleHardwareLimitVoltage” can be read by using the net-snmp command line tool snmpget . The commands

> snmpget -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103
moduleHardwareLimitVoltage.<module name>

> snmpget -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.3.6.1.4.<slot index>

will return the hardware voltage limit as floating point number. It is given as percentage of the nominal voltage of the HV channels.

6.2.4 Reading the hardware current limit (module variable moduleHardwareLimitCurrent)

The read-only variable “moduleHardwareLimitCurrent” can be read by using the net-snmp command line tool snmpget . The commands

> snmpget -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103
moduleHardwareLimitCurrent.<module name>

> snmpget -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.3.6.1.5.<slot index>

will return the hardware current limit as floating point number. It is given as percentage of the nominal current of the HV channels.

6.3 LV Channel Configuration, supervision behavior

The channel configuration allows to define various aspects of the channel's behavior. LV and HV channels are configured and react differently. This section handles LV channels. They are configured by defining among others the outputSupervisionBehavior variable. Read-only variables can be set by using MUSEcontrol.

6.3.1 Reading and writing the supervision behavior (channel variable outputSupervisionBehavior)

The variable "outputSupervisionBehavior " can be read and written by using the net-snmp command line tools snmpget and snmpset. The commands

```
> snmpget -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103
        outputSupervisionBehavior.<channel name>

> snmpget -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.3.2.1.15.<channel index>
```

will return the supervision behavior as integer number. It can be set by the commands

```
> snmpset -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103
        outputSupervisionBehavior.<channel name> i <value>

> snmpset -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.3.2.1.15.<channel index> i <value>
```

The allowed values for <value> are given in the hardware user manual. The placeholder <channel name> has to be replaced by "u0" … "u1999". <channel index> has to be replaced by the integer values 1 … 2000. See the chapter on crate structure for more details.

The supervision behavior defines how the channel, the module, the group or the entire crate should react if a supervision condition is violated. LV and HV channels have different possibilities as shown in the tables below.

The behavior of LV channels can be defined by combining the bits 0 .. 15 of the integer value as follows:

Bits	Value	Action
0 and 1	00_{bin}	Ignore if the measured sense voltage is below the minimum sense voltage.
	01_{bin}	Switch off channel if the measured sense voltage is below the minimum sense voltage.
	10_{bin}	Switch off all channels of the same group if the measured sense voltage is below the minimum sense voltage.
	11_{bin}	Switch off all LV channels if the measured sense voltage is below the minimum sense voltage.

2 and 3	00_{bin}	Ignore if the measured sense voltage is above the maximum sense voltage.
	01_{bin}	Switch off channel if the measured sense voltage is above the maximum sense voltage.
	10_{bin}	Switch off all channels of the same group if the measured sense voltage is above the maximum sense voltage.
	11_{bin}	Switch off all LV channels if the measured sense voltage is above the maximum sense voltage.
4 and 5	00_{bin}	unused
	01_{bin}	Switch off channel if the measured terminal voltage is above the maximum terminal voltage.
	10_{bin}	Switch off all channels of the same group if the measured terminal voltage is above the maximum terminal voltage.
	11_{bin}	Switch off all LV channels if the measured terminal voltage is above the maximum terminal voltage.
6 and 7	00_{bin}	Ignore if the measured current is above the maximum current.
	01_{bin}	Switch off channel if the measured current is above the maximum current.
	10_{bin}	Switch off all channels of the same group if the measured current is above the maximum current.
	11_{bin}	Switch off all LV channels if the measured current is above the maximum current.
8 and 9	00_{bin}	unused
	01_{bin}	Switch off channel if the measured temperature is above the maximum temperature.
	10_{bin}	Switch off all channels of the same group if the measured temperature is above the maximum temperature.
	11_{bin}	Switch off all LV channels if the measured temperature is above the maximum temperature.
10 and 11	00_{bin}	unused
	01_{bin}	Switch off channel if the measured power is above the maximum power.
	10_{bin}	Switch off all channels of the same group if the measured power is above the maximum power.
	11_{bin}	Switch off all LV channels if the measured power is above the maximum power.
12 and 13		unused

14 and 15	00_{bin}	Ignore if an internal communication timeout occurred.
	01_{bin}	Switch off channel if an internal communication timeout occurred.
	10_{bin}	Switch off all channels of the same group if an internal communication timeout occurred.
	11_{bin}	Switch off all LV channels if an internal communication timeout occurred.

