s::can
Intelligent. Optical. Online.

Manual

moni::tool V3.1

September 2018 Release

ABOUT THIS MANUAL

Structure

This manual is structured into four parts:

- Part A – General Information
- Part B – Getting Started
- Part C – Operation of *moni::tool*
- Part D – Additional Information

General Information comprises safety guidelines, background information on s::can products as well as information regarding product updates. Furthermore, and of major relevance for using this manual, an introductory chapter on the used terminology defines how certain terms are used in this manual and with respect to s::can products in general.

Getting Started focuses on the steps from unpacking the product to the initial start-up. The first chapter addresses technical product information on the intended use as well as on the functional principles of key product components. Following a chapter on the installation of the system finally the initial start-up is explained.

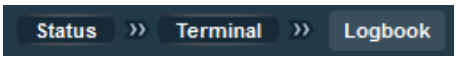
The main part of the manual addresses ***Operation of moni::tool***. Its structure follows the graphical user interface of *moni::tool* and all components are addressed as they are accessed when working through the various software menus and items.

Finally, ***Additional Information*** provides insight into various issues of function checks, troubleshooting as well as into using the VNC software for taking over *con::cube* display, touch and keyboard from remote.

Usability

Terms in this document which are marked italic and underlined (*example*) indicate that a corresponding item can be found on the display of your terminal or on your s::can product. Terms which are formatted italic only refer to proprietary notations of or within s::can products.

For easy navigation in areas of *moni::tool* with a deep hierarchical order, i.e. several levels of windows, on top of the screens a “navigation line” indicates the level of the currently active screen. The elements of this line can also be used for navigating to the higher software levels.



The concept of *Service Levels* in *moni::tool* helps adapting the information on some screens to actual needs. This is indicated in the respective chapter with →*Service Level* / *Expert*

While this manual focuses on *moni::tool* version 3.0, it basically covers version 2.5 as well since operational aspects have not been affected by this version step. However, due to the change from a Microsoft Windows© operating system (V 2.5) to a Linux-based platform (V 3.0) activities which are implemented on the level of the operating system need to be handled differently. These differences are indicated in the respective chapters of this manual.

Disclaimer

In spite of careful elaboration this manual may contain errors or incompletion. s::can does not assume liability for errors or loss of data due to such faults in the manual. The original manual is published in English and German by s::can. This original manual serves as the reference in case discrepancies occur in versions of the manual after translation into third languages.

This manual, at the time of its publication (see release date printed on the top of this document), concerns the s::can products listed in the chapter *Product Packages* of *Part A* in this manual. Information and technical specifications regarding these items in s::can manuals from earlier release dates are herewith replaced by this manual.

TABLE OF CONTENTS

PART A – GENERAL INFORMATION

TERMINOLOGY

Terminal	*Terminal* refers to the hardware which is used to operate the monitoring station and to which the various sensors are connected. In the framework of this manual this will usually mean a *con::cube* running *moni::tool*.
Sensor	All devices producing the data which are then further processed as measurement values are referred to as *sensor*, irrespectively whether it is complex hardware such as a *spectro::lyser* or a simple device such as a temperature sensor.
Raw values	*Raw values* are the data as they are originally produced by the respective sensor, without any further processing. Very often this will be measurements of mV or mA which have no deeper meaning for a standard user and are hence usually not accessible.
Measured values	*Measured values* are delivered by the respective device to the user. This usually involves the physical *sensor* measuring the *raw data* and some data processing relating those raw values to the typical unit of the respective parameter (e.g. mV → °C). The second step is referred to as *calibration*.
Clean values	*Clean values* are *measured values* which have been further processed by the *vali::tool* software.
Local / Remote Access	Basically a monitoring station can be accessed and operated either locally or remotely. *Local access* means that input devices (touch screen, optional mouse, keyboard using the USB interface) and output devices (display) of the station hardware are used. *Remote access* means that the web-server forming a key component of *moni::tool* (and actually producing and providing the various *moni::tool* screens) is not accessed through the built-in web-browser but through any standard web-browser running on another computer. The connection between the "remote" computer and the monitoring station (*con::cube*) which facilitates the required internet communication between the two devices is based on http and uses port 80. However, It is important to note that whenever this manual talks about services to be available locally only, the respective activities can be implemented via VNC as well (please refer to section 3 of Part D for further information).

SAFETY GUIDELINES

Installation, electrical connection, initial set-up, operation and maintenance of any s::can product as well as complete s::can measuring systems must be performed by qualified personnel only. Such qualified personnel need to be trained and authorized by the plant operator or by s::can for these activities. The qualified personnel must have read and understood this manual and have to follow the instructions contained in this manual.

For setting up s::can measuring systems correctly, the manuals for

- the terminal (*con::lyte*, *con::cube* or PC / notebook with *con::nect*),
- the operating software *(ana::lyte*, *ana::pro* or *moni::tool*) as well as
- the connected sensors

have to be consulted.

The operator is responsible for obtaining required local operating permits and for complying with the joint constraints associated with these. Additionally, local legal requirements have to be observed (e.g. regarding safety of personnel and means of labor, disposal of products and materials, cleaning, environmental constraints). Before putting the measuring device into operation, the operator has to ensure that during mounting and initial start-up – in case they are executed by the operator himself – the local legislation and requirements (e.g. regarding electrical connection) are observed.

All s::can products are leaving our factory in immaculate technical and safety conditions. However, inappropriate or not intended use of products can cause danger! The manufacturer is not responsible for damage caused by incorrect or unauthorized use.

Any kind of manipulation of the instrument is strictly prohibited - except for the activities described in this document. Conversions and changes to the device must not be made, otherwise all certifications and guarantee / warranty become invalid. For details regarding guarantee and warranty please refer to our general conditions of business.

Special Hazard Warning

As s::can measuring systems are frequently installed in industrial and communal waste water applications, special care has to be taken during mounting and demounting of the system as parts of the device might be contaminated with dangerous chemicals or pathogenic germs. All necessary precautions should be taken to prevent endangering of one's health during work with the measuring device.

PRODUCT PACKAGES

The following variants and extension packages of the *moni::tool* software are available. For detailed information about software variants please refer to the s::can homepage www.s-can.at or directly to www.moni-tool.at**.**

moni::tool License Options	free	one time license fee											
	S-11-04-moni	S-11-08-moni	S-11-24-moni	S-11-64-moni	S-11-data-export	S-11-free-formula	S-11-SMS	S-11-autosampler	S-11-basic-PLC	S-11-camera	S-14-vali	S-15-ana	S-20-MVA
Basic Features	•	•	•	•									
4 Parameters	•												
8 Parameters		•											
24 Parameters			•										
64 Parameters				•									•
Automatic data transfer (via SSH, FTP, TML)					•								•
Configurable mathematical formula						•							•
SMS notification							•						•
Auto sampler feature								•					•
Basic PLC functionality (time control, pulsing, custom bits)									•				•
Camera input										•			•
vali::tool											•	•	•
ana::tool (includes vali::tool)												•	•
Affordable license for all moni::tool features, vali::tool and ana::tool													•

* Every s::can con::cube terminal is delivered with the basic moni::tool version for four parameters at no extra costs!

Annual support and carefree packages are available. Please contact your s::can Sales Partner regarding these products.

SCOPE OF DELIVERY

Upon receipt, please check the received consignment for completeness on the basis of the delivery note and check for any possible damage incurred during shipping. Please inform the delivering dispatcher and s::can immediately in case of any damages in transit.

The following parts should be included in the delivery:

- s::can software *moni::tool*, already pre-installed on your terminal (item-no. S-11-xx-moni),
- s::can manual *moni::tool*.

The following parts might be included in the delivery if ordered as an option:

- s::can software *vali::tool* and *ana::tool*, already pre-installed on your terminal (item-no. S-14-vali and S-15-ana)

In case of incompleteness please contact your s::can sales partner as soon as possible!

PRODUCT UPDATES

The manufacturer reserves the rights to implement, without prior notice, technical developments and modifications in the light of continuous product care. Software updates will be made available in the form of update packages that can be downloaded and installed on the s::can terminal.

PART B – GETTING STARTED

Intended Use

The s::can software *moni::tool* is intended for operation of s::can monitoring stations. Based on an open, innovative, location-independent architecture, it uses a local database for collecting all data and a web server for visualization and operation.

moni::tool supports the following functions:

- All sensor and monitoring station management tasks (e.g. initialization, configuration, calibration)
- Automatic restart and continuation of measurement after interruption of power supply
- Data management, visualization and export of results (values, time series, UV/Vis-spectra in 2D/3D)
- Indication of sensor status and performance
- Advanced system for quality assurance and quality control (QA/QC)
- Logging and tracking of all maintenance actions performed on the monitoring station (e.g. changes of sensor settings or components, calibration and any other activities)
- Automatic feedback to the operator using the incoming and logged information
- Online calculation of customized parameters on the terminal
- Support of all terminal interfaces
- PLC (*Programmable Logic Controller*) functions

For the *moni::tool* basic module the following add-ons are available:

- *vali::tool* - advanced module for real time data validation
- *ana::tool* - advanced module for real time event detection
- higher number of supported parameters
- advanced features - advanced module for data export (Auto-Export, TML), see software variants above for complete list.

Functional Principle

moni::tool

The *moni::tool* software package consists of several components. The central component of the system is a PostgreSQL database in which all measurement data as well as all configurations and settings are stored. Measurement data are obtained from all s::can sensors through specific drivers that communicate with the devices using either the Modbus protocol or a mA signal.

Information is presented to the user via webpages that are generated by a webserver on the s::can terminal. On these webpages measurement results and status information can be viewed, configurations can be changed and service actions can be performed. When the s::can terminal is connected to a network, these webpages can also be accessed from any computer with access to the terminal. In this way true remote control of the station is possible.

All implemented tools are programmed in Java and the platform independent client runs on any Java enabled web browser. This means that sensors and stations can be accessed from any suitable device that is connected to the internet.

Tools, devices and protocols can easily be integrated or customized at any time.

vali::tool

vali::tool is s::can's data validation module. Its task is to automatically detect, mark and (optionally) clean-up untrustworthy data. This data evaluation provides valuable quality information on the individual measurements performed by the system.

The results from this online validation are utilized in several ways. Firstly, the user can be provided with indications that a sensor requires maintenance or that malfunctions have been detected. And secondly, the marking of questionable results enables the alarm tool (*ana::tool*, see below) to ignore data of insufficient quality and hence improve its accuracy significantly.

The use of automatically cleaned measurement results is of special interest for process control where a loss of signal can lead to undesirable consequences in the process while intelligent data clean-up allows the controls to continue working properly.

For data validation a number of simple but robust statistical methods such as the detection of outliers, discontinuities or noise have been implemented.

ana::tool

ana::tool is s::can's event detection module. It is designed to evaluate data cleaned by the validation module regarding its normality and, in cases of identified significant deviations, to trigger specific alarms. While *ana::tool* has been optimized for utilizing multi-dimensional spectral data, it will work just as well with single or multiple one-dimensional inputs from conventional sensors. However, the integration of spectral data provides a much more complete picture of water quality than can be obtained through single parameters.

The applied methods for identifying alarm situations comprise static thresholds, dynamic thresholds, pattern recognition, spectral alarms and cumulative alarms.

Installation

The *moni::tool* software has been developed for its use on the s::can *con::cube* hardware. It has already been installed on any purchased terminal and no separate installation of the software is required.

Update or recovery packages are available on the s::can portal (www.s-can.at/services/customer-portal). Please refer also to the manual delivered with the respective s::can terminal.

Requirements Web Browser

The local browser on your s::can terminal is preinstalled and fulfils all requirements for optimal performance. For using *moni::tool* remotely, the browser on the PC / notebook has to meet the following specifications:

- Mozilla Firefox >= V3.6 (no compatibility mode)
- Google Chrome >= V14
- Opera >= V8.9 (equates V11.5)
- MS Internet Explorer >= V9 (no compatibility mode)
- Apple Safari >= V5

Requirements Spectrometer probes

For being operated with *moni::tool*, spectrometers have to be equipped with firmware V1.0.z (V0133) or higher. s::can recommends to always use the most actual firmware version available.

Conditions for Use / Licensing Terms and Conditions

For each terminal an individual license file is needed which is already installed at delivery according to the ordered software package. For an upgrade of the *moni::tool* software (see section *Product Packages* above) a new license file can be downloaded to the terminal.

The license text can be found in *moni::tool* in the ***Help*** tab. Licenses are deemed accepted when an s::can product is used.

System Set-up

The s::can terminal is delivered with a pre-installed operating software *moni::tool*. Whenever the terminal is newly connected to power supply (please refer to the terminal manual), *moni::tool* will start automatically. Depending on the type of the terminal it will take 1 - 2 minutes for the start-up process to be completed.

After the initial start-up the monitoring station needs to be configured and sensors need to be installed. For this purpose a list of tasks has to be performed as is indicated below.

Task	*moni::tool* Path	Manual page
Set **terminal language**	Service / Terminal / Terminal Language Bottom Menu / flag symbol (external browser only!)	52 16
Log-in	Bottom Menu / User icon or Tab *Service* (Default password for user "Administrator" is "admin1")	16, 34
Set **time zone** and **network time server**	Service / Terminal / Date & Time / Time Configuration	52
Set **date and time**	Service / Terminal / Date & Time / System Time	52
Set **station name**	Service / Terminal / Settings / Station	43
Set **computer name** and **network configuration**	Service / Terminal / Network	42
Configure **users**	Service / Terminal / Settings / User Accounts	43
Install **sensors**	Service / Add new sensor	65
Configure the **parameter order** in which they are displayed	Service / Terminal / Parameters	45
Configure **measurement interval**	Service / Terminal / Measurement	39
Configure automatic **cleaning devices**	Service / Cleaning Devices	85
Configure ***vali::tool***	Service / Terminal / Parameters	49
Configure **outputs**	Service / Outputs / • Analog Outputs • Digital Outputs • Fieldbus Outputs • SMS Notification • Auto-sampler Service / Outputs / File Output / • File Generation / Setup • FTP(SSH) Transfer	71 76
Collect samples	Service / Samples & Calibration / Taking Samples	37
Check sensor readings, **calibrate** if necessary	Service / Sensors / Calibrate sensor	62

Once the system has been operated successfully for some days and validated results have been produced, the set-up process can be completed.

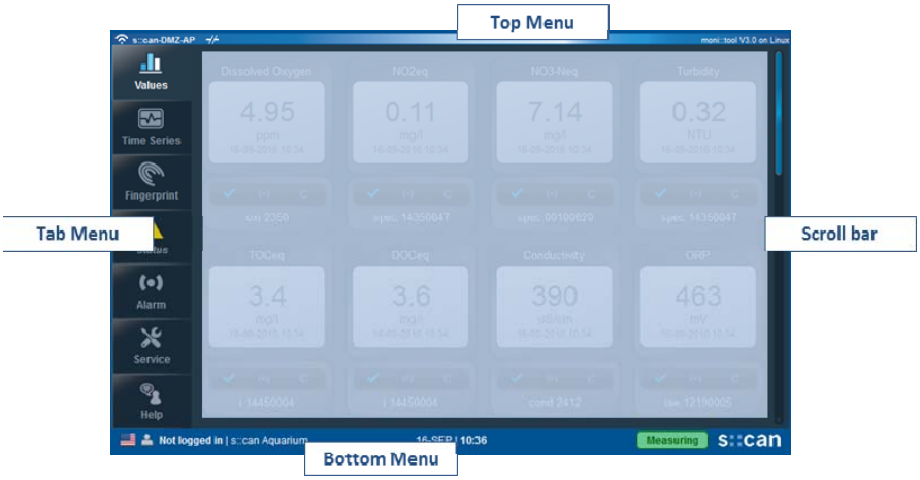
Configure ***ana::tool***	Service / ana::tool	68

Please refer to the respective chapters of this manual for details!

⚠ Typically, some of the tasks listed above will be implemented only during the initial system set-up. However, other activities will need to be repeated on a regular basis. Hence, the list above can also be seen as an overview on operations and maintenance and as a guide to the respective sections of *moni::tool* and of this manual.

PART C – OPERATION OF *MONI::TOOL*

0 Permanently Visible Components



These *moni::tool* elements are shown as part of the start screen but are available on all other screens as well.

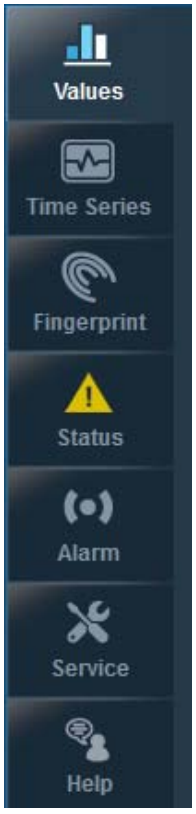
They form the backbone of *moni::tool* navigation and provide access to all its functions.

Their respective functionalities are explained below.

0.1 Tab Menu

The seven Tab Menu icons are the main navigation tool for accessing the different views and possibilities of the *moni::tool* software. Furthermore, they also highlight key status information of the monitoring station, as shown for instance for the *Status* tab below.

The currently selected tab is highlighted, as shown for *Values* below.



Values is *moni::tool*'s default view and displays the current readings as well as information on the measurement quality for all installed parameters and provides additional functions.
*(see **Chapter** 1)*

Time Series is the main view to assess recorded measurements through displaying time series of parameter results, including various manipulation options.
*(see **Chapter** 2)*

Fingerprint is the tab specialized on analysing absorbance spectra as produced by spectro::lyser.
*(see **Chapter** 3)*

Status provides actual information on all the measurement system's components. It covers various issues and its icon blinks yellow if an error is detected.
*(see **Chapter** 4)*

Alarm provides access to the component designed to detect irregularities within the monitored media. The icon blinks yellow if an alarm has been triggered.
*(see **Chapter** 5)*

Service aggregates all aspects of system configuration as well as of setting specific controls. The icon blinks yellow if the monitoring station has been set into service mode.
*(see **Chapter** 6)*

Help addresses various formal aspects and provides background information on the system and its functionality. However, the use of this tab can also be adapted to user preferences.
*(see **Chapter** 7)*

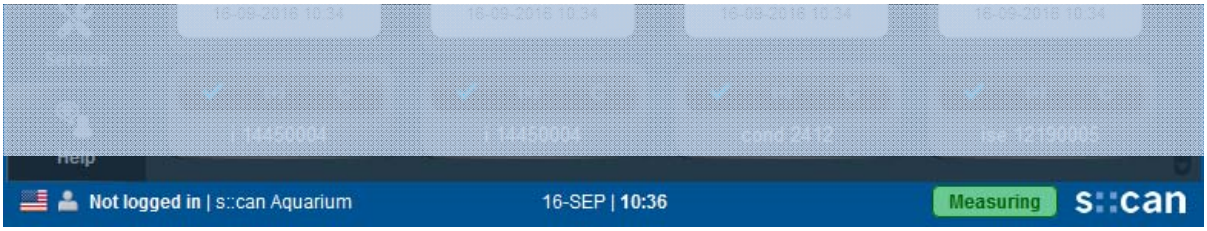
⇨ *Available **Quick reference guides**:*

Manuals	*moni::tool*	note_monitool_overview

Please refer to the s::can Customer Portal!

0.2 Bottom Menu

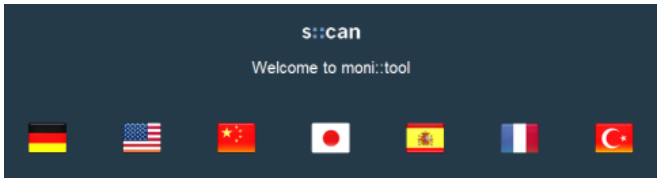
The *Bottom Menu* shows some key information about the monitoring station's status and provides access to some key functions. An overview on the individual elements is provided below.



No.	Element	Information	Push-Function
1	**Language**	Language selected for the remote browser session indicated by the flag symbol	Change the language used for the remote browser *moni::tool* session
2	**User**	User currently logged-in	log-in / log-out; *con::cube* shutdown/re-start; change Service Level
3	**Station name**	Name of the monitoring station	-
4	**Date & time**	System date and time	Display date / time of the last measurement
5	**Activity**	Activity the system is currently performing (partly incl. remaining time)	-
6	**Logo**	s::can logo	Pushing the logo will show information about the terminal and *moni::tool* (as *Help* / *Version Info*)

Details on the individual elements and their respective push-functions are presented in the following sections of this chapter.

0.2.1 Language Selection



Select one of the offered languages by pushing its flag symbol.

This selection influences only the language used for the current remote session via an external browser. It has no impact on the language settings for the terminal itself!

⇨ *Relevant **Manual sections**:*

Terminal language selection	Service / Terminal / Terminal Language	6.4.7, p. 52

0.2.2 User

Beside the *Bottom Menu*'s User icon the current log-in status is shown. When no user is logged-in, pushing the text field opens the log-in dialogue.

Usually, in this dialogue a username and the respective password can be entered. Alternatively, the options for a simple (only the password needs to be entered) or for an automatic log-in can be activated in the *Service* tab.