The values serving as reference for the comparison can be set by defining other SNMP variables. Some reference values are read-only. They can only be changed by using MUSEcontrol.

6.3.2 Reading and writing the supervision minimum sense voltage (channel variable outputSupervisionMinSenseVoltage)

The variable “outputSupervisionMinSenseVoltage” can be read and written by using the net-snmp command line tools snmpget and snmpset.

The commands

> snmpget -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103
outputSupervisionMinSenseVoltage.<channel name>

> snmpget -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.3.2.1.16.<channel index>

will return the supervision minimum sense voltage as floating point number. It can be set by the commands

> snmpset -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103
outputSupervisionMinSenseVoltage.<channel name> F <value>

> snmpset -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.3.2.1.16.<channel index> F <value>

The allowed values for <value> are given in the hardware user manual. The placeholder <channel name> has to be replaced by “u0” … “u1999”. <channel index> has to be replaced by the integer values 1 … 2000. See the chapter on crate structure for more details.

If the measured voltage is below the minimum sense voltage, the action defined by the bits 0 and 1 of the outputSupervisionBehavior variable will be performed.

6.3.3 Reading and writing the supervision maximum sense voltage (channel variable outputSupervisionMaxSenseVoltage)

The variable “outputSupervisionMaxSenseVoltage” can be read and written by using the net-snmp command line tools snmpget and snmpset.

The commands

> snmpget -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103
outputSupervisionMaxSenseVoltage.<channel name>

> snmpget -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.3.2.1.17.<channel index>

will return the supervision maximum sense voltage as floating point number. It can be set by the commands

> snmpset -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103
outputSupervisionMaxSenseVoltage.<channel name> F <value>

> snmpset -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.3.2.1.17.<channel index> F <value>

The allowed values for <value> are limited by the value for the SNMP variable "outputConfigMaxSenseVoltage". The placeholder <channel name> has to be replaced by "u0" … "u1999". <channel index> has to be replaced by the integer values 1 … 2000. See the chapter on crate structure for more details.

If the measured voltage is above the maximum sense voltage, the action defined by the bits 2 and 3 of the outputSupervisionBehavior variable will be performed.

6.3.4 Reading and writing the supervision maximum terminal voltage (channel variable outputSupervisionMaxTerminalVoltage)

The variable "outputSupervisionMaxTerminalVoltage" can be read and written by using the net-snmp command line tools snmpget and snmpset. The commands

> snmpget -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103
outputSupervisionMaxTerminalVoltage.<channel name>

> snmpget -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.3.2.1.18.<channel index>

will return the supervision maximum terminal voltage as floating point number. It can be set by the commands

> snmpset -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103
outputSupervisionMaxTerminalVoltage.<channel name> F <value>

> snmpset -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.3.2.1.18.<channel index> F <value>

The allowed values for <value> are limited by the value returned for the SNMP variable "outputConfigMaxTerminalVoltage". The placeholder <channel name> has to be replaced by "u0" … "u1999". <channel index> has to be replaced by the integer values 1 … 2000. See the chapter on crate structure for more details.

If the measured terminal voltage is above the maximum terminal voltage, the action defined by the bits 4 and 5 of the outputSupervisionBehavior variable will be performed.

6.3.5 Reading and writing the supervision maximum current (channel variable outputSupervisionMaxCurrent)

The variable "outputSupervisionMaxCurrent" can be read and written by using the net-snmp command line tools snmpget and snmpset. The commands

> snmpget -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103
outputSupervisionMaxCurrent.<channel name>

> snmpget -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.3.2.1.19.<channel index>

will return the supervision maximum current as floating point number. It can be set by the commands

> snmpset -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103
outputSupervisionMaxCurrent.<channel name> F <value>

> snmpset -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.3.2.1.19.<channel index> F <value>

The allowed values for <value> are limited by the value returned for the SNMP variable "outputConfigMaxCurrent". The placeholder <channel name> has to be replaced by "u0" … "u1999". <channel index> has to be replaced by the integer values 1 … 2000. See the chapter on crate structure for more details.

If the measured current is above the maximum current, the action defined by the bits 6 and 7 of the outputSupervisionBehavior variable will be performed.

6.3.6 Reading the configuration maximum sense voltage (channel variable outputConfigMaxSenseVoltage)

The variable "outputConfigMaxSenseVoltage" can be read and written by using the net-snmp command line tools snmpget and snmpset. The commands

> snmpget -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103
outputConfigMaxSenseVoltage.<channel name>

> snmpget -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.3.2.1.21.<channel index>

will return the configuration maximum sense voltage as floating point number. The placeholder <channel name> has to be replaced by "u0" … "u1999". <channel index> has to be replaced by the integer values 1 … 2000. See the chapter on crate structure for more details.

Important: This value can be written using MUSEcontrol.

This variable limits the value that can be defined for "outputSupervisionMaxSenseVoltage".

6.3.7 Reading the configuration maximum terminal voltage (channel variable outputConfigMaxTerminalVoltage)

The variable "outputConfigMaxTerminalVoltage" can be read by using the net-snmp command line tool snmpget. The commands

> snmpget -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103
outputConfigMaxTerminalVoltage.<channel name>

> snmpget -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.3.2.1.22.<channel index>

will return the configuration maximum terminal voltage as floating point number. The placeholder <channel name> has to be replaced by "u0" … "u1999". <channel index> has to be replaced by the integer values 1 … 2000. See the chapter on crate structure for more details.

Important: This value can be written using MUSEcontrol.

This variable limits the value that can be defined for "outputSupervisionMaxTerminalVoltage".

6.3.8 Reading the configuration maximum current (channel variable outputConfigMaxCurrent)

The variable "outputConfigMaxCurrent" can be read by using the net-snmp command line tool snmpget. The commands

> snmpget -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103
outputConfigMaxCurrent.<channel name>

> snmpget -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.3.2.1.23.<channel index>

will return the configuration maximum current as floating point number. The placeholder <channel name> has to be replaced by "u0" … "u1999". <channel index> has to be replaced by the integer values 1 … 2000. See the chapter on crate structure for more details.

Important: This value can be written using MUSEcontrol.

This variable limits the value that can be defined for "outputSupervisionMaxCurrent".

6.3.9 Reading the maximum power (channel variable outputSupervisionMaxPower)

The variable "outputSupervisionMaxPower" can be read by using the net-snmp command line tool snmpget. The commands

> snmpget -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103
outputSupervisionMaxPower.<channel name>

> snmpget -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.3.2.1.24.<channel index>

will return the maximum power as floating point number. The placeholder <channel name> has to be replaced by "u0" … "u1999". <channel index> has to be replaced by the integer values 1 … 2000. See the chapter on crate structure for more details.

Important: This value can be written using MUSEcontrol.

If the measured power is above the maximum power, the action defined by the bits 10 and 11 of the outputSupervisionBehavior variable will be performed.

6.4 LV Channel Configuration, rise/fall rates

The channel configuration allows to define various aspects of the channel's behavior. LV and HV channels are configured and react differently. This section handles LV channels. The voltage and current rise and fall rates can be configured individually for each channel.

6.4.1 Reading and writing the voltage rise rate (channel variable outputVoltageRiseRate)

The variable "outputVoltageRiseRate" can be read and written by using the net-snmp command line tools snmpget and snmpset. The commands

> snmpget -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103
outputVoltageRiseRate.<channel name>

> snmpget -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.3.2.1.13.<channel index>

will return the voltage rise rate as floating point number. It can be set by the commands

> snmpset -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103
outputVoltageRiseRate.<channel name> F <value>

> snmpset -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.3.2.1.13.<channel index> F <value>

The allowed values for <value> are given in the hardware user manual. The placeholder <channel name> has to be replaced by “u0” … “u1999”. <channel index> has to be replaced by the integer values 1 … 2000. See the chapter on crate structure for more details.