When a user has been logged-in already the respective user name is displayed (e.g. *Administrator*) and pushing the name provides access to the following options:

- *Terminal*: *Shutdown* (switches the *con::cube* off, app. 15 s; before a restart the power needs to be disconnected for app. 60 s) / *Reboot* (re-starts the *con::cube*, app.3 minutes)
- *Logout*: Log-out of the current user (happens also automatically after an inactive period)

- *Service Level*: *Basic* / *Advanced* / *Expert* (the available levels depend on the active user profile)

⇨ *Relevant **Manual sections**:*

User administration	Service / Terminal / Settings / User Accounts	6.4.4, p. 43
Log-in mode	Service / Terminal / Settings / Login Mode	6.4.4, p. 43

0.2.3 Station Name

The name of the station allows the quick identification of the respective system which is especially relevant when several stations are accessed remotely.

⇨ *Relevant **Manual sections**:*

Setting the station name	Service / Terminal / Settings / Station	6.4.4, p. 42

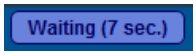
0.2.4 Date & Time

The *Bottom Menu* displays the current date and time as configured for the station and provides quick information on the time of the last measurement.

⇨ *Relevant **Manual sections**:*

Setting date and time	Service / Terminal / Date & Time / Time Configuration	6.4.6, p. 52
Setting the time zone	Service / Terminal / Date & Time / System Time	6.4.6, p. 52

0.2.5 Activity



The Activity element indicates the activity status of the station.

This is helpful when an automatic measurement program is active for understanding at which point of the measurement cycle the system stands and how long the respective phase will still last.

⇨ *Relevant **Manual sections**:*

Configure automatic cleaning devices	Service / Cleaning Devices	6.10, p. 85
Configure the measurement interval	Service / Terminal / Measurement	6.4.1, p. 39

0.2.6 Logo

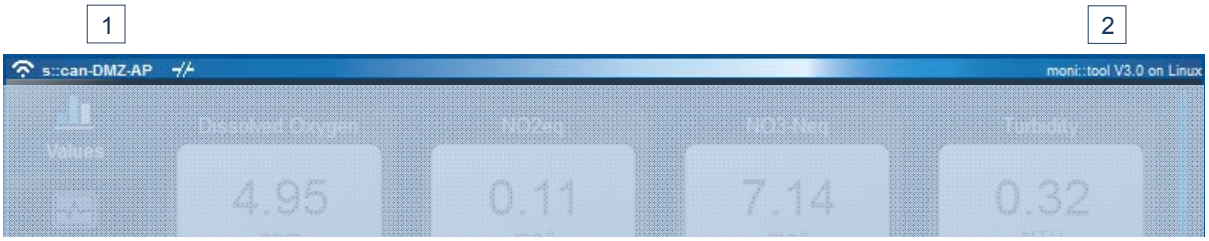
The s::can logo provides direct access to information on the used terminal, the installed software and the network, which is part of the Help tab.

⇨ *Relevant **Manual sections**:*

Version Info	Help / Version Info	7.1, p. 87

0.3 Top Menu

The Top Menu provides information about the station's connection status and the current software version.



No.	Element	Information	Push-Function
1	**Connection status**	Active network connection	Access to network configuration (*admin rights required*)
2	**Software version**	Software version and operating system	-

0.3.1 Connection Status

The first part of the element symbolizes the physical type of the network connection:

Local Area Network (LAN)

Wireless LAN

Modem

Virtual Private Network (VPN)

Beside the network-type symbol the name of the network (if available) or of the used settings-profile and the connection status are shown.

con::cube not connected to network using LAN adapter

con::cube connected to network using LAN adapter

JAHome

con::cube connected to network using WLAN adapter; signal strength 1 (poor signal), 2 (weak signal) or 3 (good signal). In addition the WLAN network name is displayed

T-Mobile A

con::cube connected to network using internal modem (optional); signal strength 0 (no signal), 1 (very poor signal), 2 (poor signal), 3 (weak signal), 4 (good signal) or 5 (perfect signal). In addition the Sim-card provider name is displayed.

The available options for configuring the respective settings can be accessed directly through this element. However, this will require a user with administrator rights.

*Relevant **Manual sections**:*

Configure network settings	Service / Terminal / Network	6.4.3, p. 42
User administration	Service / Terminal / Settings / User Accounts	6.4.4, p. 43

0.3.2 Software Version

This element of the Top Menu shows the installed *moni::tool* version and the *con::cube*'s operating system.

0.4 Scroll Bar

The Vertical Scroll Bar can always be found on the right side of the frame. It is highlighted when it is active for navigating through views which are longer than the standard *moni::tool* screen.

The *con::cube* touchscreen also offers the possibility to navigate by using typical gestures ("*swiping*").

1 Values

The *Values* screen provides a detailed overview of the last parameter results as well as status information from all configured sensors.

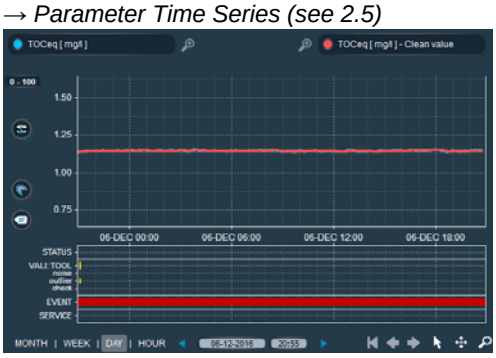
The information for up to eight parameters can be displayed simultaneously, the respective indices run from top left (=1) to bottom right.

In situations where more than the displayed number of parameters are monitored the scroll bar on the right screen side allows moving down and up for viewing the other parameters.

⇨ *Relevant **Manual sections**:*

Configure parameter order	Service / Terminal / Parameters / Parameter Order	6.4.5, p. 46
Configure Values screen	Service / Terminal / Display / Values Layout	6.4.8, p. 53
Configure logged-out display	Service / Terminal / Display / Parameter Display	6.4.8, p. 53

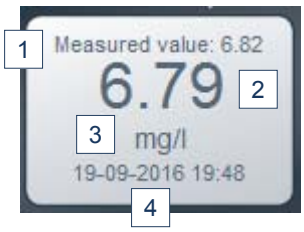
For each configured parameter a display module is defined, which consists of a main upper part showing names and values and a lower part providing additional information.

No.	Element	Information	Push-Function
1	**Parameter name**	Name assigned to this parameter	-
2	**Parameter results**	Last measurement results	→ *Parameter Time Series (see 2.5)* 
3	**Parameter info**	Additional information on how results are produced and used	→ *Parameter Details (see 1.3)* 

1.1 Parameter Name

The name for a parameter can be chosen freely and is independent from the name of the respective sensor (see 6.4.5). It is used for display in *moni::tool* as well as for identifying entries in the various log files.

1.2 Parameter Results



No.	Element	Information	Push-Function
1	**Measured value**	Reading as delivered by the sensor (visible only when "clean value" is displayed as parameter reading)	→ *Parameter Time Series* *(see 2.5)*
2	**Parameter reading**	Reading of the parameter (either measured or clean value)	
3	**Unit**	Physical unit of the parameter	
4	**Time stamp**	Date and time of the measurement	

1.2.1 Measured Value

This is the value which has actually been produced by the respective sensor, i.e. without applying *vali::tool*'s cleaning algorithms (which is also indicated in *Parameter Info* - see below). Accordingly, it is displayed only if a cleaned value is displayed,

1.2.2 Parameter reading

If the clean value as calculated by *vali::tool* is selected for display, the cleaned value is shown as parameter reading which might differ from the *Measured Value*, which is then displayed above (see also 1.3.3).

1.2.3 Unit

The physical unit of a parameter can be assigned in the parameter configuration.

1.2.4 Time Stamp

The displayed the date and the time show when the measurement of the displayed parameter reading took place. It is also used to identify the respective measurement in the results data base, e.g. for display in *Time Series*.

1.2.5 Functions

Besides displaying the information as described above, *Parameter Results* has two more functions:

- The **background color** provides information on the measurement quality.

White	Displayed measurement okay
Red	An error occurred – see status message
Grey	The displayed result is older than three measurement intervals

- **Pushing** the display field switches directly to a ***specific view*** of *Time Series* and will display measured and cleaned values of the respective parameter over the last 24 hours (see also 2.5)

⇨ *Relevant **Manual sections**:*

Parameter settings	Service / Terminal / Parameters	6.4.5, p. 45
Parameter validation with *vali::tool*	Service / Sensors / Parameters	6.5.4, p. 61
Results analysis with *Time Series*	Time Series	2, p. 23
Checking the station status	Status	4, p. 28

1.3 Parameter Info

No.	Element	Push-Function
1	**vali::tool status**	→ *Parameter Details (see 1.3.5)*
2	**Alarm status**	
3	**Value Type**	
4	**Sensor name**	

1.3.1 vali::tool Status

The colour of the checkmark indicates the status of the *vali::tool* component:

- *vali::tool* is active and monitors the parameter permanently
  Respective information is available within the system and written into output files
- *vali::tool* is inactive, no respective information is available for this parameter

1.3.2 Alarm Status

If *ana::tool* is installed and used for generating alarms from a parameter's results, this is indicated by the *alarm symbo*l. The actual alarm status is indicated by the color of the symbol.

- *Grey*: The current parameter reading is within the limits configured for alarm / warning
- *Yellow and blinking*: The current parameter reading is outside the limits configured for alarm/warning
  ➔ ALARM / WARNING event has been detected by the system.

1.3.3 Value Type

The symbol in this field indicates which type of values is displayed as the parameter result.

- C Clean value display is active (measured value from the sensor is displayed in small letters additionally).
- C Clean value display is inactive, the value measured by the sensor is displayed.
- V The V indicates that this is a virtual parameter. This parameter is not measured by a physical sensor but generated on base of a mathematical calculation and hence not observed by vali::tool. Accordingly, no clean value is available for display.

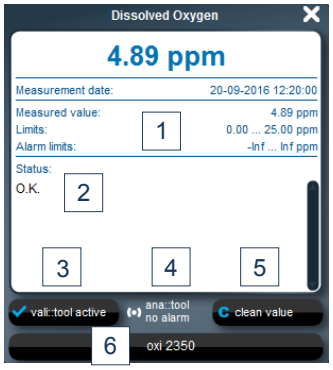
⚠ When *vali::tool* is active, both non-corrected ("*measured*") and validated ("*clean*") results are stored in the database, irrespective of the value type selected for display.

1.3.4 Sensor Name

The name of a sensor is configured during its installation for easy identification. But the name can also be changed later on.

1.3.5 Parameter Details

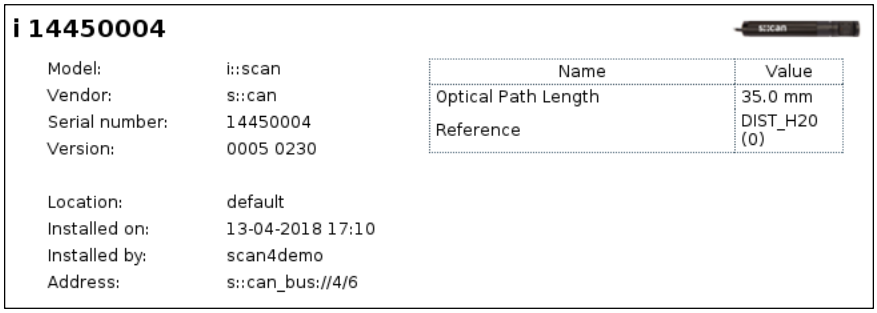
Pushing anywhere on *Parameter Info* will pop up additional detailed information about the parameter and also provide access to configuration functions.



No.	Element	Information	Push-Function
1	**Limits**	Measuring and alarm limits	-
2	**Status**	Parameter status information	-
3	**vali::tool**	Status of vali::tool	Activate / deactivate
4	**Alarm**	Alarm status	-
5	**Clean value**	Value type	Toggle between measured / cleaned value
6	**Sensor**	Sensor name	→ *Sensor Details* *(see 1.3.6)*

1.3.6 Sensor Details

Pushing the sensor name in *Parameter Details* will pop up additional detailed information about the parameter.



⇨ *Relevant **Manual sections**:*

Parameter settings	Service / Terminal / Parameters	6.4.5, p. 45
Parameter validation with *vali::tool*	Service / Sensors / Parameters	6.5.4, p. 61
Defining static alarms	Service / Terminal / Parameters	6.4.5, p. 46
Parameter monitoring with *ana::tool*	Service / ana::tool	6.7, p. 68
Managing alarms	Alarms	5, p.31
Graphical results analysis	Time Series	2, p. 23
Configure virtual parameters	Service / Terminal / Parameters	6.4.5, p. 46
Sensor settings	Service / Sensors / Sensor Settings	6.5.1, p. 60

⇨ *Available **Quick reference guides**:*

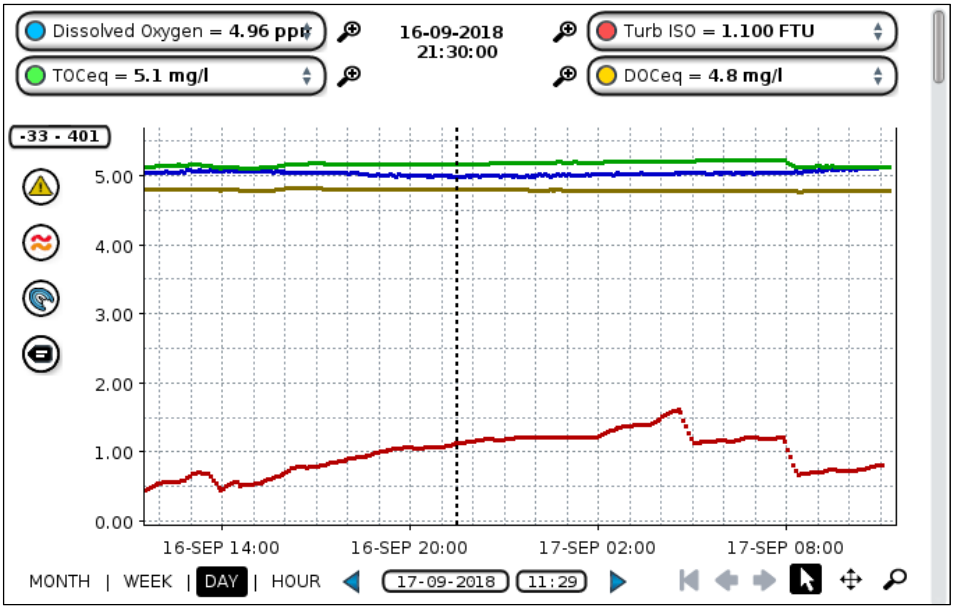
Manuals	*moni::tool*	note_moni-tool_overview

Please refer to the s::can Customer Portal!

2 Time Series

2.1 Overview

The standard view of the *Time Series* screen, is displayed when the respective tab is selected. It shows historical parameter results graphically.
As a default view the last day (24 hours) of results for the first four parameters are presented.



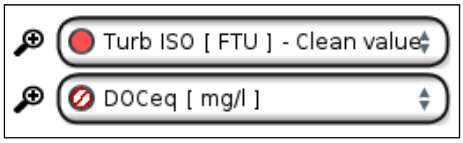
Using the various screen elements, the display can be adapted to the needs

- Parameter selection (2.2);
- Axes scaling (2.3**Fehler! Verweisquelle konnte nicht gefunden werden.**);
- Data selection (2.4);
- Data scaling (2.5).

2.2 Selecting Parameters

By default four parameters are combined in a single graph, based on the order used on the *Values* screen. If more than four parameters have been configured several graphs are produced automatically. This is indicated by the active scroll bar which can also be used to navigate between the different graphs.

However, the composition of parameters displayed together in a single graph can also be chosen freely.

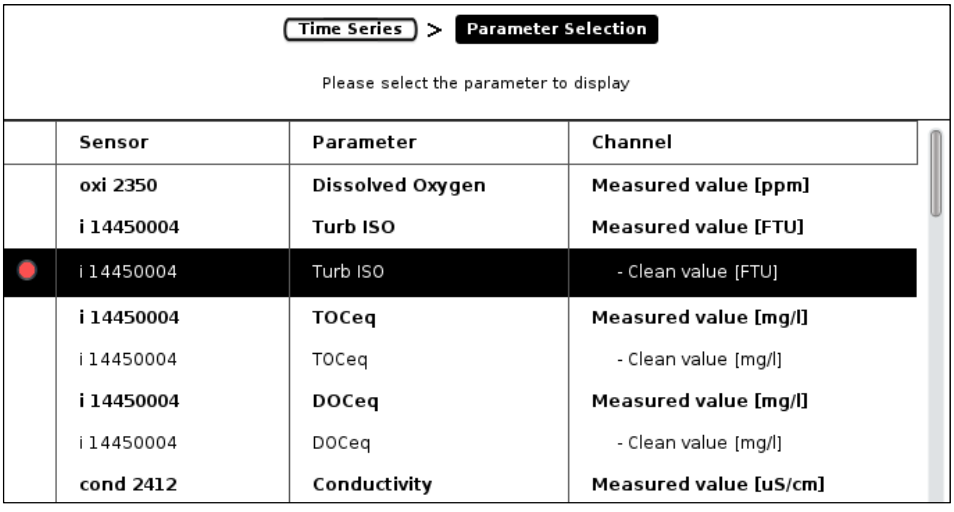


Firstly, a parameter can be temporarily hidden by pushing the respective parameter's coloured marker in the legend (the marker will then change into a red / white symbol). A hidden parameter can be displayed again by pushing its symbol once more.

Secondly, parameters displayed in a graph can be replaced by pushing their name in the legend on top of the graph.

This will open a selection list, from which the desired parameter can be selected. The parameter to be replaced is marked in the list as it is in the legend.

The selection list includes not only all currently active parameters, but also historical parameters, i.e. parameters that have meanwhile been removed. Respective results exist only for the phases when the parameters actually existed and accordingly can be displayed only for those periods.



2.3 Axes Scaling

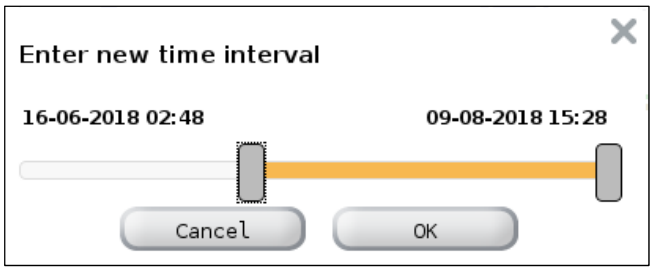
For exploring parameter data effectively it is useful to scale the **x-axis (displayed period)** and the **y-axis (parameter values)** of the graphs appropriately. In this respect it can also be helpful to configure the parameters displayed together accordingly (see 2.2) as the average values of different parameters can differ significantly, hence making it difficult to produce meaningful graphs.

Selecting the displayed period

The default view in *Time Series* shows the results collected over the last 24 hours ("*Day*"). Other periods are activated by pushing the respective buttons.

The time stamp of the last displayed value is shown in the respective fields which can also be used for entering the desired values directly. Pushing the blue triangles shifts the displayed data-window for half of the currently selected period.

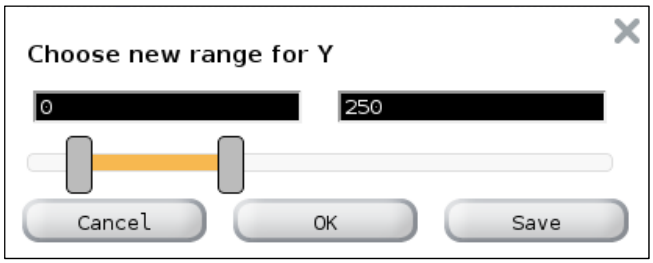
Another possibility to change the displayed period is to push directly the x-axis which makes a window as shown on the right pop up. Here both start and end date of the displayed period can be selected directly by pulling the respective markers (see also Data Navigation below).



Selecting the displayed values

Automatic „y-axis zoom" can be done by activating the magnifying glasses shown next to each parameter name which will scale the y-axis so that the minimum and maximum values of the respective parameters can be displayed.

Besides, pushing directly the y-axis produces a similar window as described above which can be used accordingly.



The selected range can be stored in the button for quick y-axis setting on the according display by pushing *Save*.

Zoom

The activation of the magnifying class allows a free and simultaneous selection of both the displayed period and the displayed values.

Undo – Redo

All axis-related changes can be managed with the following graphical elements.

The *Reset Button* will undo all zoom and scroll (see below) functions performed in the current graph. *History Back* will undo the last action while *History Next* can be used to redo an action that was just undone.

In all *Time Series* views with an x-axis comprising more than six hours aggregated data are displayed. This means that the number of data points shown has been reduced and the larger the displayed time window the fewer data points are shown per interval. As a result of this data aggregation the full dynamics of the data might not always be visible! However, when zooming in far enough, each single measurement will become visible again.

2.4 Data Navigation

Besides the scaling of axes *Time Series* allows also to navigate through historic data.

The date and time fields always indicate the newest displayed data. They can also be used to enter values directly while the blue arrow buttons *Older* and *Newer* can be used to shift the data window a half of the selected display period.

By activating the *Move* button the data window can be moved around freely with the cursor while maintaining the selected span of the both axes.

2.5 Data Exploration

Once the axes are scaled and the appropriate period has been selected, the displayed data can be explored further (see Figure in section 2.1 also).

When the *Data Selection* button is activated a yellow marker line is displayed within the data window which can then be snapped to every timestamp displayed. Simultaneously, the respective time stamps of the marked data as well as the corresponding parameter values are displayed in yellow letters. Details for each parameter can then be displayed by pushing the parameters' names. The marker is deactivated by pushing the arrow button again.

Other possibilities for data exploration are provided through the list of buttons on the left which are explained below.