6.4.2 Reading and writing the voltage fall rate (channel variable outputVoltageFallRate)

The variable “outputVoltageFallRate” can be read and written by using the net-snmp command line tools snmpget and snmpset. The commands

> snmpget -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103
outputVoltageFallRate.<channel name>

> snmpget -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.3.2.1.14.<channel index>

will return the voltage fall rate as floating point number. It can be set by the commands

> snmpset -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103
outputVoltageFallRate.<channel name> F <value>

> snmpset -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.3.2.1.14.<channel index> F <value>

The allowed values for <value> are given in the hardware user manual. The placeholder <channel name> has to be replaced by “u0” … “u1999”. <channel index> has to be replaced by the integer values 1 … 2000. See the chapter on crate structure for more details.

6.4.3 Reading and writing the current rise rate (channel variable outputCurrentRiseRate)

The variable “outputCurrentRiseRate” can be read and written by using the net-snmp command line tools snmpget and snmpset. The commands

> snmpget -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103
outputCurrentRiseRate.<channel name>

> snmpget -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.3.2.1.25.<channel index>

will return the current rise rate as floating point number. It can be set by the commands

> snmpset -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103
outputCurrentRiseRate.<channel name> F <value>

> snmpset -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.3.2.1.25.<channel index> F <value>

The allowed values for <value> are given in the hardware user manual. The placeholder <channel name> has to be replaced by “u0” … “u1999”. <channel index> has to be replaced by the integer values 1 … 2000. See the chapter on crate structure for more details.

6.4.4 Reading and writing the current fall rate (channel variable outputCurrentFallRate)

The variable "outputCurrentFallRate" can be read and written by using the net-snmp command line tools snmpget and snmpset. The commands

> snmpget -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103
outputCurrentFallRate.<channel name>

> snmpget -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.3.2.1.25.<channel index>

will return the current fall rate as floating point number. It can be set by the commands

> snmpset -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103
outputCurrentFallRate.<channel name> F <value>

> snmpset -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.3.2.1.25.<channel index> F <value>

The allowed values for <value> are given in the hardware user manual. The placeholder <channel name> has to be replaced by "u0" … "u1999". <channel index> has to be replaced by the integer values 1 … 2000. See the chapter on crate structure for more details.

6.5 LV Channel Configuration, user config

The channel configuration allows to define various aspects of the channel's behavior. LV and HV channels are configured and react differently. This section handles LV channels. It shows how to configure low voltage channels with sense lines. It also explains how to define the switch-off and the inhibit behavior.

6.5.1 Reading and writing the user config (channel variable outputUserConfig)

The variable "outputUserConfig " can be read and written by using the net-snmp command line tools snmpget and snmpset. The commands

> snmpget -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103 outputUserConfig .<channel name>

> snmpget -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.3.2.1.37.<channel index>

will return the user config as integer number. It can be set by the commands

> snmpset -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103 outputUserConfig .<channel name> i <value>

> snmpset -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.3.2.1.37.<channel index> i <value>

The allowed values for <value> are given in the hardware user manual. The placeholder <channel name> has to be replaced by "u0" … "u1999". <channel index> has to be replaced by the integer values 1 … 2000. See the chapter on crate structure for more details.

The individual bits in the user configuration defines how the LV channel reacts to inhibits and configures the sense lines.

Bits	Value	Action
0	0_{bin}	Ramp down at switch off.
	1_{bin}	Switch off immediately without ramp.
1 and 2	00_{bin}	Fast sense regulation mode for short cables, up to 1 meter.
	01_{bin}	Moderate sense regulation mode for cables from 1 to 30 meter.
	10_{bin}	Slow sense regulation mode for cables longer than 30 meter.
	11_{bin}	Slow sense regulation mode (should not be used).
3	0_{bin}	Use an external sense line.
	1_{bin}	Use internal sense line.
4	0_{bin}	Ignore external inhibit.
	1_{bin}	Enable external inhibit.

6.6 HV Channel Configuration

The channel configuration allows to define various aspects of the channel's behavior. LV and HV channels are configured and react differently.

6.6.1 Reading and writing the supervision behavior (channel variable outputSupervisionBehavior)

The variable "outputSupervisionBehavior " can be read and written by using the net-snmp command line tools snmpget and snmpset. The commands

> snmpget -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103 outputSupervisionBehavior .<channel name>

> snmpget -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.3.2.1.15.<channel index>

will return the supervision behavior as integer number. It can be set by the commands

> snmpset -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103 outputSupervisionBehavior .<channel name> i <value>

> snmpset -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.3.2.1.15.<channel index> i <value>

The allowed values for <value> are given in the hardware user manual. The placeholder <channel name> has to be replaced by "u0" … "u1999". <channel index> has to be replaced by the integer values 1 … 2000. See the chapter on crate structure for more details.

The behavior of HV channels can be defined by combining the bits 0 .. 15 of the integer value as follows:

Bits	Value	Action
0 and 1		unused

2 and 3		unused
4 and 5		unused
6 and 7	00_{bin}	Ignore if a current trip occurs.
	01_{bin}	Ramp down channel if a current trip occurs.
	10_{bin}	Switch off channel without a ramp if a current trip occurs.
	11_{bin}	Switch off the entire module without a ramp if a current trip occurs.
8 and 9		unused
10 and 11		unused
12 and 13	00_{bin}	Ignore if an external inhibit occurs.
	01_{bin}	Ramp down channel if an external inhibit occurs.
	10_{bin}	Switch off channel without a ramp if an external inhibit occurs.
	11_{bin}	Switch off the entire module without a ramp if an external inhibit occurs.
14 and 15		Unused

Important: please note that configuring current trips limits the voltage ramp for the concerned module.

Important: please note that the crate controller's interlock can't be configured here. If an interlock occurs, it will always ramp down of all channels of all modules in the crate.

Important: The inhibit functionality is an option for certain HV modules such as the EHS series. It isn't available for other series such as EBS. As the module refuses this command, setting the bits 12 and 13 in outputSupervisionBehavior might result in a timeout.

For HV channels the values serving as reference for the current trip is the value defined by the SNMP variable outputCurrent.

6.6.2 Reading and writing the delayed trip time (channel variable outputTripTimeMaxCurrent)

The variable "outputTripTimeMaxCurrent " can be read and written by using the net-snmp command line tools snmpget and snmpset. The commands

> snmpget -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103 outputTripTimeMaxCurrent .<channel name>

> snmpget -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.3.2.1.27.<channel index>

will return the delayed trip time as integer number. It gives a time delay in milliseconds. It can be set by the commands

> snmpset -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103 outputTripTimeMaxCurrent .<channel name> i <value>

> snmpset -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.3.2.1.27.<channel index> i <value>

The allowed values for <value> are given in the hardware user manual. The placeholder <channel name> has to be replaced by “u0” … “u1999”. <channel index> has to be replaced by the integer values 1 … 2000. See the chapter on crate structure for more details.

6.6.3 Reading the nominal voltage (channel variable outputConfigMaxTerminalVoltage)

The variable “outputConfigMaxTerminalVoltage” can be read by using the net-snmp command line tool snmpget. The commands

> snmpget -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103
outputConfigMaxTerminalVoltage.<channel name>

> snmpget -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.3.2.1.22.<channel index>

will return the nominal voltage as floating point number. The placeholder <channel name> has to be replaced by “u0” … “u1999”. <channel index> has to be replaced by the integer values 1 … 2000. See the chapter on crate structure for more details.

6.6.4 Reading the nominal current (channel variable outputConfigMaxCurrent)

The variable “outputConfigMaxCurrent” can be read by using the net-snmp command line tool snmpget. The commands

> snmpget -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103
outputConfigMaxCurrent.<channel name>

> snmpget -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.3.2.1.23.<channel index>

will return the nominal current as floating point number. The placeholder <channel name> has to be replaced by “u0” … “u1999”. <channel index> has to be replaced by the integer values 1 … 2000. See the chapter on crate structure for more details.