The first button allows selecting one parameter for which then the measured and the cleaned values are displayed graphically, together with the status over time. This is also the view which can be accessed directly from the *Values* tab by pushing a parameter's result and from the *Alarm* tab. Other parameters can be displayed using the scroll bar.

The second button allows selecting two parameters which are then displayed together on two y-axes which makes it easier to detect mutual dependencies and joint influences. Again, the scollbar is working to compare the first selected parameter with all the others.

The fingerprint button changes to the *Fingerprint* view for the currently selected timestamp (using the *Arrow* button). This fingerprint can then be compared directly to another one. For further details please refer to the *Fingerprint* chapter (3).

The *Tag* button can be used to assign an individual name to the currently selected timestamp. Tagged measurements can then be selected for displaying them while tag names can be exported into files. Please refer also to 6.4.4 / Measurement Tags.

The Back button, which is visible when the fingerprint of a selected measuring point is displayed, will switch back to the previous time series view.

When data exploration changed to another view, the *Time series* button brings the user back to the standard time series view.

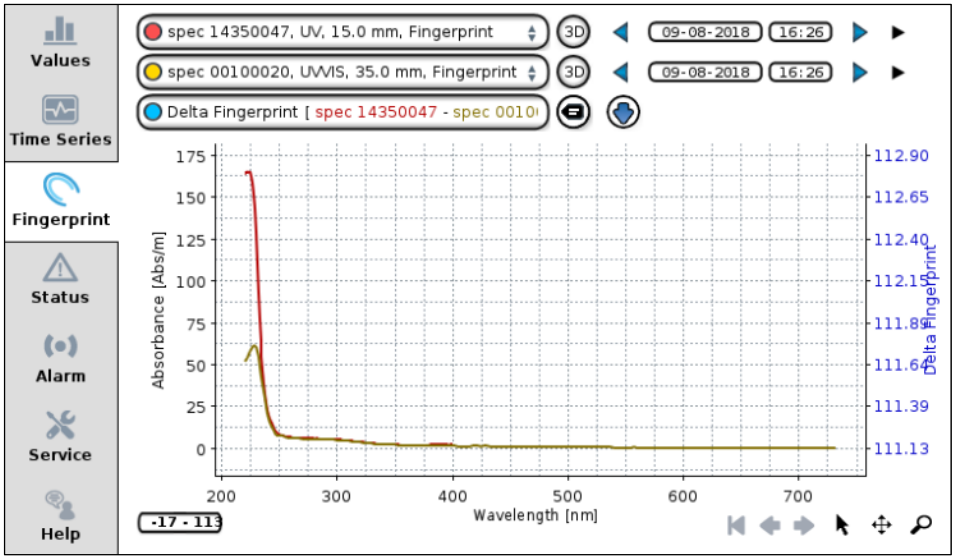
⇨ *Relevant **Manual sections**:*

Parameter settings	Service / Terminal / Parameters	6.4.5, p. 45
Parameter validation with *vali::tool*	Service / Sensors / Parameters	6.5.4, p. 61
Parameter monitoring with *ana::tool*	Service / ana::tool	6.7, p. 68
Measurement tags	Service / Terminal / Settings / Measurement Tags	6.4.4, p. 45

3 Fingerprint

3.1 Standard view

When one or more spectro::lysers are installed, the *Fingerprint* screen as a standard shows the most recent absorption spectrum („fingerprint“) measured by these sensors.

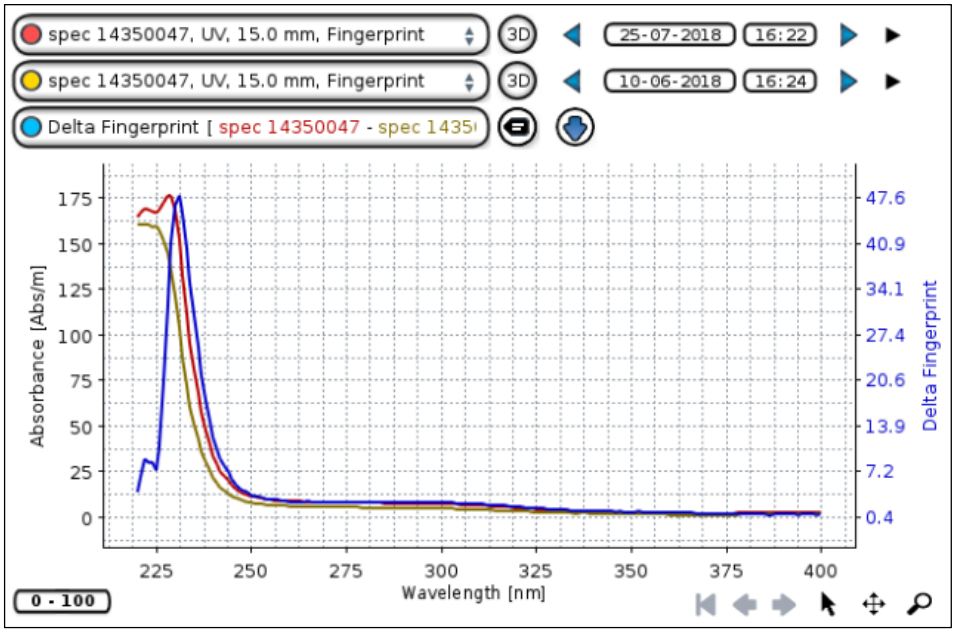


On the horizontal axis (x-axis) the wavelengths in nm are presented while on the vertical axis (y-axis) the measured absorbance values in Abs/m are displayed.

Two fingerprints can be displayed simultaneously where the second fingerprint can be either from another time stamp or from another spectro::lyser.

Selecting the sensor(s) to be displayed is done by pushing the parameter names in the legend on top of the graph.

When a second fingerprint from the same sensor is selected, the difference between the two fingerprints (*Delta Fingerprint*) is shown as a blue line scaled individually on its own (right) y-axis. With this method developments in the media composition over time can be easily detected.



For selecting the time stamps of the displayed fingerprints the following options exist:

- Entering date and time directly in the respective legend fields;
- Using the blue navigation arrows which will move to the previous or next fingerprint.

3.2 View Configuration

Configuring the view of *Fingerprint* according to the individual preferences and needs can be done similar to how it has been described under *Time Series* by making use of the same buttons and functions (axes scaling, zooming, moving undoing, data selecting).

It is also possible to temporarily remove fingerprints from the screen. Like for *Time Series* this is done by pushing the colored marker of the fingerprint in the legend. This will change the marker into a red and white symbol and the respective fingerprint will be hidden. It can be displayed again by pushing the marker symbol once more.

3.3 Special Features

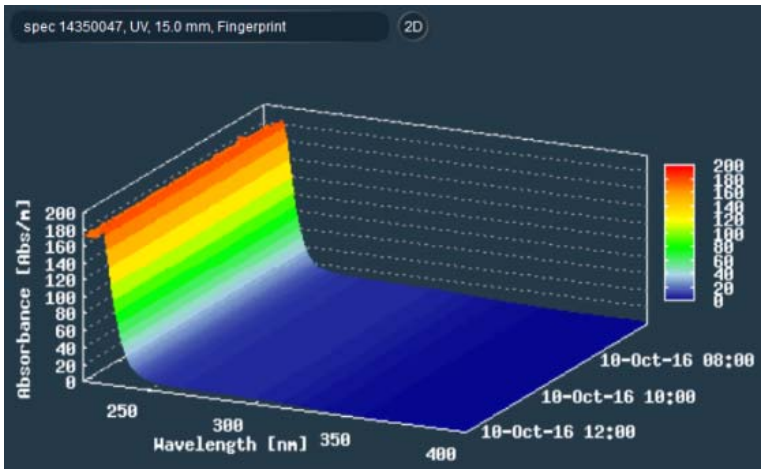
While basically working with fingerprints is very much like manipulating time series, *moni::tool* also comprises some features which exist only for analyzing data from spectro::lysers.

Animation

When the selected time stamp is older than the newest available data the *Play* buttons beside the time stamp fields allows starting an animation. One fingerprint after the other is loaded and displayed automatically, hence providing a lively impression of how quality developed over time.

During the animation the *Play* button changes to *Pause* and can be used to stop the animation.

3D Graph



Besides the standard two-dimensional view it is also possible to show fingerprints from one spectro::lyser in a three- dimensional graph which provides a good impression of quality developments.

This option is activated by pushing one of the *3D* buttons in the legend next to the name of the sensors. Each button will link to a 3D graph for the respective sensor, displaying the fingerprints of six hours before the entered time stamp.

For re-activating the normal 2D-display either the *2D* button within the 3D-view can be used or the *Fingerprint* tab can be reloaded.

Wavelength time series

When the *Data Selection* tool is activated the *3D* buttons as described above change into the *Wavelengths* button. A click on it changes to a special view within the *Time Series* tab showing data of four wavelengths of the respective spectro::lyser. As a standard initially the first four wavelengths are displayed, subsequent wavelengths can be accessed using the *Scroll Bar*.

The *Return* button can be used to get back to the previous *Fingerprint* view.

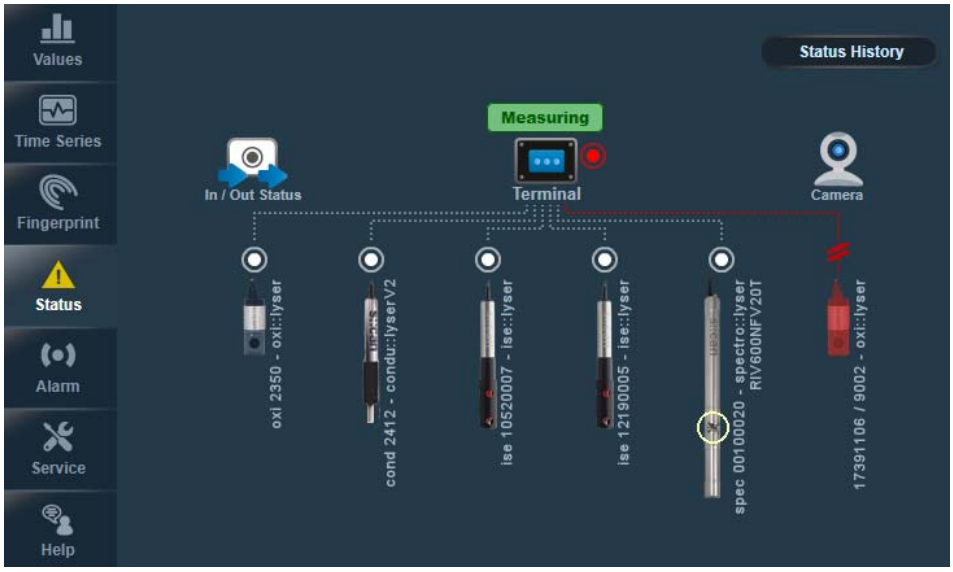
Data download

The detailed data from spectro::lyser can be used in a variety of ways which go beyond the capacities of the standard terminals. Therefore the *Download* button beside the *Delta Fingerprint* field offers the opportunity select one of the installed sensors and to download the fingerprint data for the indicated time stamp. The respective data is saved as a csv-type file which can be easily processed with standard calculation software.

4 Status

4.1 Overview

The *Status* screen provides an overview of the active monitoring system, including the terminal, all sensors, in- and output modules as well as other devices (e.g. camera) that are currently configured in *moni::tool*.



Within this station overview various coloured circles indicate the components' status:

- *White* - OK
- *Yellow* – spectrometer measuring
- *Red* - error or fault

A *red connection* between the terminal and a sensor together with a *red sensor symbol* indicates that communication with the sensor is not possible.

Above the terminal symbol the remaining time until the next measurement or Measuring is displayed.

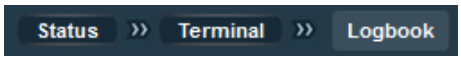
Whenever any problem has been detected the *Status* tab is animated with a yellow sign.

For each system component further status information can be retrieved by pushing the respective symbol. In case of problems the messages provide the following information:

- **Origin** of the error (e.g. the related parameter of the sensor)
- **Timestamp** indicating when the error occurred.
- General **description** of the error.
- A **code** for communicating the error to the s::can sales partner (see Part D, chapter 2.2)

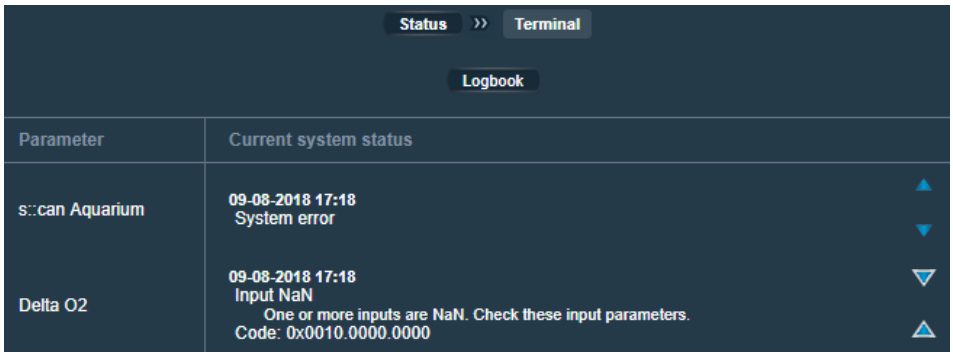
Pushing the general error description provides the detailed information and suggestions on how to resolve the problem.

The *Status History* button leads to a graphical overview of status and *vali::tool* messages.



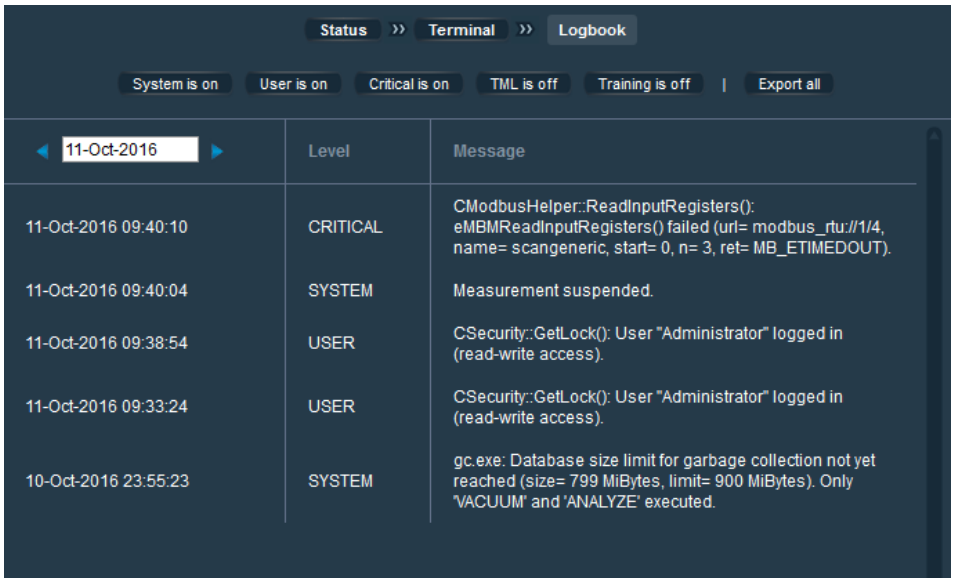
On top of the details screens a "navigation line" indicates the hierarchical level of the currently active screen. Its elements can also be used for navigating.

4.2 Terminal



Pushing the *Terminal* icon opens the respective *Status* view, displaying all relevant messages.

Additional details on the messages, e.g. error codes and clear text messages can be displayed by pushing the blue triangles.

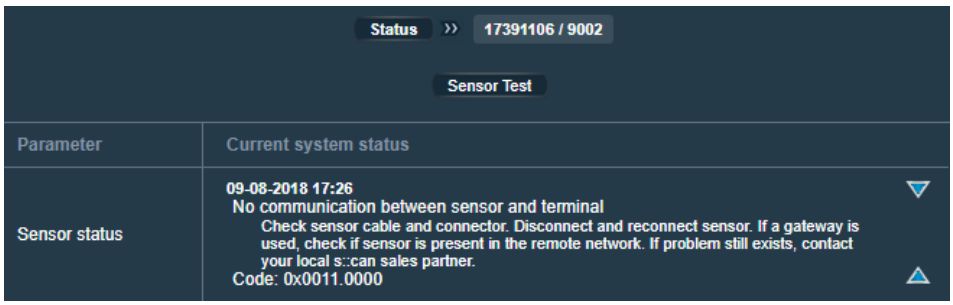


Pushing the *Logbook* button in the terminal status window opens a list with all logbook messages. The types of messages (*Level*) to be displayed can be configured using the buttons above the list.

Also the date of the messages can be selected in the respective field.

Pushing the *Export all* button leads directly to a file dialog pop-up for exporting the available logbook entries to a compressed zip-File.

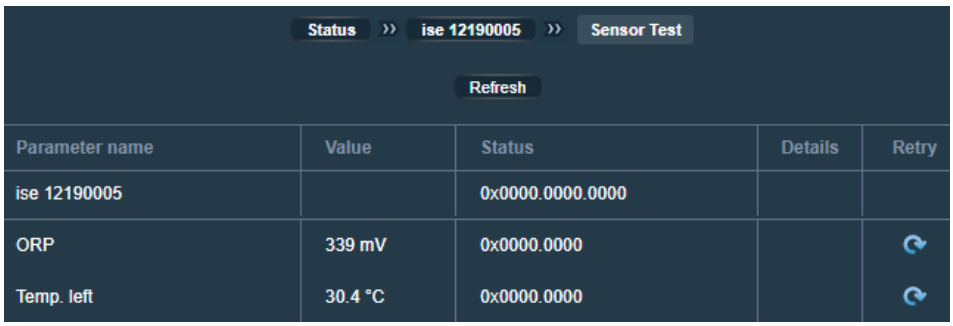
4.3 Sensors



Pushing the icon of any sensor opens the respective *Status* view, displaying all relevant messages for both the hardware and the measured parameters.

Additional details on the messages, e.g. error codes and clear text messages can be displayed by pushing the blue triangles.

Besides the standard status messages as described above, the sensor status screen provides also a *Sensor Test* button.



Pushing this button collects and displays the current readings and status information from the sensor.

The information can be refreshed either by pushing *Refresh*" or for each parameter individually by pushing the *Retry* icons on the right side.

4.4 Other components

4.4.1 Input / Output Modules

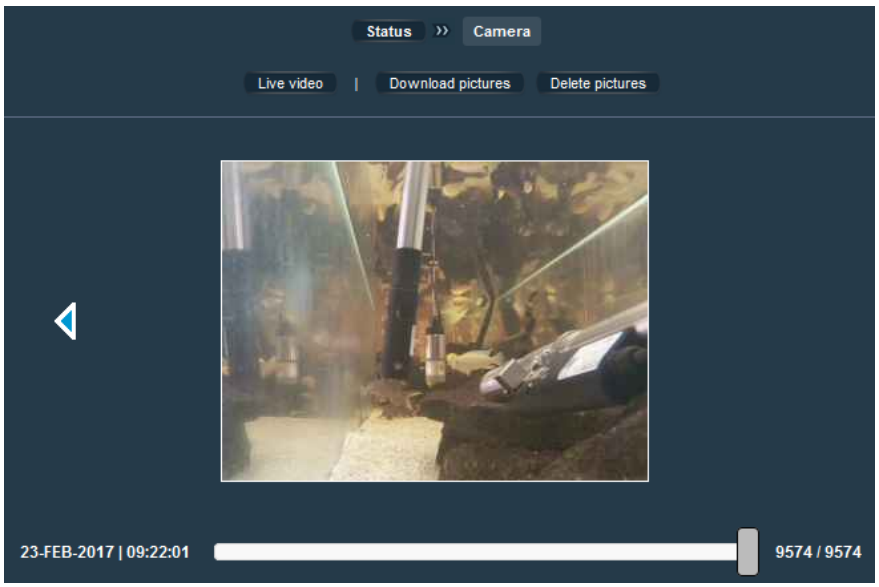
The In / Out Status window showing all configured interfaces as well as their current status is displayed which can be refreshed manually can be opened by pushing the respective icon.

The Address is displayed in user level *Expert* only.

4.4.2 Camera

If a camera is part of the monitoring system the respective status window shows the newest picture made and provides the options to launch a live video or to download the saved pictures.

Please be aware that depending on the speed of the network connection displaying images might take some time.

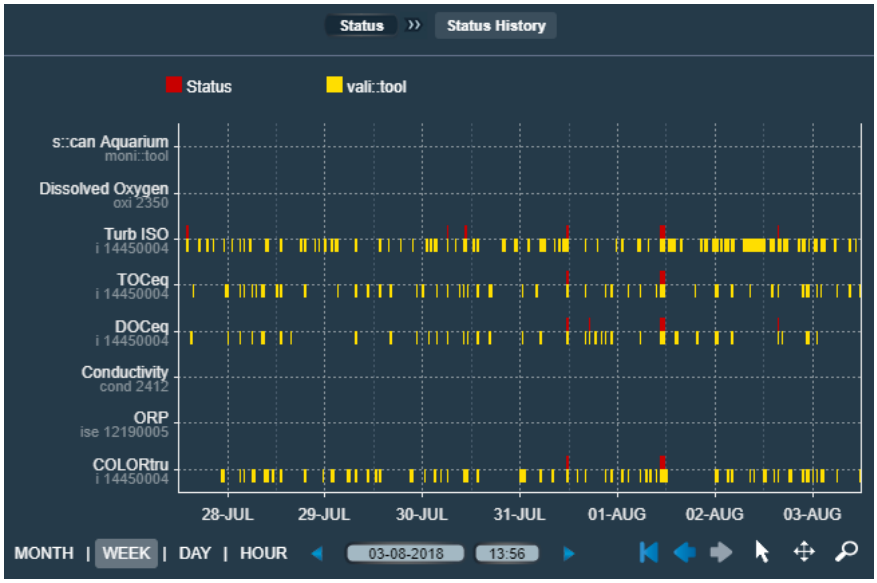


4.5 Status History

Pushing the *Status History* button on the status overview screen's upper right corner opens a graph displaying the history of both the system status and, if it is activated, the *vali::tool* status.