6.7 Group Configuration and Actions

The “groupsSwitch” variable allows to perform actions for all channels associated to a group.

6.7.1 Executing actions for an entire group (group variable groupsSwitch)

For HV modules this variable allows to trigger or clear an emergency off. It is the only possibility to configure the “fine adjust” and the “kill” options. It also allows to clear events.

The group index 64 addresses all HV modules at once. The group index 128 addresses all LV modules and channels. The group index 0 addresses all HV and LV modules. Other group indices can be configured for LV channels by setting the “outputGroup” variable.

Reading the groupsSwitch variable doesn’t make much sense. The command always returns -1. The command line tool snmpset allows to write to this variable:

> snmpset -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103 groupsSwitch.<group index> i
<value>

> snmpset -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.3.4.1.9.<group index> i <value>

The placeholders <group index> must be replaced by 0 to access all modules, by 64 to address HV modules, by 128 for all LV modules and by other valid group indices to address groups defined for a set of LV channels.

The following actions are performed for different values for the placeholder <value>:

0	All channels of the group will be switched off.
1	All channels of the group will be switched on.
2	The “emergency off” is cleared for all HV channels (group 0 or 64 only).
3	The “emergency off” signal is sent to all HV channels (group 0 or 64 only).
4	The “kill” option is enabled for all HV modules (group 0 or 64 only).
5	The “kill” option is disabled for all HV modules (group 0 or 64 only).
6	The “fine adjust” option is disabled for all HV modules (group 0 or 64 only).
7	The “fine adjust” option is enabled for all HV modules (group 0 or 64 only).
10	All channel events are cleared for all HV modules (group 0 or 64 only).

Important: Please note that if “kill” is enabled, the voltage ramp for the concerned HV module is limited.

CHAPTER 7 MEASUREMENTS

The following variables allow to retrieve measured values from an MPOD crate, it's modules and channels.

7.1 Crate Measurements

At crate level the voltages and currents of the auxiliary power supplies can be measured. These auxiliary power supplies are mostly used to provide a 24V power supply to the inserted HV power supply modules.

7.1.1 Reading the auxiliary power supply voltage (crate variable psAuxiliaryMeasurementVoltage)

The variable "psAuxiliaryMeasurementVoltage" can be read by using the net-snmp command line tool snmpget. The commands

> snmpget -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103
psAuxiliaryMeasurementVoltage.<aux name>

> snmpget -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.6.5.1.3.<aux index>

will return the auxiliary power supply voltage as floating point number. The placeholder <aux name> has to be replaced by "u0" … "u7". <aux index> has to be replaced by the integer values 1 … 8. This might vary depending on the number of auxiliary power supplies. See the chapter on crate structure for more details.

7.1.2 Reading the auxiliary power supply current (crate variable psAuxiliaryMeasurementVoltage)

The variable "psAuxiliaryMeasurementCurrent" can be read by using the net-snmp command line tool snmpget. The commands

> snmpget -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103
psAuxiliaryMeasurementCurrent.<aux name>

> snmpget -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.6.5.1.4.<aux index>

will return the auxiliary power supply current as floating point number. The placeholder <aux name> has to be replaced by "u0" … "u7". <aux index> has to be replaced by the integer values 1 … 8. This might vary depending on the number of auxiliary power supplies. See the chapter on crate structure for more details.

7.2 Module Measurements

Only HV modules allow to measure the temperature of the module. LV modules provide a temperature for each channel.