The graph shows for each parameter when a status error message has been recorded (red marks) and when *vali::tool* reported suspicious data quality (yellow marks).

The default view shows the status history for the previous day but the shown period can be adjusted using the x-axis scaling tools (see also section 2.3).



If not all configured parameters can be displayed in one window the scroll bar will be activated.

5 Alarm

5.1 Background

Basically warnings and alarms can be triggered in three different ways:

- ***Static alarms*** based on checking single parameter readings between an upper and a lower limit;
- ***Spectral alarms*** based on data from spectro::lysers;
- ***Pattern alarms*** based on the combined signals of several sensors.

Both, warnings and alarms are always based on the same criteria but a warning is issued already at a certain level before the actual alarm limit has been reached.

For further details on alarms and the respective configuration steps please refer also to chapters 6.4.5 and 6.5.4 dealing with the *Parameters* sections of individual sensors and the monitoring system, respectively.

Static alarms

For each parameter the upper and lower thresholds can be set individually. Whenever the cleaned value is outside the defined boundaries an alarm will be triggered for the parameter.

Spectral alarms

This alarm type is based on virtual parameters building on a combination of wavelength measurements from spectrometer sensors. For further details on the configuration of spectral alarm parameters please refer to chapter 6.4.5. Information on the capacities of the ana::tool software in monitoring virtual parameters is presented in chapter 6.7.

Pattern alarms

This alarm type is based on virtual parameters building on a combination of several parameters from one or several sensors. For further details on the configuration of pattern alarm parameters please refer to chapter 6.4.5. Information on the capacities of the ana::tool software in monitoring virtual parameters is presented in chapter 6.7.

5.2 Overview

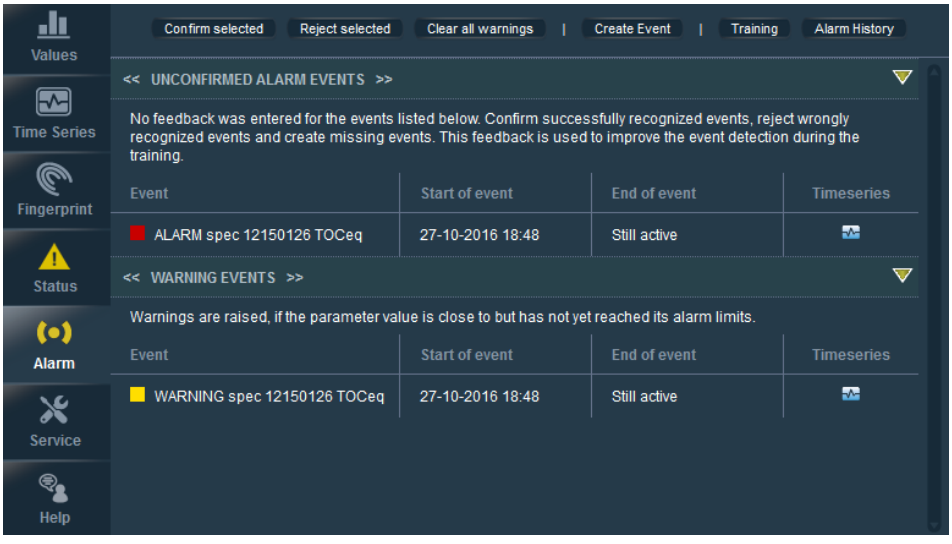
The *Alarm* main window provides an overview of all currently pending warnings and alarms.

Each entry provides the following information:

- Source of event
- Start and end of event

The headline of each column can be used for sorting the recorded events.

Using the *Timeseries* icon of an entry, the event can be explored in the respective view of the *Time Series* tab (see also 2.5).



Whenever the system identified an event, user feed-back must be provided, firstly, for acknowledging the respective information and secondly, because a correct working of ana::tool depends on secured data quality based on user feedback.

Warnings have to be cleared.

⚠ As long as alarms or warnings are pending, the overall system status will be set to *Alarm*, which is also indicated by a yellow blinking tab icon

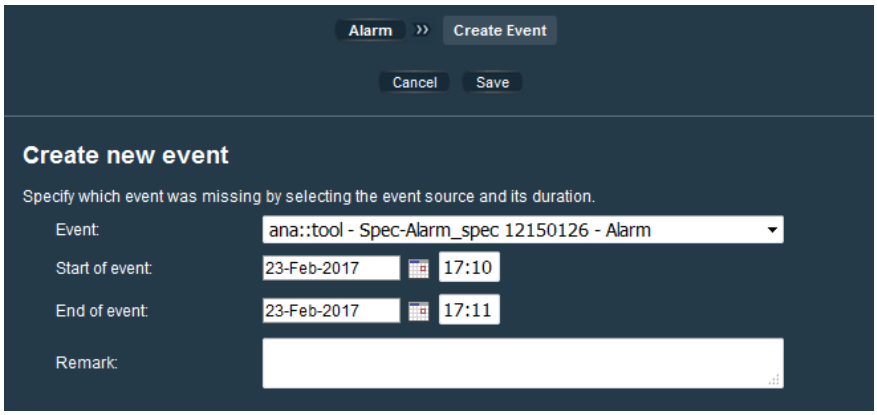
If the detected event is not considered to be a real alarm situation it has to be rejected by marking the respective entry and pushing *Reject selected* afterwards. In all other cases – also when unsure if the system has identified a real event - the alarm should be confirmed by marking the respective alarm line and pushing *Confirm selected* afterwards.

In both cases a new page will appear afterwards in which further details about the event can be entered.

Neither confirmed alarms nor warnings will have an effect on the next training. Only rejected alarms and manually created alarms (in case of missed events, see below) will change the *ana::tool* training dataset and will hence have an impact on the results of the next training.

⚠ Once an alarm has been triggered no other events will change eventually assigned digital outputs or notifications. Therefore pending alarms should be either confirmed or rejected as soon as possible so the system becomes ready to react on new events.

5.3 Create Events



Relevant events which were not detected by *ana::tool* should be made known to the system by using the *Create* function.

Consequently, a window will pop up to enter the following information:

- Parameter which triggered the alarm;
- Start and end time of the alarm;
- Additional remarks.

Pushing *Save* will store the customer created alarm which is automatically considered as *confirmed*.

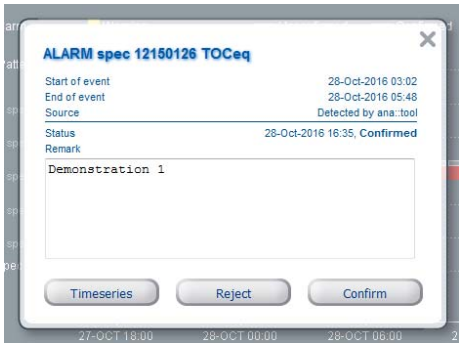
5.4 Alarm History

The entry *Alarm History* provides a graphical overview of the alarm status of all parameters over time.

Yellow indicates that a parameter was in the state of *warning* and *red* indicates a state of *alarm*.

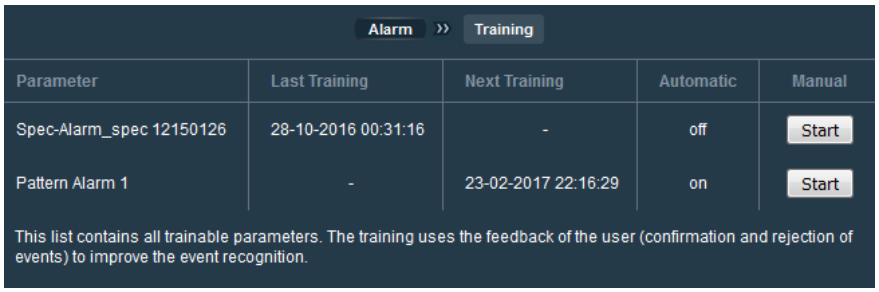
Furthermore, the actual status of the currently saved events is indicated (*Unconfirmed*, *Confirmed*, *Rejected*).

With the activated "*Select*" tool (white arrow) individual events can be marked. This will open a dialogue window providing detailed information about the event. Furthermore, a remark on the event can be added and it is also possible to modify the event's status. Events created by the user can also be deleted.



5.5 Training

As mentioned already earlier, the system's capabilities for detecting spectral and pattern alarms based on the *ana::tool* component can (and should!) be trained through learning from user feed-back (see above).



Pushing the button *Training* opens a window providing an overview of all trainable parameters, their last training and the planned next training (if *Automatic* is on).

Besides, also a *manual training* can be initiated through the *Start* button.

If the user has no rights to change settings, the window is in read mode only.

⚠ For detailed information on training and other aspects of alarm parameters please refer to the chapter on *ana::tool* under the *Service* tab (6.7)

⇨ *Relevant **Manual sections**:*

Configure parameters	Service / Terminal / Parameters	6.4.5
Configure sensor parameters	Service / Sensors / Parameters	6.5.4
Parameter monitoring with *ana::tool*	Service / ana::tool	6.7

6 Service

6.1 Overview

The *Service* tab provides access to all functions related to configuration, service and maintenance of all the monitoring station's components.

Service can be accessed only when a user is logged in.

The actually available options depend on the specific rights granted to the respective users, with insufficient rights read-only access is possible to some areas.

From the tab's main screen all elements of the monitoring system can be accessed via their respective icons. If more than 6 sensors are installed, additional icons for active or new sensors can be found after scrolling down to the next page.

Besides, the two buttons *Enter Service Mode* and *Sample & Calibration* provide access to additional specific tasks.

The table below provides an overview on the components accessible via the various elements as well as the respective chapters.

Enter / Leave Service Mode (6.2)
- Suspend / continue measurements

Sample & Calibration (6.3)
- Take samples
- Sample list

Terminal (6.4)
- Measurement
- *con::cube*
- Network
- Settings
- Parameters
- Date & Time
- Terminal Language
- Display
- Security
- Extras

Sensors (6.5)
- Sensor settings
- Remove sensor
- Replace sensor
- Parameters
- Global calibration
- Calibrate sensor
- Optical path length
- Function check

Add New Sensor (6.6)
- s::can sensors
- Analog sensors
- Sensors via TCP
- 3rd party Modbus sensors

ana::tool (6.7)
- Manage spectral and pattern alarm parameters
- *ana::tool* training

Digital Inputs (6.8)
- List al *con::cube* digital inputs

Outputs (6.9)
- Analog outputs
- Digital outputs
- Fieldbus outputs
- File output
- TML
- Auto-sampler
- SMS Notification

Cleaning Devices (6.10)
- Configure cleaning devices
- Test cleaning devices
- Remove cleaning devices

⇨ *Available **Quick reference guides**:*

Manuals	*moni::tool*	note_moni-tool_overview

Please refer to the s::can Customer Portal!

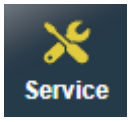
6.2 Service Mode

Some of the configuration options available under *Service* cannot be changed while the monitoring system is operating. Hence, changing into *Service Mode* stops automatic measurements and data storage while all parameter results and outputs are frozen.

Service Mode can be activated manually by pushing the respective button but will also be triggered automatically by starting certain activities within the *Service* tab.

Automatic operation can be resumed by pushing the *Leave Service Mode* button or on the *Measurements stopped!* field (see figure on the right).

Measurements are also resumed after user logout, which happens automatically at the latest after 20 minutes of inactivity (see also chapter 6.4.4 / User Accounts).



As long as the monitoring system is in service mode, the *Service* symbol in the main navigation bar is changed to yellow and blinks.

6.3 Sample & Calibration

The *Sample & Calibration* button of the *Service* main window provides access to taking and managing sample measurements. It is hence a key component for the advanced calibration features available for the different sensors.

For details on how to actually perform sensor calibration please refer also to chapter 6.5.6!

Before dealing with moni::tool's sampling features it is useful to look into some details on calibration. Hence, the next chapter deals with the characteristics and requirements of various calibration types, while taking and managing the respective samples is dealt with in subsequent chapters!

6.3.1 Calibration types

Calibration in *moni::tool* basically means procedures which allow accurately assigning sensor measurements to results of reference analytics.

One respective option are pre-defined relations valid for certain circumstances, so called *Global Calibrations* (see 6.5.5) They do not depend on sample or laboratory measurements.

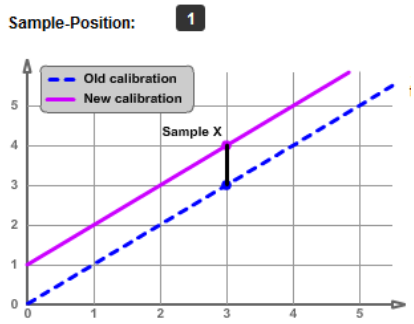
A second approach is to take samples from the medium to be tested and collect both the sensor readings and the corresponding laboratory measurements for the required parameters from the respective sample. This approach is facilitated by *moni::tool*'s *Sample & Calibration* function.

Depending on the number and the quality of available sample data various calibration approaches can be used for sensor calibration (please refer also to chapter 6.5.6). What they have in common is that they all make use of a linear relationship between sensor readings and reference analytic results, i.e. that a calibration is always characterized by a straight line defined by its *offset* and its *slope*.

The differences between the various calibration methods refer to the parameters which are actually changed (offset and/or slope) and to the way the new values are identified, as is shown in the following paragraphs.

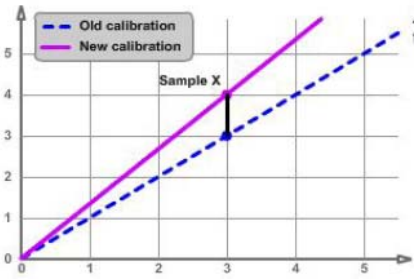
Offset

- *Offset* will be changed while *slope* remains unchanged (for *ammo::lyser* and *oxi::lyser* parameters please refer also to the respective sensor manuals for additional details)
- One sample measurement and one laboratory result are needed
- Sample and laboratory results are stored on the sensor
- Special case Zero: One sample measurement in zero medium is needed



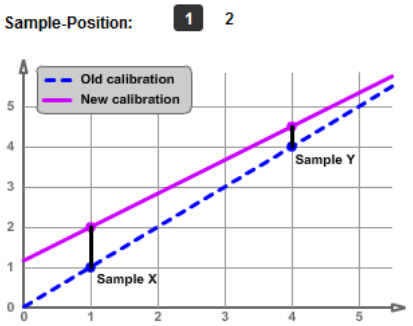
Span

- *Slope* will be changed.
- One sample measurement and one laboratory result in a relatively high concentration range are needed
- Sample and laboratory results are stored on the sensor



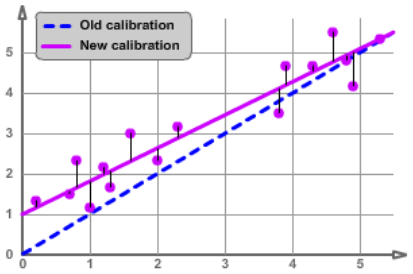
Linear

- *Offset* and *slope* will be changed (for *ammo::lyser* and *oxi::lyser* parameters please refer also to the respective sensor manuals for additional details).
- Two sample measurements (position 1 and 2) and two laboratory results are needed
- Sample and laboratory results are stored on the sensor



Multi

- The performed calibration is a linear fit with smallest possible deviation
- *Offset* and *slope* will be changed
- Store up to 1000 samples and laboratory result on the terminal to see correlation and identify outliers
- Samples are stored with a sample ID and a sample timestamp.



Sample measurements can be done directly in the medium, in grab samples or in standard solutions. *Laboratory* measurements can be results of laboratory analyses, values of standard solutions or readings of a validated reference device. Please also refer to the specific sensor-manuals for respective recommendations.

6.3.2 Taking samples

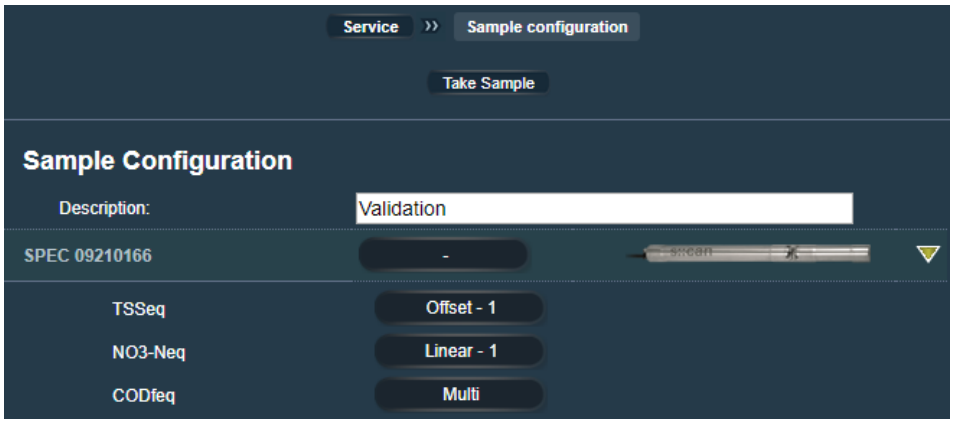
Once *Sample & Calibration* has been selected, activating the *Take sample* function changes the system into *Service Mode*.

Firstly, the *Sample Configuration* screen will be displayed and in the top field a sample description can be entered which supports selecting the correct samples when it comes to sensor calibration. However, samples are identified in the sample list (see chapter 6.3.3) by their automatically assigned ID and by their time stamp.

Pushing the button to the left of the sensor picture opens the *Select calibration type* window which allows selecting a calibration type (as described in chapter 6.3.1) for the sensor in general. Still, pushing the yellow triangle on the right of the sensor picture displays all parameters configured on this sensor and calibration types can be defined individually for each parameter.

Calibration types:

- *None*:
  sample will not be used
- *Offset* (Index 1):
  sample will be stored (*moni::tool* database and on the sensor) as sample 1
- *Linear (Index 1/2)*:
  will be stored (database & sensor) as sample 1 or 2
- *Multi*: will only be stored in the *moni::tool* database and not on the sensor.



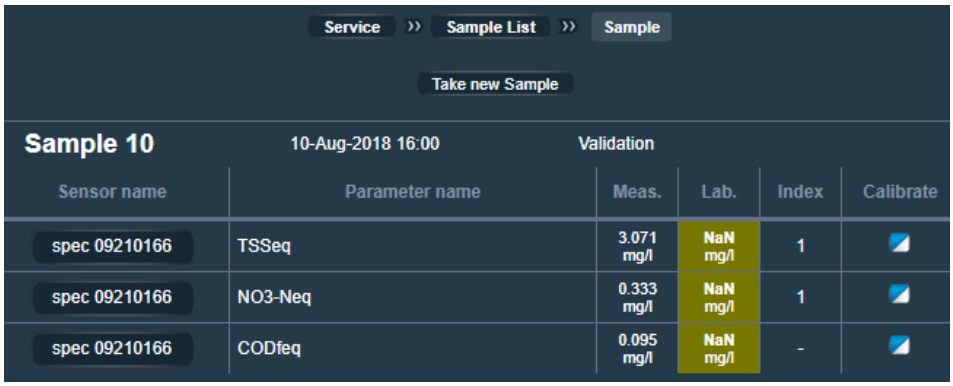
As the calibration type Linear uses two samples it is necessary to define whether sample 1 or sample 2 shall be replaced by the new sample. This is done by selecting either index 1 or index 2 in the *Select calibration type* window.

Once all settings have been done, the *Take Sample* button starts the measurement. The measuring progress and the results will then be displayed on a new screen. During the measurement a sample can be taken from the medium or reference measurements can be made.

As soon as the sampling process is finished all raw values are displayed, together with the number assigned for use in linear calibration (*Index*).

This window can also be accessed at a later time via *Sample list* (see below).

The *Lab* column can be activated for entering the results of laboratory analyses. In the respective dialogue all information regarding the current sample are displayed. The entered result of the laboratory analysis can be stored (*Save*) or deleted (*Clear*).



⚠ When several samples of a parameter are assigned to the same sample index (1 or 2) of a calibration type, only the one which has been saved latest will be actually available for calibration as only one sample per index can be stored on the sensors. Alternatively, several valid samples can be used for calibration simultaneously by selecting the calibration type *Multi* as it builds on the *moni::tool* samples database.

Pushing a parameter's *Calibrate* icon will open the respective calibration window. Please refer to chapter 6.5.6 on sensor calibration!

With *Take new Sample* the sampling procedure can be re-started.

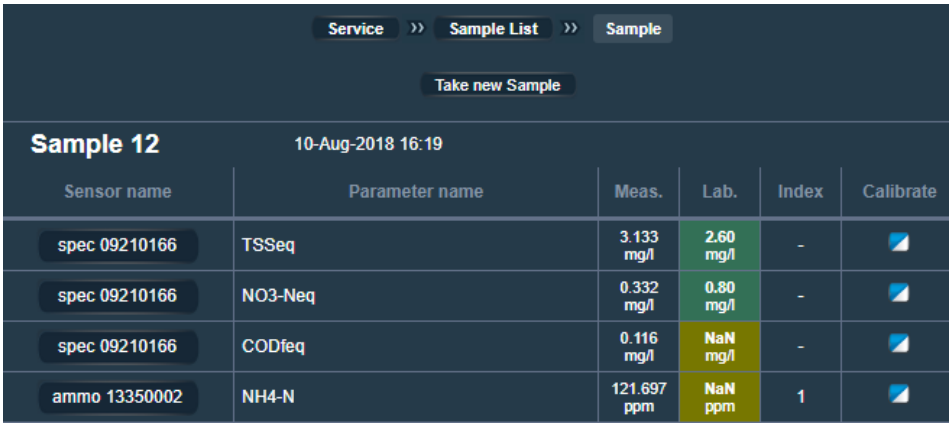
6.3.3 Sample list

When selecting *Sample list* as a first step (see beginning of this section) an overview of all samples stored in the database so far and of the covered sensors will be displayed.

Pushing a sensor symbol opens a detailed view on all sample results collected for that sensor.

Pushing a sample's ID opens the *sample details* window while a results field opens the *data entry* window (both as described already in 6.3.2).

Pushing the trash icon on the right hand side will delete the sample and all related data.

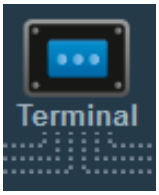


⇨ *Available **Quick reference guides**:*

Manuals	*moni::tool*	note_moni-toolV2.2_sensorcalibration

Please refer to the s::can Customer Portal!

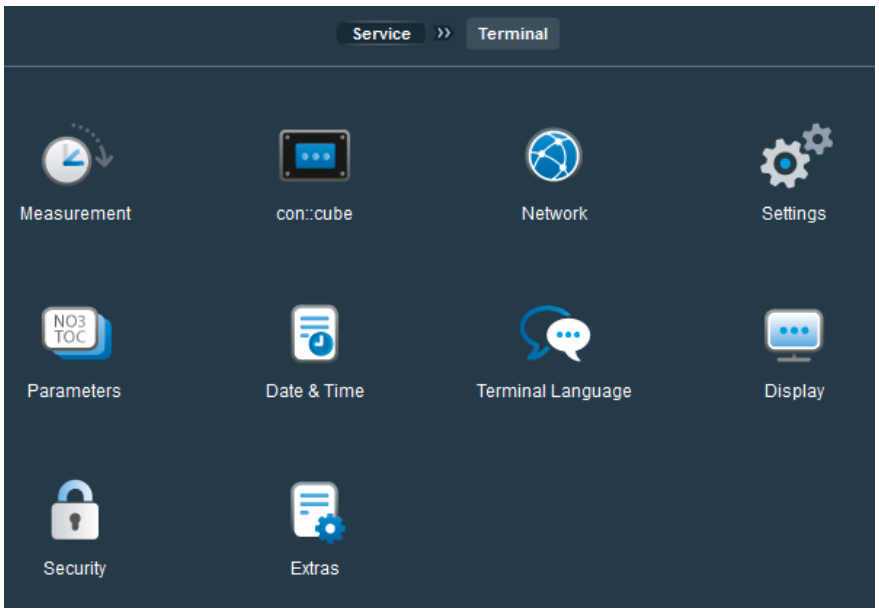
6.4 Terminal



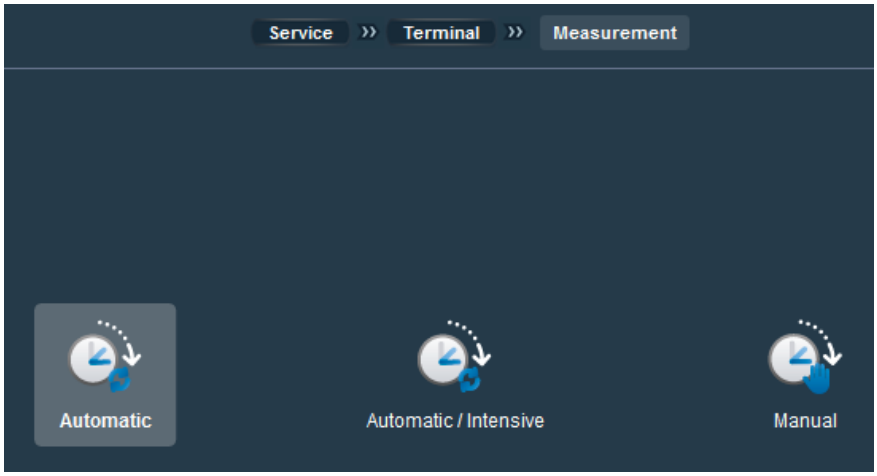
Terminal aggregates all settings and options not linked directly to any of the devices symbolized by individual icons on the *Service* tab main window.

The image on the right shows the view displayed after pushing the *Terminal* icon.

Further details on the individual components are provided in the following sections.



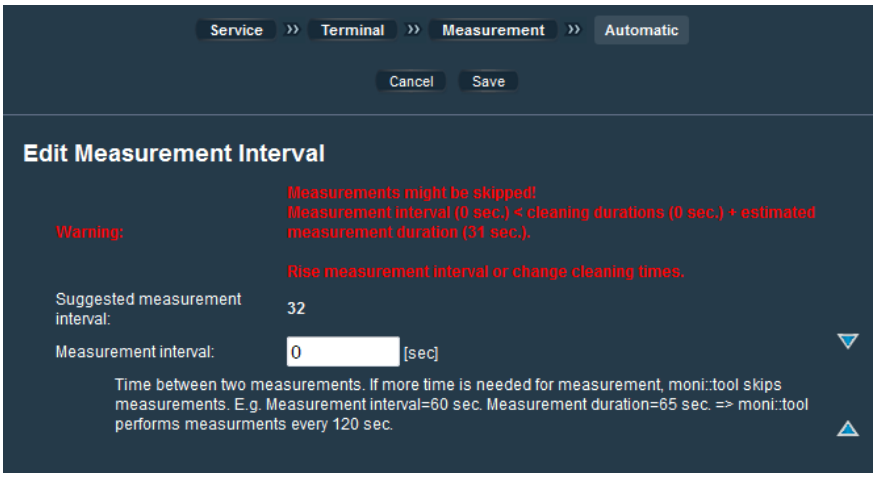
6.4.1 Measurement



Measurement offers three options for controlling the execution of measurements which can be selected by pushing the respective icons.

Automatic

This is the standard operation mode in which *moni::tool* triggers new measurements at regular intervals. In this window the time between automatic measurements (*Measurement interval*), which is set to 120 sec. by default, can be changed (changes will be stored when pushing *Save*).



The user will be warned when the total time of one measuring plus cleaning cycle should be longer than the configured *Measurement interval*.

Whenever the *Measurement interval* is changed, all *vali::tool* and *ana::tool* functions will be re-initialized which may make them unavailable until a new run-in period has been completed.

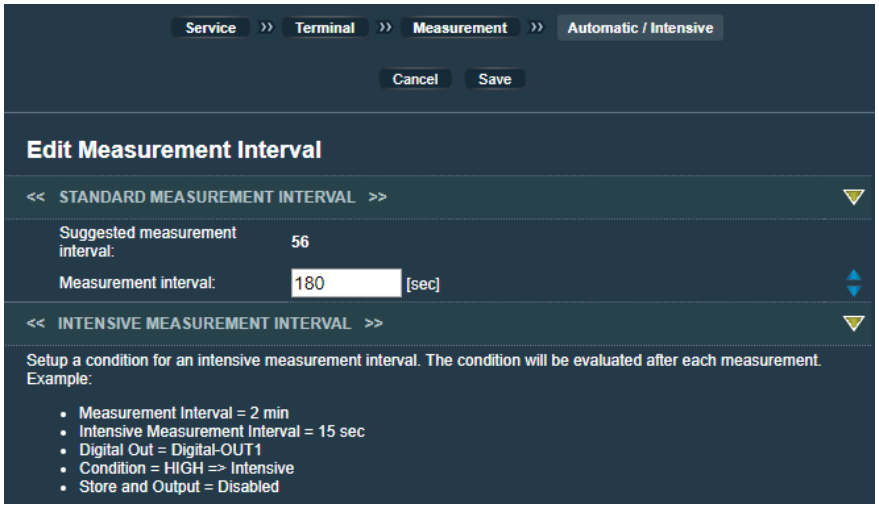
Automatic / Intensive

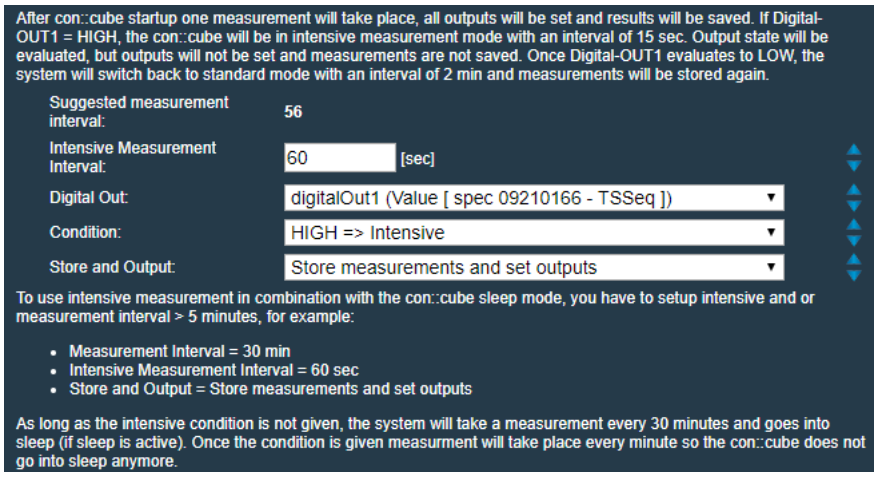
In this measurement mode besides a *Standard Measurement Interval* a condition can be defined which, as long as it is fulfilled, changes to an *Intensive Measurement Interval* with reduced intervals.

The sections for entering the required settings can be displayed or hidden by pushing the yellow triangles.

Basically changing from the standard to the intensive measurement interval is triggered by a configured digital output linked to a respective event, e.g. to a parameter alarm (see 6.4.5, p. 51) or to an external trigger for reacting on situations not detected by the con::cube sensors (see 6.8).

Further details on this measurement mode are provided directly within *moni::tool* and can be displayed by pushing the blue triangles.





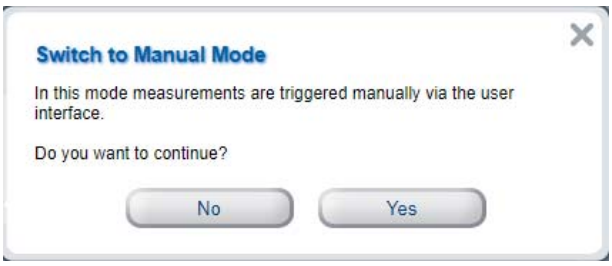
Available ***Quick reference guides****:*

Manuals	*moni::tool*	note_moni-tool_IntensiveMode_ExtTrigger

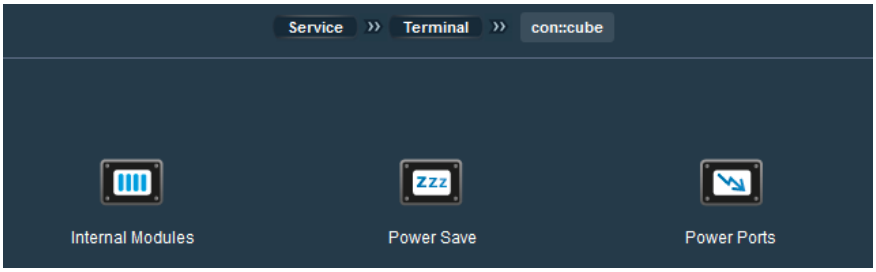
Please refer to the s::can Customer Portal!

Manual

When the system is brought into the *Manual* mode all measurements will be stopped and new measurements can be triggered only by pushing the *Start Measurement* button which then replaces the *activity* field in the bottom menu. Whenever this button is pushed a new window will pop up in which the user has to decide if only a measurement (*Only Measure*) or also a cleaning procedure (*Clean + Measure*) shall be executed.



6.4.2 con::cube



con::cube provides access to hardware-related settings and information.

Internal Modules

This function is only available on the *con::cube* and shows all installed internal modules (digital input, analog input, analog output and digital output).

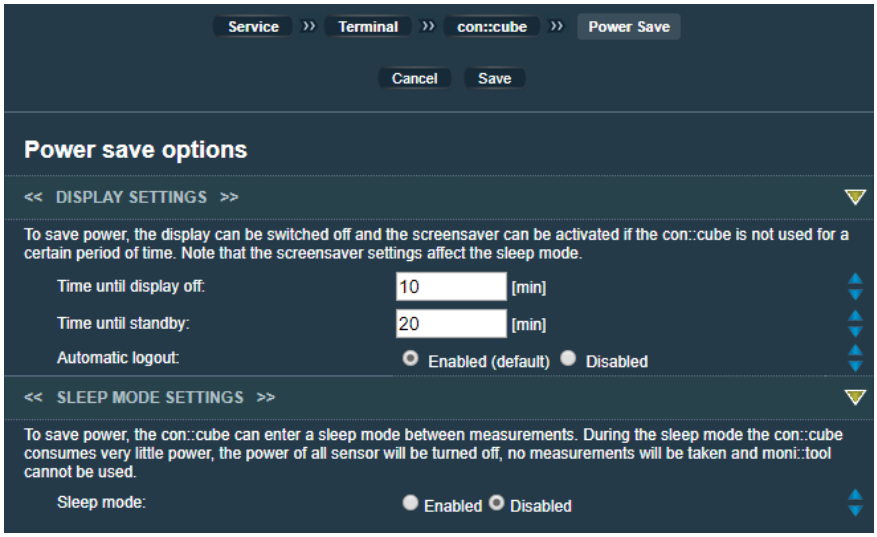
→*Service Level* / *Expert*

- Also the module's addresses are displayed

Modules ordered together with the system are already configured during the assembly of the terminal. For installing additional modules later on, they need to be configured which is triggered by pushing the button *Search Internal Modules*. *(with newer moni::tool versions all modules are recognized automatically during re-start)*.

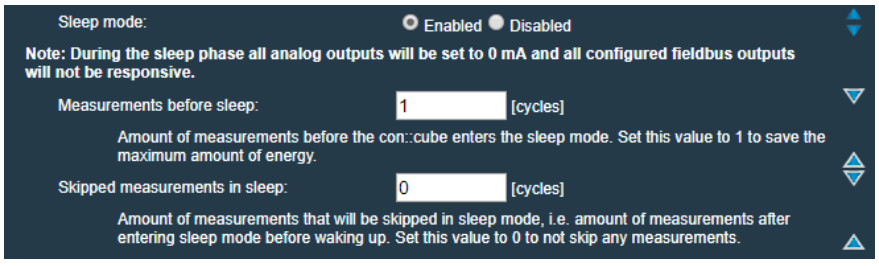
Power Save

In the respective window delays for switching off the display and for bringing the system into stand-by mode can be defined.



For enhanced energy saving during automatic operation (e.g. when operated by a battery) the *con::cube* can also enter into a *sleep mode* between measurements. This switches off the power for most system components and hence no measurements will be taken and *moni::tool* cannot be used.

Sleep mode is enabled by selecting the respective option. Consequently, the number of measurements to be taken before a sleep phase is started and the number of measurements to be skipped (according to the automatic measurement settings, see 6.4.1) have to be configured.



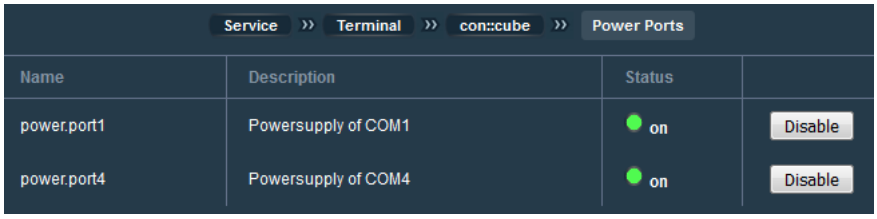
⚠ The *sleep mode* is available only for *con::cube*s already equipped with the required hardware. If a model does not yet support this function a user information will be displayed.

→*Service Level* / *Expert*

- Automatic user log-out after a specified time of inactivity can be enabled / disabled. More details can be found within the online-help.

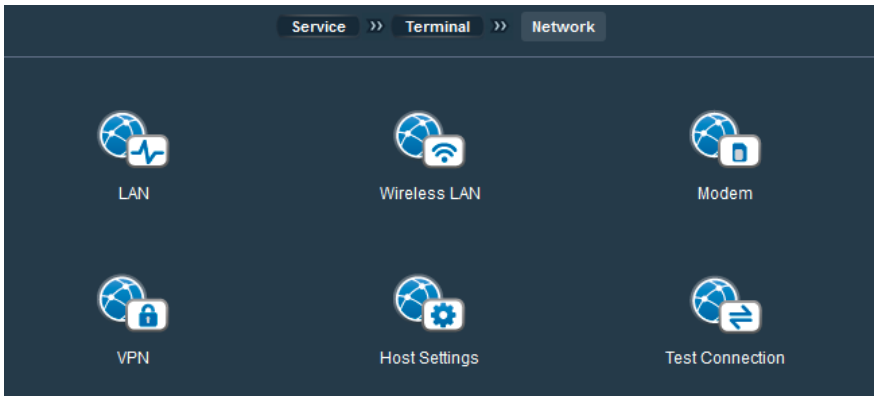
Power Ports

The available con::cube power ports for supplying sensors, can be enabled or disabled within this window.



⚠ This function shall be used by s::can Service only (check correct jumper configuration on board).

6.4.3 Network



Under this components necessary for operating the system in a network environment are addressed.

Usually these points will be configured during the initial set-up (e.g. assigning a computer name under *Host Settings*).

The default setting for accessing the local network under *LAN* is *DHCP activated*. This means that the *con::cube* will accept an IP address assigned by a DHCP server automatically as soon as a network connection is established.

Please refer to the *con::cube* manual and the available *Quick reference guides* for changes of the various network settings.

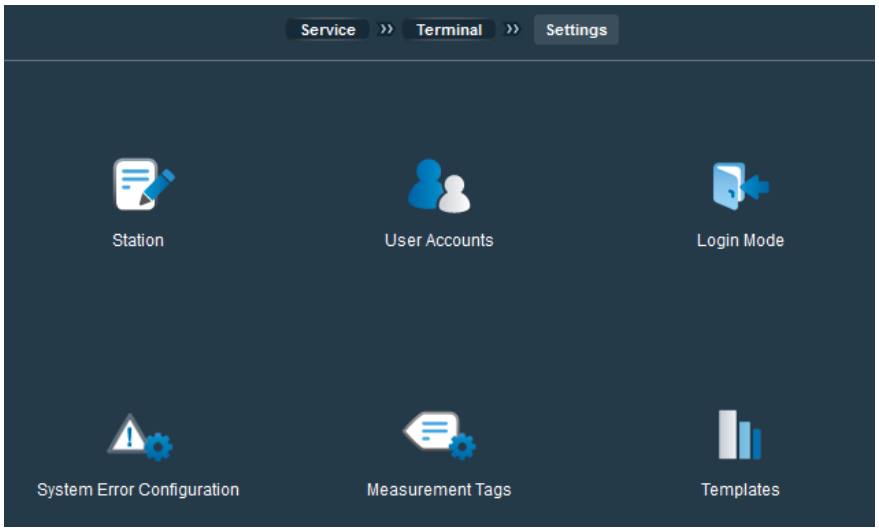
⚠ *moni::tool* can also be controlled by SMS remote control using specific messages.

⇨ *Available **Quick reference guides**:*

Manuals	*moni::tool*	note_moni-tool_3G-Modem-Setup
Manuals	*con::cube*	note_concube_vpn1.s-can.at_access
Manuals	*con::cube*	note_concube_SMS_remoteControl_moniV3

Please refer to the s::can Customer Portal!

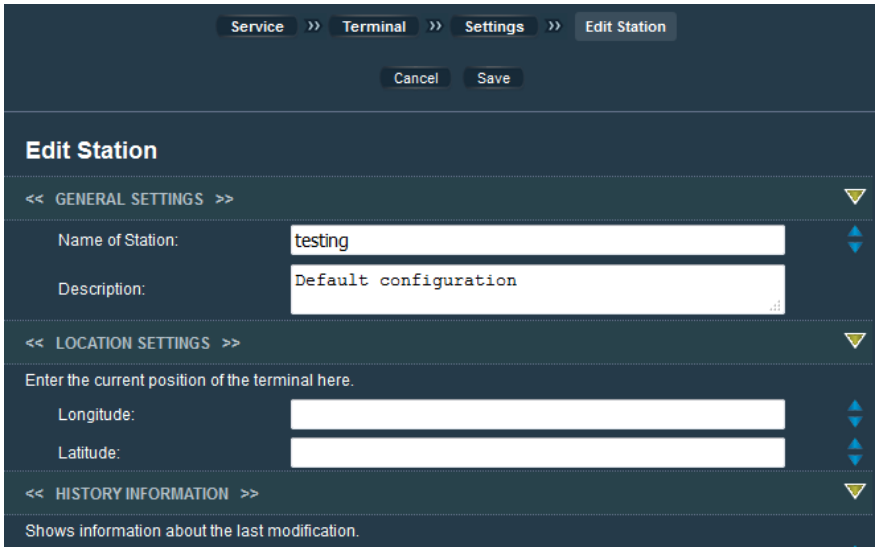
6.4.4 Settings



Station

The name of the station can be set which should happen already during the initial setup (see also p. 13).