7.2.1 Reading the module temperature (module variables moduleAuxiliaryMeasurementTemperature0 ... moduleAuxiliaryMeasurementTemperature3)

The variables "moduleAuxiliaryMeasurementTemperature0" … "moduleAuxiliaryMeasurementTemperature3"can be read by using the net-snmp command line tool snmpget. The commands

> snmpget -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103
moduleAuxiliaryMeasurementTemperature0.<module name>

> snmpget -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103
moduleAuxiliaryMeasurementTemperature1.<module name>

> snmpget -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103
moduleAuxiliaryMeasurementTemperature2.<module name>

> snmpget -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103
moduleAuxiliaryMeasurementTemperature3.<module name>

> snmpget -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.3.6.1.12.1.<slot index>

> snmpget -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.3.6.1.12.2.<slot index>

> snmpget -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.3.6.1.12.3.<slot index>

> snmpget -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.3.6.1.12.4.<slot index>

will return the module temperature as floating point number. If less than 4 temperature sensors are present, some of the returned values might be 0. The placeholder <module name> has to be replaced by "ma0" … "ma9". <slot index> has to be replaced by the integer values 1 … 10. See the chapter on crate structure for more details.

7.3 Channel Measurements

Several SNMP variables allow to measure channel parameters. Output voltages and currents can be measured. For LV channels the voltages can be measured at the terminal and at the sense lines. LV channels also inform on the temperature of the individual channels.

7.3.1 Reading the voltage on the sense lines (channel variable outputMeasurementSenseVoltage)

The variable "outputMeasurementSenseVoltage" can be read by using the net-snmp command line tool snmpget. The commands

> snmpget -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103
outputMeasurementSenseVoltage.<channel name>

> snmpget -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.3.2.1.5.<channel index>

will return the voltage measured on the sense lines as floating point number. The placeholder <channel name> has to be replaced by "u0" … "u1999". <channel index> has to be replaced by the integer values 1 … 2000. See the chapter on crate structure for more details.

Important: For HV channels the returned value is identical to the voltage on the output terminals. This is also true for LV channels if no sense lines are used.

7.3.2 Reading the voltage on the output terminal (channel variable outputMeasurementTerminalVoltage)

The variable "outputMeasurementTerminalVoltage" can be read by using the net-snmp command line tool snmpget. The commands

> snmpget -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103
outputMeasurementTerminalVoltage.<channel name>

> snmpget -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.3.2.1.6.<channel index>

will return the voltage measured on the output terminals as floating point number. The placeholder <channel name> has to be replaced by "u0" … "u1999". <channel index> has to be replaced by the integer values 1 … 2000. See the chapter on crate structure for more details.

Important: For HV channels the returned value is identical to the voltage on the sense lines. This is also true for LV channels if no sense lines are used.

7.3.3 Reading the channel current (channel variable outputMeasurementCurrent)

The variable "outputMeasurementCurrent" can be read by using the net-snmp command line tool snmpget. The commands

> snmpget -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103
outputMeasurementCurrent.<channel name>

> snmpget -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.3.2.1.7.<channel index>

will return the current measured on the outputs as floating point number. The placeholder <channel name> has to be replaced by "u0" … "u1999". <channel index> has to be replaced by the integer values 1 … 2000. See the chapter on crate structure for more details.

7.3.4 Reading the channel temperature (channel variable outputMeasurementTemperature)

The variable "outputMeasurementTemperature" can be read by using the net-snmp command line tool snmpget. The commands

> snmpget -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.103
outputMeasurementTemperature.<channel name>

> snmpget -v2c -c guru 192.168.0.103 iso.3.6.1.4.1.19947.1.3.2.1.8.<channel index>

will return the temperature measured on the channels as integer number. The placeholder <channel name> has to be replaced by "u0" … "u1999". <channel index> has to be replaced by the integer values 1 … 2000. See the chapter on crate structure for more details.

CHAPTER 8 ADDITIONAL EXAMPLES

The following examples illustrate how to perform particular tasks by reading and writing SNMP variables to an MPOD crate.

8.1 Using External Sense Lines For LV Channels

MPOD crates allow to use sense lines on low voltage channels. They can be configured using the SNMP protocol.

8.2 Configuring Current Trips For HV Channels

Critical experiments can be protected by using the current trip functionality on high voltage modules and their channels. This option can be configured using the SNMP protocol.