The station's name will be displayed permanently in the bottom menu and will also be used for generating file names (see chapter 6.9.4).

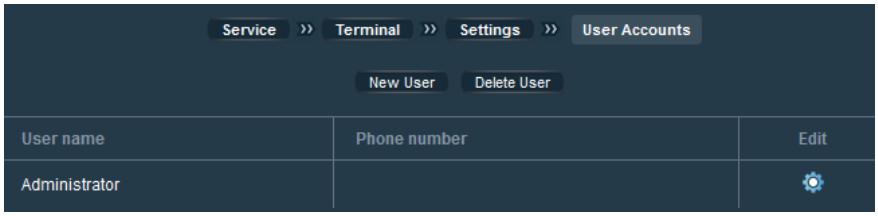


→ *Service Level* / *Expert*

- Also a description of the station and its geographic position can be configured

User Accounts

Based on an initial *Administrator* account, new users can be defined and their specific rights during working with the monitoring system can be configured individually.



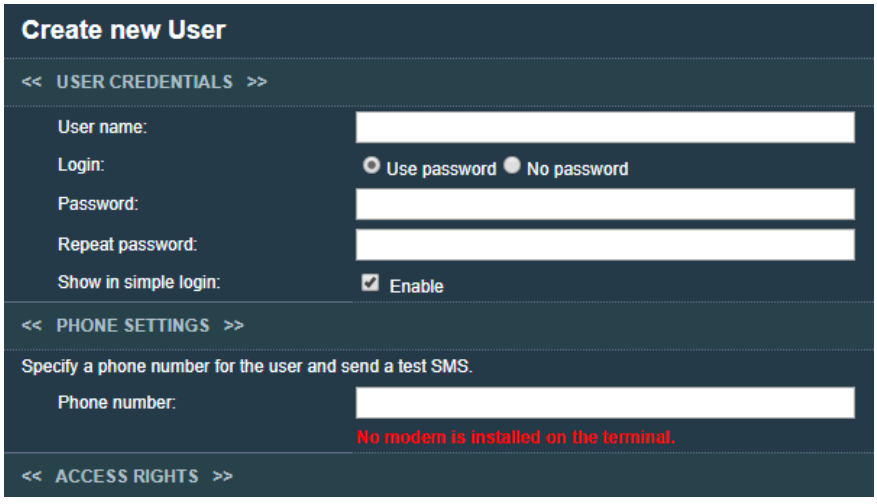
By default the Administrator account is protected by the standard password *admin1*. For security reasons this password should be changed.

In the first window an overview of all current users is displayed.

A new user can be added after pushing *New User* and after defining *User name* and *Password.*

If the *con::cube* is equipped with an internal modem a phone number can be assigned so that the user can receive messages via SMS.

Furthermore, the *Access Rights* for the new user need to be assigned. This is a critical step as it defines which options are actually available when this user is logged into the system and should thus be handled rather restrictively.



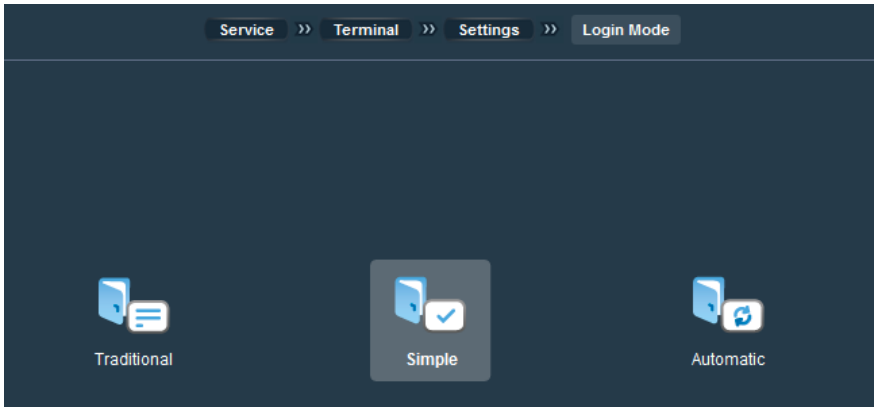
Details on each right can be displayed by pushing the blue arrows on the right side.

→ *Service Level* / *Expert*

- Rights for *Software Update* and *Terminal Control* can be granted
- Service levels *Advanced* and *Expert* can be assigned

A user can be deleted by any user with access rights to *User Management* by pushing the user's name and then pushing the button *Delete User*. To change the user password or to check and re-configure the assigned access rights user profiles can be edited by pushing the blue cogwheel on the right-hand side of a user name.

Login Mode

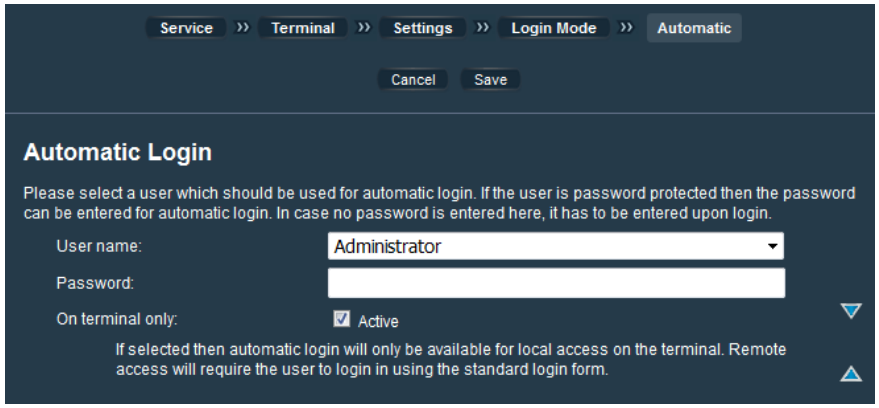


For logging into the system three modes are available:

Traditional:
A user name and the password must be entered during log-in.

Simple:
Configured usernames are available via Icons, the respective password must be entered manually.

Automatic:
A configured user will be logged-in either automatically or after entering the password.

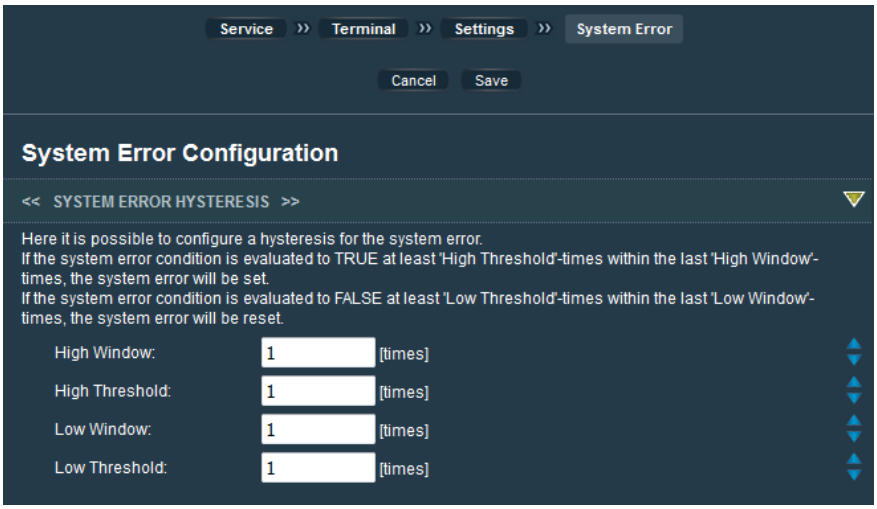


→ *Service Level* / *Expert*

- Automatic log-in can also be activated for remote sessions

System Error Configuration

A “System Error” is indicated by the blinking yellow *Status* symbol and by a red *con::cube* LED.

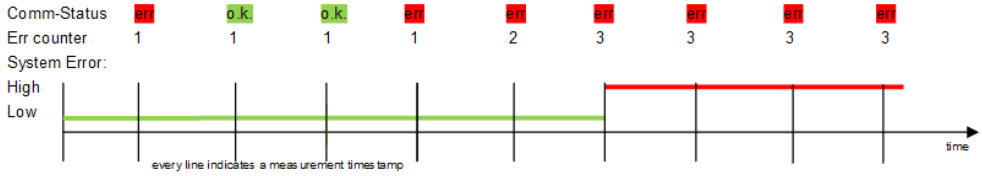


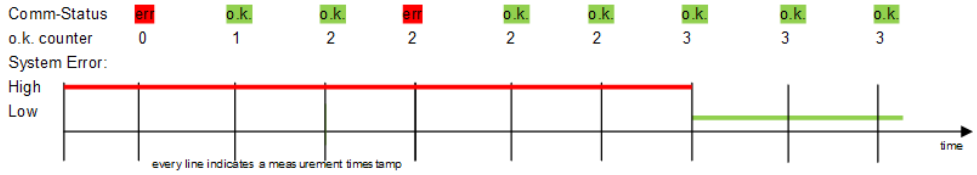
In this menu it can be configured under which conditions a system error is set and when it will be cancelled again.

Per default “System Error” is set whenever any error occurs (e.g. a communication error between a sensor and the terminal). This is also the recommended setting.

However, it is also possible to configure observation windows (= number of observed measurements) and thresholds (= number of respective conditions met) both for setting system error (=”High”) and for cancelling it (=”Low”).

The following graphs show examples for the impact of setting all options to 3.

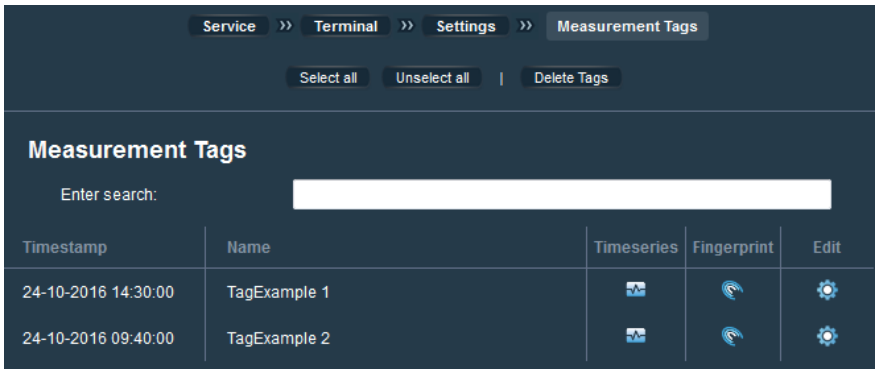




Measurement Tags

As described already in chapter 2.5, individual measurement points can be marked with tags so that they are easy to find later.

This menu displays a list of already assigned tags and allows editing of individual tags as well as shifting directly to the respective time series or fingerprint view by pushing the corresponding icons in the respective tag's line.



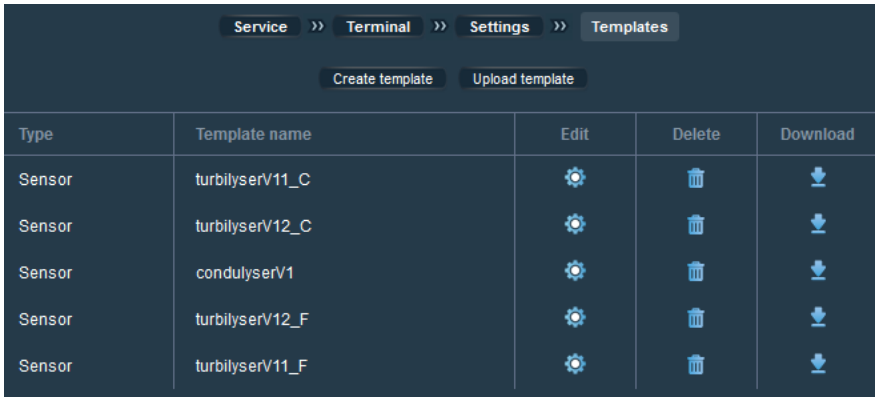
Stored tags can be selected either individually by pushing the respective line or jointly by pushing the respective button (*Select all*) above the list. Selected tags can then be deleted by pushing the respective button (*Delete Tags*).

Templates

Under this point templates for communicating with 3rd party sensors can be managed (see 6.4.4).

The start view shows a list of currently installed templates which can be edited, deleted or downloaded individually by pushing the corresponding icons in the respective template's line.

New templates can either be uploaded as files or be created manually by pushing the respective buttons above the list.

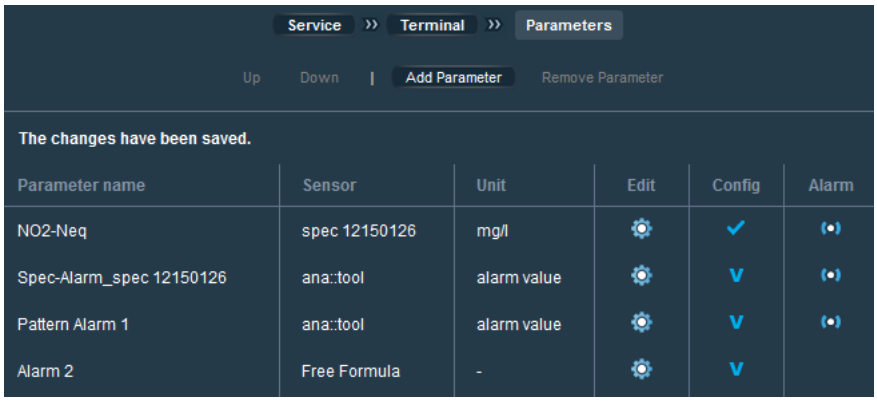


*Available **Quick reference guides**:*

Manuals	*moni::tool*	note_moni-tool_Install3rdParty_Modbus_Sensors

Please refer to the s::can Customer Portal!

6.4.5 Parameters



This section addresses the configuration of parameter measurements of the *moni::tool* system.

It is closely linked to chapter 6.5.4 which deals with managing sensors and their respective parameters as when installing a sensor usually all parameters provided by this instrument will be installed automatically.

However, as this respective section addresses parameter-related issues more comprehensively and addresses virtual parameters (see below) as well, many aspects are described here in full detail.

For virtual parameters the software component in which they are produced is shown in the *Sensor* column instead of a sensor ID and a blue *V* is shown in the *Config* field (as it is in the *Values* tab, see 1.3.3).

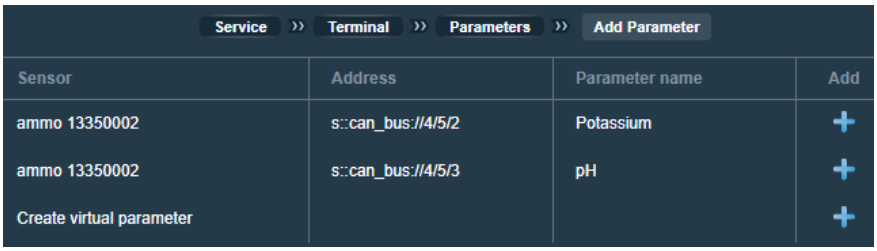
Parameter Order

All active parameters from all installed sensors are listed in the main window and their ranking defines the order in which they are displayed in the Value and Time Series screens. The order can be changed by pushing the parameter name to select the parameter (several names can be selected at the same time, a selection is undone by pushing the parameter a second time) and then pushing *Up* or *Down* to move the parameter(s) in the list.

Deactivate / Reactivate Parameters

Individual parameters can be deactivated using the *Remove Parameter* button both in this window and during defining settings for the respective sensor.

Deactivated parameters are listed on the *Add Parameter* screen and can be reactivated by pushing the blue *plus* sign in the column *Add* on the right hand side.



Add Parameter also offers the option to create virtual parameters as is described below.

Add Virtual Parameters

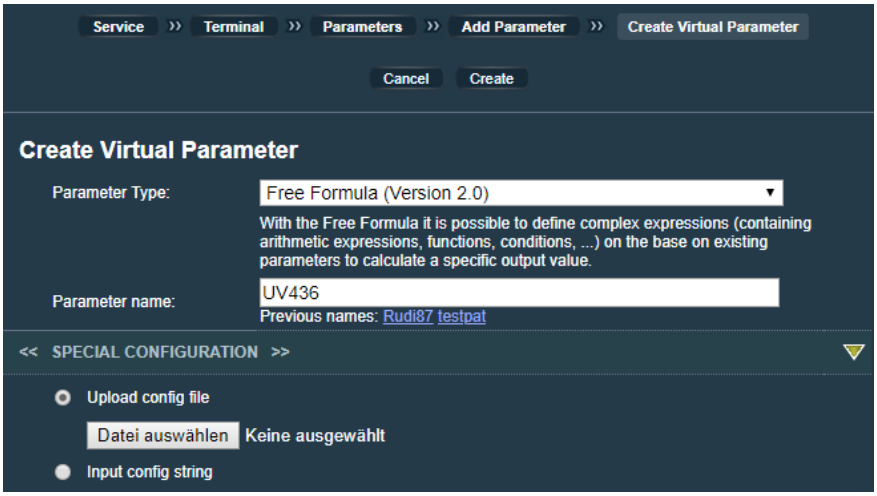
Besides the option to reactivate sensor parameters as described above, there also exists the possibility to create so called *virtual parameters* in *Add Parameter*. These are derived parameters using the readings of other sensors as inputs and performing calculations within the *moni::tool* software. The respective formulas either exist predefined in the software (e.g. Water Quality Index, WQI) or they can be configured individually by the customer.

Several types of virtual parameters exist which are explained in the following sections.

Each of them is created using the *Add Parameter* button and pushing the blue plus symbol in the *Create virtual parameter* line (see image in the right side).

For each parameter the parameter type needs to be selected and a parameter name must be assigned.

The creation process is finalized using the *Create* button and in the following window the information required for the specified parameter type can be entered.

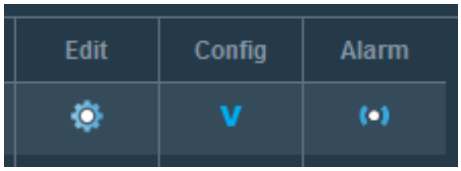


Generally, for all virtual parameter types it needs to be configured which information shall be used as an input and how this information shall be used.

Further details for the individual types are provided in the sections below.

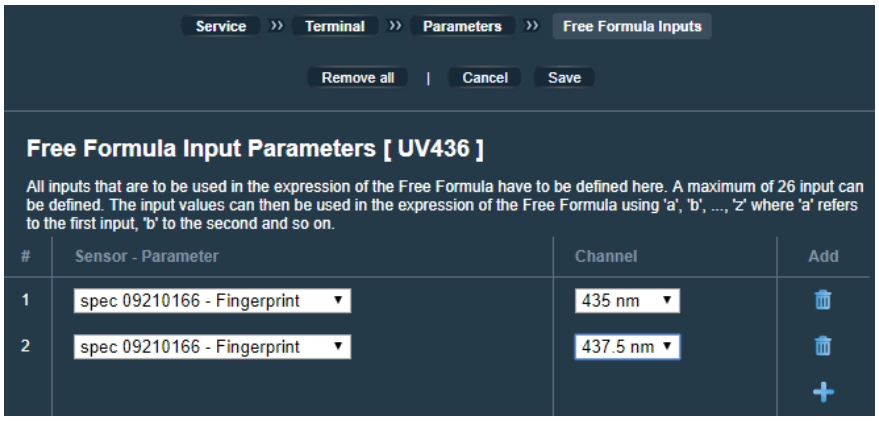
Once a virtual parameter has been created it is marked in the *Config* field with a blue *V*. Pushing the *V* provides access to editing the various setting.

Like for all parameters, the *Cogwheel* symbol can be used for editing basic parameter settings while the *Alarm* icon provides access to various alarm settings (see sections below).



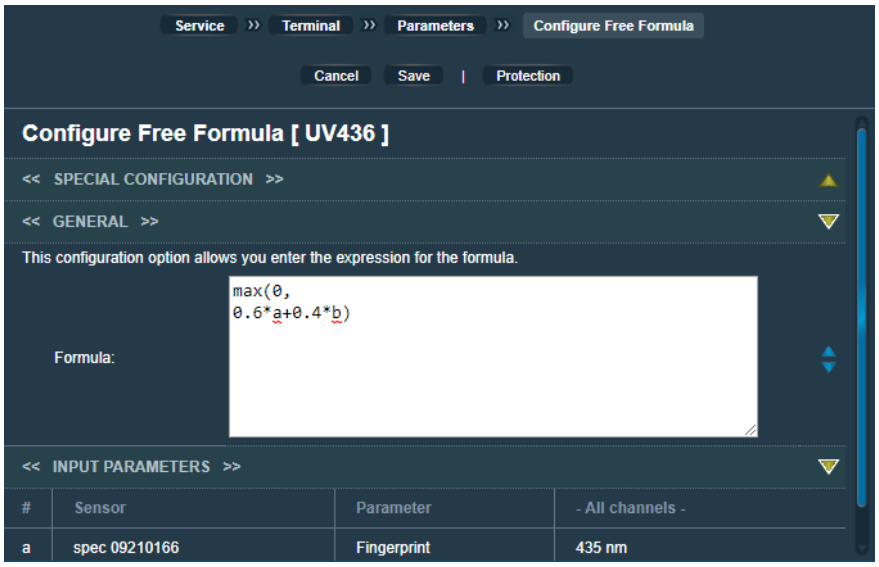
Free Formula

As is indicated by its name, this parameter type allows defining a free formula based on any sensor reading available and on a variety of arithmetic and logical functions.



As a first step, all input parameters that are necessary for calculating the virtual parameter have to be defined. This is done by selecting the *Sensor parameter* in the first column, then selecting the parameter type (measured /clean) (or the wavelength in case of a spectro::lyser) in the column *Channel* and finally pushing the blue *plus* symbol on the right. After all required input parameters have been selected the button *Save* on the top finalizes the step.

In the next window the *Formula* for calculating the parameter can be entered.



The list of all selected input parameters is displayed below the entry field for the formula. Each parameter is assigned a letter as a variable identifier to be used in the formula, starting with "*a*".

The formula itself can be constructed building on the selected parameters and on a series of predefined functions, constants as well as arithmetic and logical expressions (see examples on the right).

As a decimal indicator the sign "." (*dot*) must be used.

Supported Opperators:
! := locical not
<= := less or equal
>= := greater or equal
!= not equal
== := equal
< := less than
> := greater than
+ := arithmetical addition
- := arithmetical subtraction
* := arithmetical multiplication
/ := arithmetical division
^ := raise x to the power of y
Logical FALSE will evaluate to 0, logical TRUE will evaluate to 1.

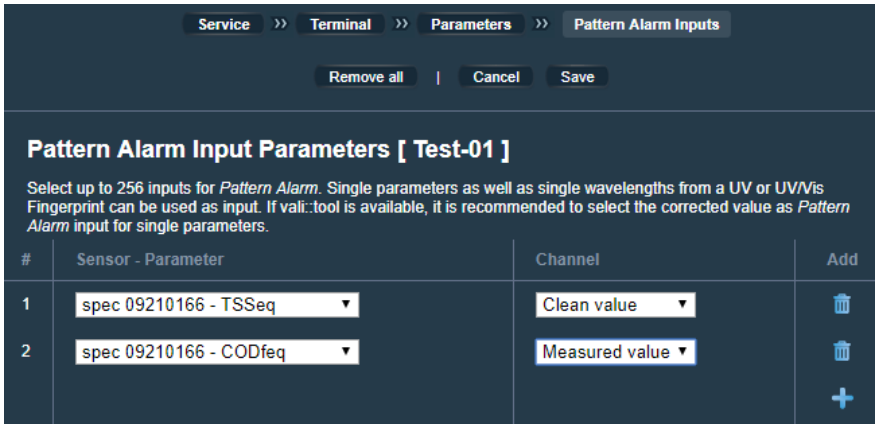
⚠ Pushing the blue and yellow triangles on the right hand side of the screen will pop up detailed additional information on how to build formulas!

After the formula has been entered, the button *Save* needs to be pushed for storing the newly defined parameter.

Pattern Alarm

A Pattern Alarm is an event detection method that builds on typical relationships between a set of parameters. If the actual pattern changes to a state outside the acceptable range an alarm will be triggered. The software component providing the respective capacities is *ana::tool,* please refer also to chapter 6.7 for further details.

The initial configuration steps for a *Pattern Alarm* are the same like those for a *Free Formula*.



However, different from a formula, alarm parameters are not controlled by a static definition but are based on a continuous learning of sensor readings patterns which are considered acceptable. The respective *training* can be done automatically or manually, respective settings are made in the *Configuration* window launched by pushing the blue *V* once the parameter has been created.

The user feed-back required for categorizing situations used for training the parameter is provided in the *Alarm* tab as described in chapter 5. Trainings can be triggered manually directly from the *Alarm* tab (see 5.5) but are managed under the menu item *Service* / *ana::tool* as described in chapter 6.7.

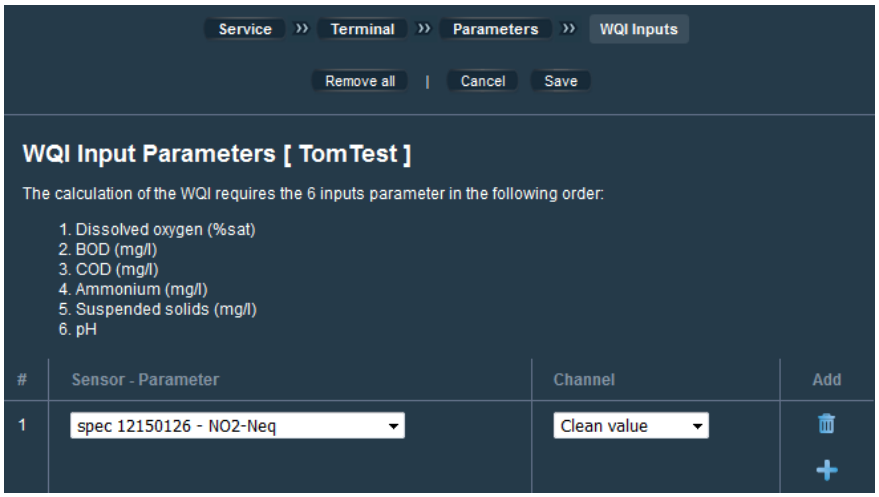
Spectral Alarm

This alarm type is very similar to the *pattern alarm*. The key difference is that instead of a set of parameter data a set of wavelength measurements of a spectrometer sensor is used. Accordingly, in the first configuration step the spectral source (“fingerprint”) that will be used as an input for the spectral alarm needs to be selected.

Water Quality Index

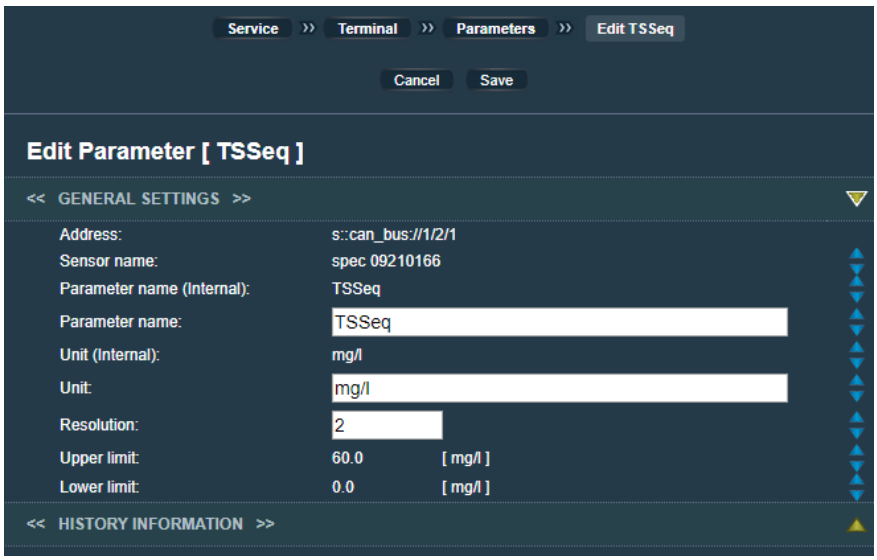
The *Water Quality Index* (*WQI*) used in *moni::tool* is a pre-defined virtual parameter based on the values for dissolved oxygen, BOD, COD, ammonium, suspended solids and pH.

Accordingly, when this parameter type is selected the system asks for adding the respective sensor-parameters.



Edit Parameter Configuration

Pushing the blue *cogwheel* in a parameter's line opens the window *Edit Parameter.*

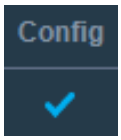


While some settings for the respective parameter are displayed for information only, the settings for *Parameter name*, *Unit* and *Resolution* can also be edited.

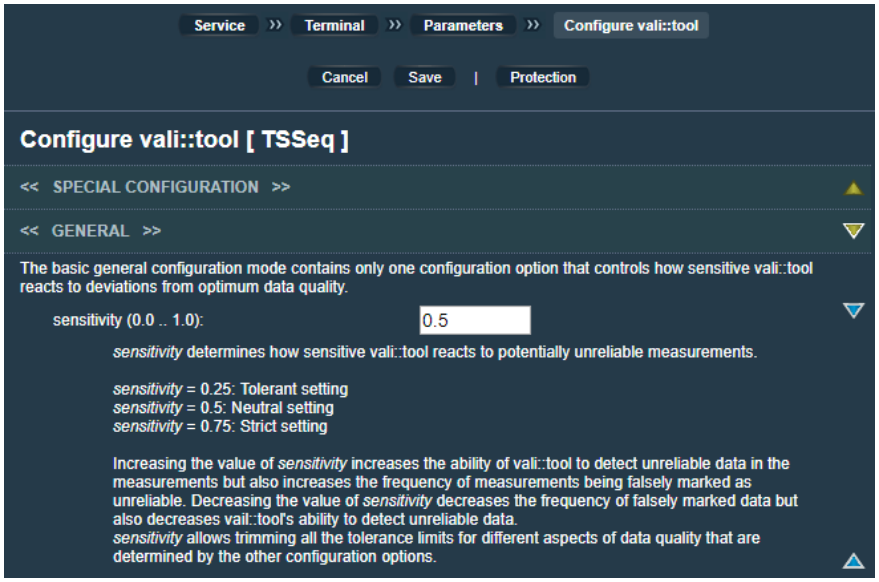
→*Service Level* / *Expert*

- Configuration history is displayed under *History Information.*

Configure vali::tool



Pushing the blue *checkmark* of a **physical parameter** opens the *Configure vali::tool* window where the specific settings for the parameter can be customized.



For the service level *Basic* this is restricted to setting *vali::tool*'s overall sensitivity between 0 and 1 according to the specific measuring situation.

Low sensitivity values mean a tolerant setting, i.e. little false errors are reported but some unreliable data might be ignored.

A value of e.g. 0.75 in return increases the ability to detect unreliable data but might result in a certain number of false alarms.

→*Service Level* / *Expert*

- Configuration of all *vali::tool* settings for physical parameters
- Configuration of virtual parameters

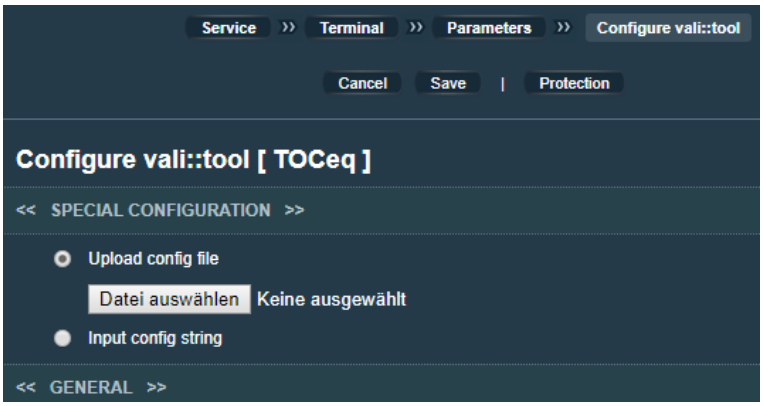
The following table provides an overview on the various settings.

	Name	Possible Settings	Effect	Function
GENERAL	*smoothingPeriod*	0 sec. - infinitely Default = 1500	500 = weak smoothing 1500 = normal smoothing 4500 = strong smoothing	Length of the time window used for the smoothing of clean data.
	rangeCheckUpper	any Default = Infinity	Measurement results above this limit will not be used.	Upper limit for physical / chemical plausible measuring results.
	rangeCheckLower	any Default = -Infinity	Measurement results below this limit will not be used.	Lower limit for physical / chemical plausible measuring results.
	sensitivity	0 - 1 Default =0,5	0,25 = tolerant setting 0,50 = neutral setting 0,75 = strict setting	Defines sensitivity of vali::tool in general.
NOISE DET.	*NOISEinstrumentNoise Threshold*	0 - infinitely Default = 1000,0	The smaller the value the more sensitive the quantification of noise detection.	Upper limit for normal noise level of the parameter.
	NOISEminimumNoise Level	any Default = 0	0 = typical setting -1 = setting for very stable input values	Lower limit for normal noise level of the parameter.

A specific *vali::tool* configuration file is stored on the terminal for each parameter from any s::can sensor and for specific applications (drinking water, effluent waste water, etc.). These configuration files provide appropriate values for the various configuration settings.

For *vali::tool* to work with highest efficiency, the appropriate configuration file needs to be selected as described below.

- Operate the s::can terminal locally;
- Open the *Configure vali::tool* window for the parameter that needs to be configured;
- Push *Upload config file*;
- Push *Browse* and select the appropriate folder
  - Files are stored at */persistent/scan/scandata/vali-tool/vali-tool/STANDARD/* (until *moni::tool* V2.5: d:\s-can\vali-tool\vali-tool\STANDARD\);
  - Select the folder corresponding to the application, the used spectrometer and the needed parameter;



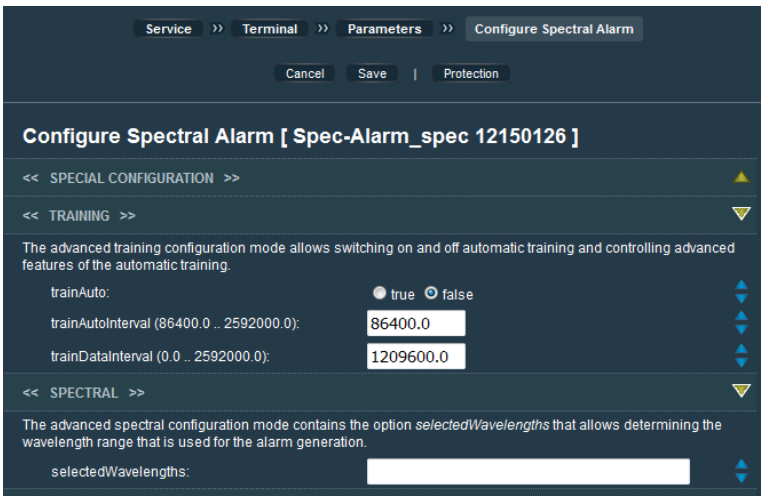
- Open the file *config.txt* within the selected folder;
- Push *Save* for storing the configuration.

Configure ana::tool

Pushing the blue *V* of a **virtual parameter** opens the configuration window for the specific settings of the respective virtual parameter, depending on its type (see example on the right).

For the service level *Basic* this is restricted for ana::tool alarm parameters to read-only!

Please refer also to the information on *ana::tool* in chapter 6.7.



→*Service Level / Expert*

- Configuration of all *ana::tool* settings for training alarm parameters

Configure alarm

This configuration option allows specifying under which conditions an alarm is triggered by the respective parameter.

The available options depend on the parameter type (physical or virtual) and on the active service level.

Physical parameters:

- Upper / Lower limit

→*Service Level / Expert*

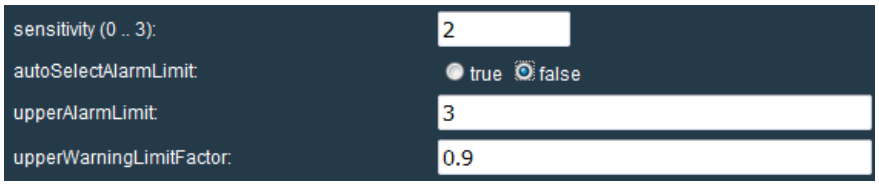
- Warning level

Virtual parameters:

- Sensitivity

→*Service Level / Expert*

- All configuration parameters

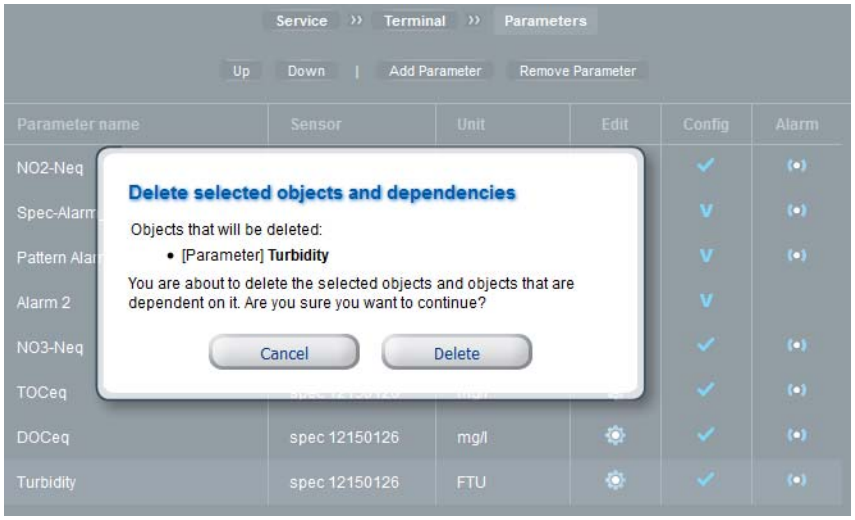


Remove Parameter

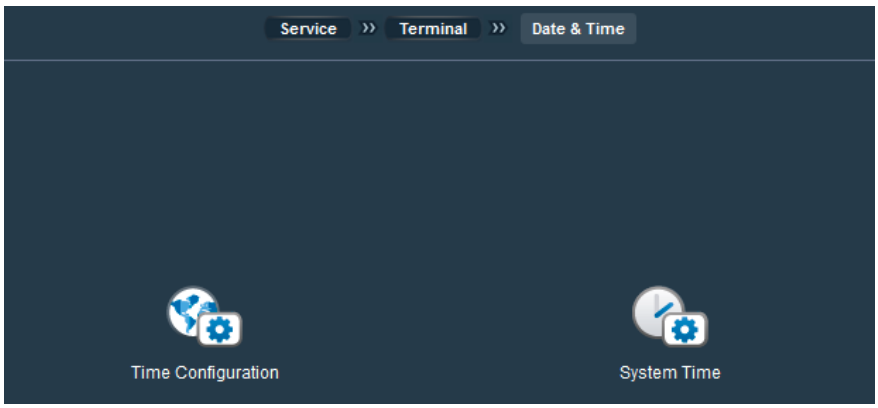
This menu item provides the possibility to delete a parameter from the parameter list.

After selecting one or several parameters and pushing the button *Remove Parameter* a query is displayed if the parameter(s) shall really be deleted.

When pushing *Delete*, the parameter will be removed from *Values* and *Time Series* and no new results will be stored in the database.



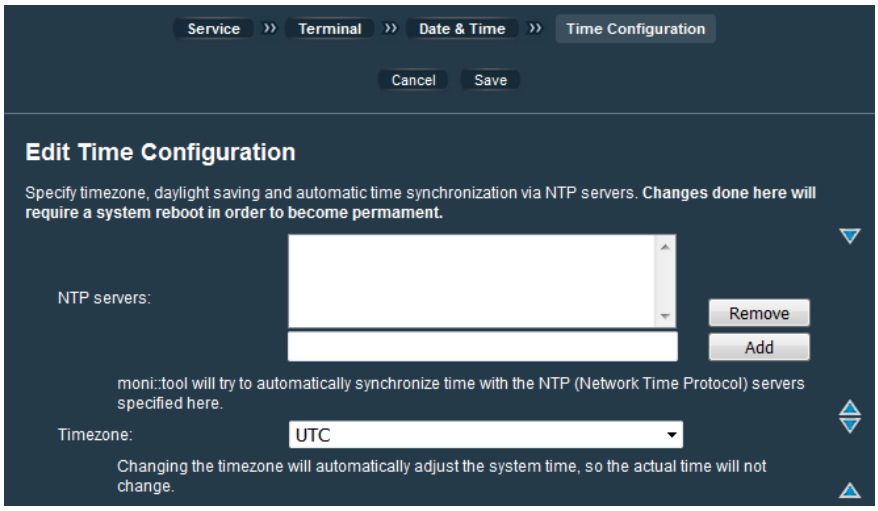
6.4.6 Date & Time



Time Configuration

This menu provides the possibility for configuring NTP (Network Time Protocol) server addresses so that the system will automatically synchronize its time with those servers when the s::can terminal is connected to the internet.

Furthermore, the correct time zone for the monitoring station can be set, using the respective drop down list.



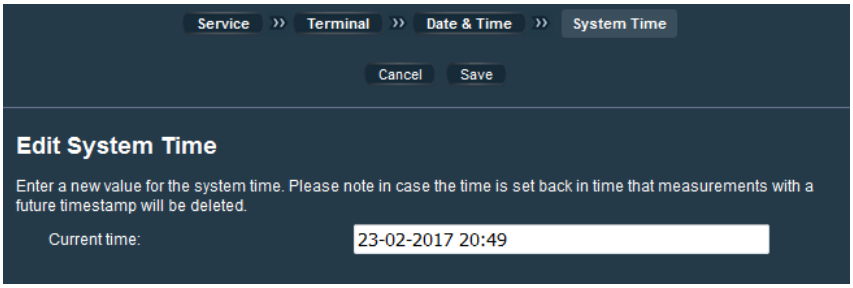
*Available **Quick reference guides**:*

Manuals	*moni::tool*	note_moni-toolV2_TimeSync

Please refer to the s::can Customer Portal!