There are 2 ways to configure a current trip. The first consists in setting the “kill enable” flag by writing the SNMP variable “groupsSwitch” for all HV modules. Now a maximum current can be defined individually for each HV channel by writing the SNMP variable “outputCurrent”.

The corresponding channel will be shut down when the maximum current is reached.

Configuring a “delayed trip” allows a much better control. It should be preferred over the “kill enable” option. Proceed as follows:

- define a maximum current by writing the SNMP variable “outputCurrent”.
- define a delayed trip time interval by writing the SNMP variable “outputTripTimeMaxCurrent”.
- define a delayed trip action by writing the variable “outputSupervisionBehavior”. The bits 6 and 7 control the action that will be triggered.

The selected delayed trip cation will be triggered whenever Imeas reaches Iset for the defined delayed trip time interval.

Important: please note that configuring current trips limits the voltage ramp for the concerned module. This concerns the “kill” option and the delayed trip.

8.3 Using The Inhibit Functionality For HV Channels

The inhibit functionality is only available for some HV modules. These modules have a 9-pin D-sub connector on their front panel.

The inhibit is triggered by applying a TTL signal to the pins of the D-Sub connector. The module’s data sheet gives more details on how this has to be done. In general each channel has a separate inhibit pin.

If an inhibit is triggered by hardware, the concerned channel reacts according to the bits 12 and 13 set in the SNMP variable outputSupervisionBehavior. That means that a hardware interlock signal can either be ignored, it can be used to ramp the channel down, it can switch the channel off without ramp, or it can be configured to switch the entire module off.

The channels “inhibit” flag is set when a hardware inhibit occurs.

Once an inhibit occurred, the module's flags have to be cleared before switching the channel on again.

Important: The module's inhibit must not be confounded with the crate controller's interlock. The controller's interlock function must be configured using MUSEcontrol. The flags in outputSupervisionBehavior have no effect on it. If an interlock is triggered all channels of all modules are ramped down.

CHAPTER 9 TROUBLESHOOTING

Programming with SNMP can be difficult if advanced options are used. As flags and functionalities interact, it is often difficult to make sure a control software is stable, simple to use and to maintain. It is also difficult to give a meaningful feedback and strategies for errors that might occur while an experiment is running, but that can't be tested when the software is implemented.

This chapter gives some hints on how to deal with these problems.

9.1 Use An Existing Control Software

If you haven't used MPOD crates before, it helps a lot to familiarize with it by paying around with existing graphical control software. The following control software solutions are available:

- MUSEcontrol: this Windows software works over a USB connection. It can be downloaded from http://wiener-d.com. Navigate to the MPOD product page. The download link is available from "Resources & Downloads". Other downloadable material is also available.
- Easy LV | HV: This control software has mainly been developed for MPOD users. It can be obtained from these web sites:
  - Physical Instruments for users in France: https://www.physical-instruments.fr/
  - Analog Flavor for international users: http://www.analogflavor.com
- isegSNMPcontrol: this control software can be downloaded from http://iseg-hv.com. Navigate to the support page and find the latest version in the file browser on this page.

9.2 Use SnmpWalk

Before starting to configure an MPOD and it's modules it makes sense to perform an SnmpWalk and to redirect the output to a file. This can be repeated several times. By comparing these files it is a lot simpler to restore a previous configuration if something goes wrong.

These files can also be searched for two different channel names if one channel doesn't behave as expected. Some flags or configuration values might be different for these two channels. This can explain a different behavior.

Take a look at the flags. Proceed as follows:

- Switch all channels off.
- Clear all channel flags.
- Clear all module flags.
- Perform an SnmpWalk, dump the output to a first file.
- Switch the channel that causes problems on.
- Perform a second SnmpWalk and generate a new file.

- Compare the two files. Start with the channel and module flags.
- If another channel behaves as expected, repeat the procedure for this channel and compare.

9.3 Keep It Simple

Most experiments don’t require a complicated setup. If it is sufficient to set up everything once and for all, the above control software can be used for these purposes.

If everything is set up a reduced set of instructions will be sufficient to run an experiment. Side effects are limited to a minimum.