System Time

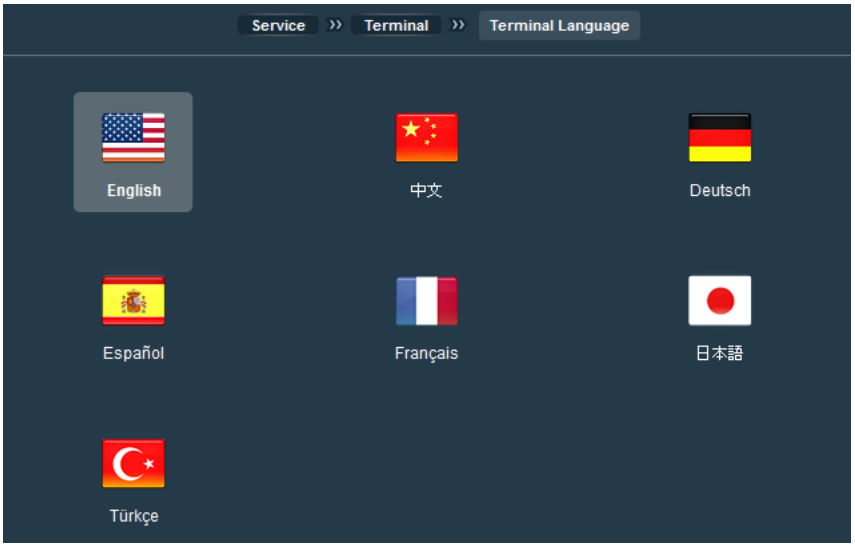
This menu item provides the possibility to set the *con::cube*'s system time and date manually.



6.4.7 Terminal Language

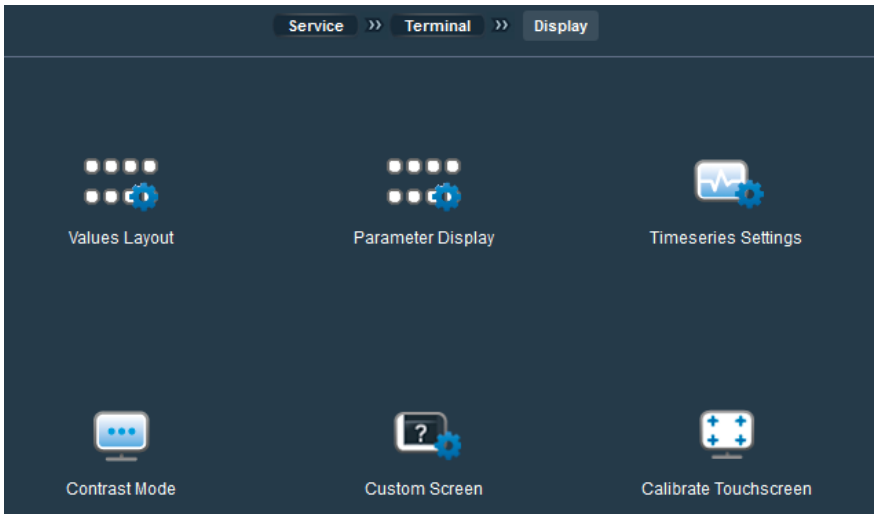
The language of all text in menus and graphics can be changed by selecting one of the available country flags.

When using a web browser for accessing the monitoring system, the used browser language can be set independently from the general setting of the terminal by using the language flag in the main frame (see 0.2.1).



6.4.8 Display

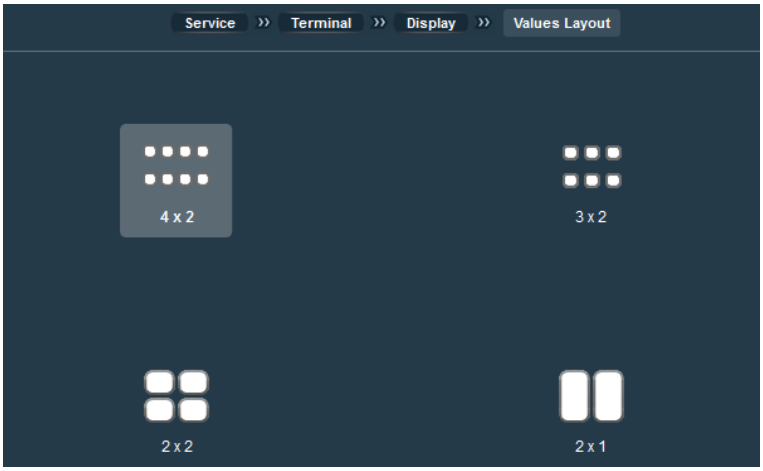
Various options for adapting the moni::tool appearance to specific needs can be configured.



Values Layout

This menu allows selecting one of the pre-defined lay-outs for the *Values* tab.

This is of special interest in cases with less than 7 parameters or when some parameters are of special relevance.



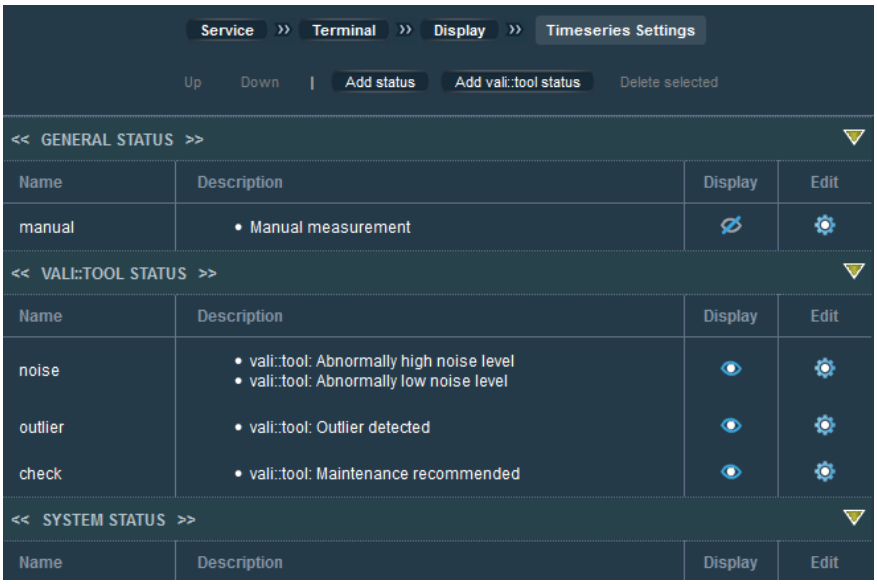
Parameter Display

In this menu the number of parameters to be displayed in *Values* and *Time Series* when no user is logged in can be limited. This can be used for example to hide parameters which serve as input for virtual parameter.

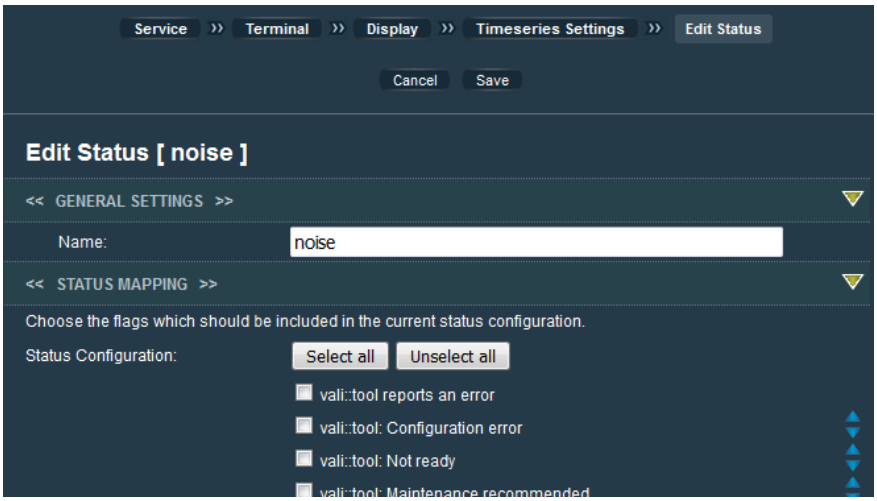
Timeseries Settings

These settings allow the configuration and definition of status messages in terms of *General Status* and *vali::tool Status*.

The predefined status messages can be edited by pushing the respective *Edit* icon, new messages can be added by pushing *Add status* or *Add vali::tool status*, respectively.



Configuring new or existing status messages is done by checking one or several tick boxes and hence defining which kind of information is used to trigger the respective status message.



The order in which status messages are shown can be controlled using the *Up* and *Down* buttons. Whether a configured status shall be actually shown in time series can be controlled by activating or deactivating the *Display* icon.

Unused status messages can be marked and removed pushing *Delete selected*.

Contrast Mode

This function helps optimizing display readability by offering various contrast mode options.

- Classic Mode: Attractive *moni::tool* colors with normal contrast for standard situations
- Day Mode: A white background with black letters for maximized contrast in bright sunlight
- Night Mode: A largely black background for good readability in the dark

Classic-Mode

Day-Mode

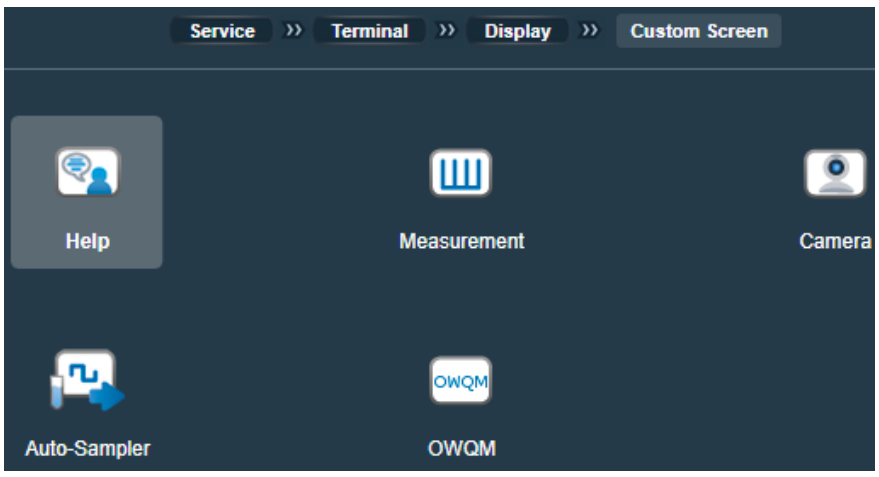
Night-Mode

This service is available only locally on the *con::cube*.

Custom Screen

The appearance of *moni::tool* can be adapted to specific needs by configuring the use of the last entry in the *Tab Menu* on the left side of the screen.

Help
Per default the last tab is *Help*.



Measurement
By selecting this point the last tab is set to *Measurement*. It allows easy switching between the automatic and the manual measurement modes (see also chapter 6.4.1). The exact way the manual measurement will be executed and other options can be configured.

Camera
If a camera is part of the monitoring system (see 6.4.9, p. 56) this point sets the last tab to *Camera*, for which the modes *Gallery* or *Livestream* can be selected.

Auto sampler
If an auto-sampler is configured the last tab can be set to *Auto-Sampler* which allows quick control and easy feedback for the auto-sampler function.

While this respective menu allows configuring some predefined screens, new views can be developed as well. Please get into contact with s::can on this option!

Calibrate Touchscreen

This service is available only locally on the *con::cube*.

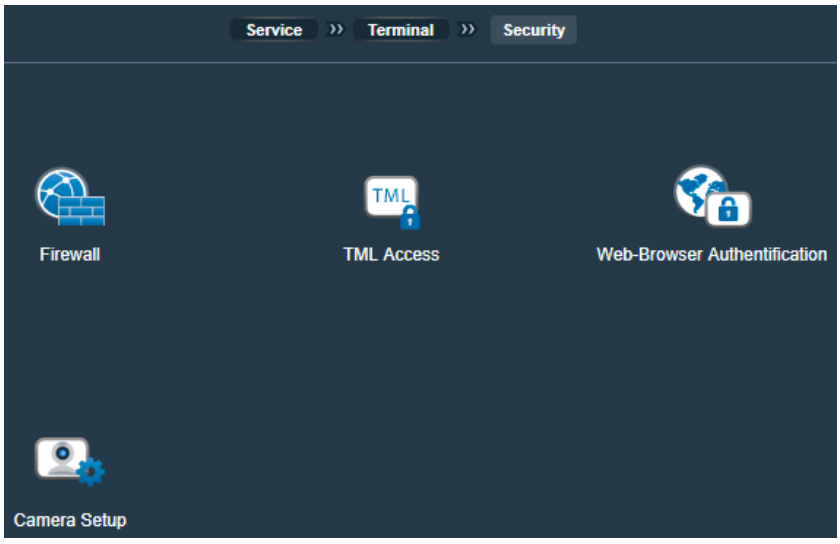
All s::can terminals are delivered with calibrated touch screens. Recalibration can sometimes become necessary due to environmental impacts (e.g. after the terminal is exposed to strong variations in temperature during operation).

When selecting this menu item the system will ask to touch the screen in the four corners (red points). Based on these inputs the screen will be recalibrated. While in *moni::tool* versions up to V2.5 this process is terminated automatically after some time, from V3.0 on it can be terminated only using an USB-keyboard.

In case the s::can terminal can no longer be operated via its touch screen, the touch calibration procedure can be started either remotely via a web browser or locally using a USB mouse. However, the touchscreen calibration itself can then be carried out only directly on the touch screen of the terminal.

6.4.9 Security

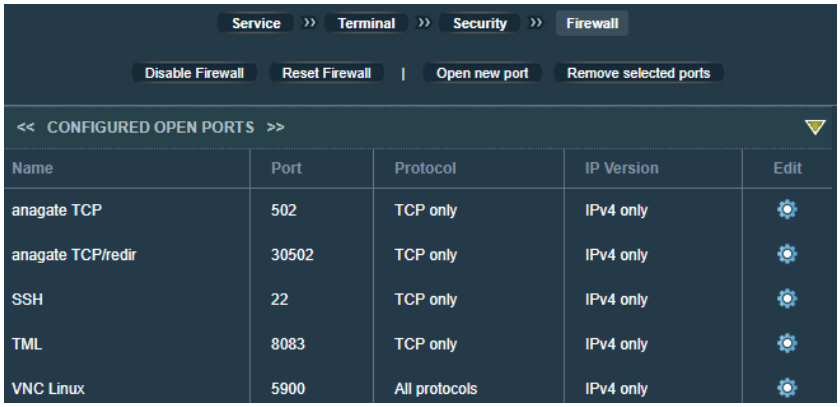
In this menu settings relevant for IT security of the *con::cube* monitoring system can be made.



Firewall

A so called firewall controls the way how a device can be accessed from the internet and how it can communicate to the internet.

For each communication path a specific port can be defined and configured in terms of port, protocol and IP version.



New ports can be opened and existing ports can be edited or removed by using the respective buttons.

⚠ Internet connectivity always bears the risk of unwanted manipulations. Accordingly, these settings should be used restrictively and any changes should be done by experienced staff only. Furthermore, disabling the firewall is NOT recommended.

⇨ *Available **Quick reference guides**:*

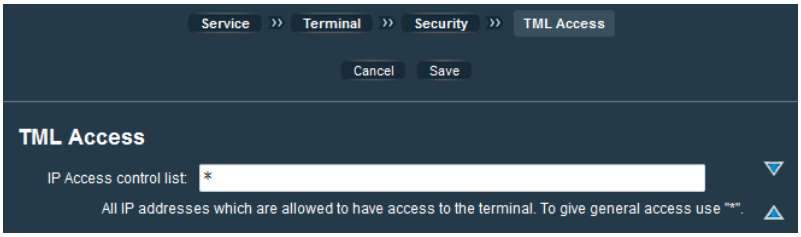
Manuals	*con::cube*	note_moni-tool_Security

Please refer to the s::can Customer Portal!

TML Access

TML is a protocol which can be used to transfer data from the s::can terminal's database. For details on TML please refer to chapter 0).

An IP Access control list can be used to restrict access to the TML server to a list of specific IP addresses.

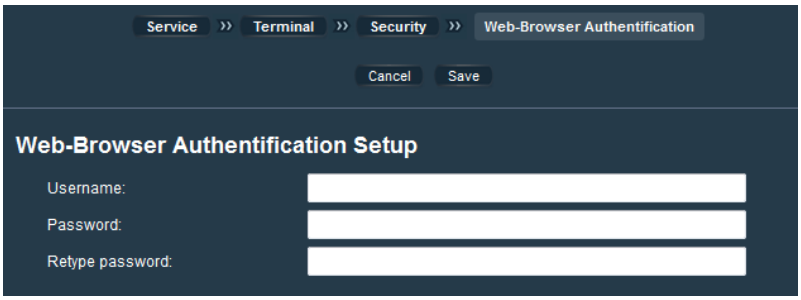


The respective addresses can be entered in this window. The default configuration does not use access restrictions.

Web-Browser Authentification

This menu item provides the possibility to set a password for a user operating the s::can system via web browser.

If a user name and password are set here, connection to *moni::tool* from a remote computer is only possible after the correct username and password have been entered.

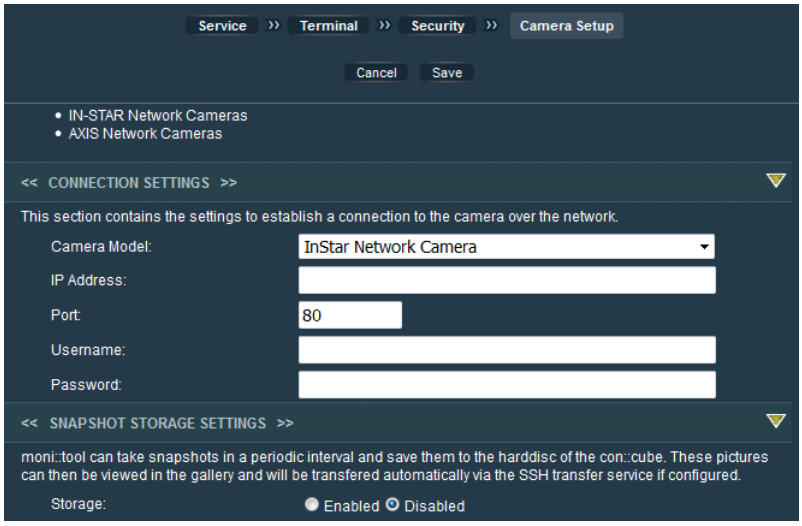


⚠ Once authentication is activated, until *moni::tool* V2.x local access to the system is possible but requires a USB keyboard connected to the *con::cube*. **From *moni::tool* V3.0 on, local access is not possible any longer. This might even prevent any further access to the system!**

Camera Setup

Cameras can be helpful for getting visual information from monitoring sites. *moni::tool* provides the option to connect to cameras accessible via the internet and to include pictures and videos in the collected information (see also chapter 4.4.2).

The information required for connecting to a given camera safely can be entered in the respective fields. Furthermore, it can be specified if camera snapshots shall be taken and stored automatically at certain intervals.

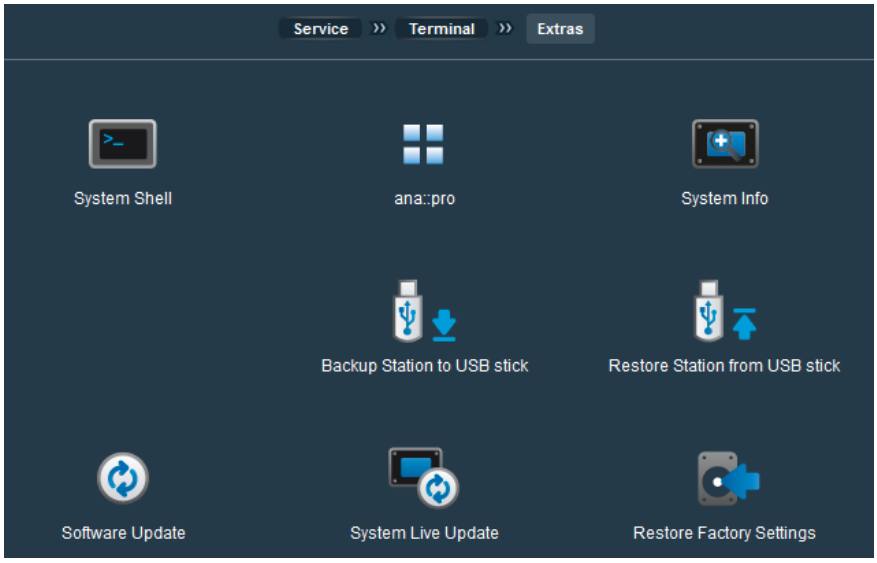


⇨ *Available **Quick reference guides**:*

Manuals	*con::cube*	note_moni-toolV2_Camera_Integration

Please refer to the s::can Customer Portal!

6.4.10 Extras



In this menu issues dealing with system administration are addressed.

System Shell

⚠ This service is available only locally on the *con::cube*. Its use should be restricted to very well trained users as it allows in-depth access to the Linux operating system. Since *moni::tool* V3.1 this icon is not available any more.

ana::pro

This menu item provides the possibility to start the *ana::pro* software allowing the advanced configuration of spectrometer sensors.

⚠ This is a local service and can be started only directly on the *con::cube*. When *ana::pro* is started the *moni::tool* software will be shut down and after finishing the service task the complete monitoring system has to be rebooted.

After this menu item has been selected a screen will be displayed showing all service tasks currently running in the background. Selecting the entry *Continue* on top of the screen will pop up a warning that all services will be stopped. This message has to be confirmed by pushing *Yes*.

⚠ *For moni::tool versions up to V2.5*: Pushing the button *Permanent* will change the startup sequence of the terminal. It will deactivate *moni::tool* and will make ana::pro the active operating software. For undoing this setting the file *D:\Programme\s-can\monitool.bat.disabled* has to be renamed to *monitool.bat*.

Once started, *ana::pro* will search for spectrometers. This process should not be interfered with, etc. by touching the screen.

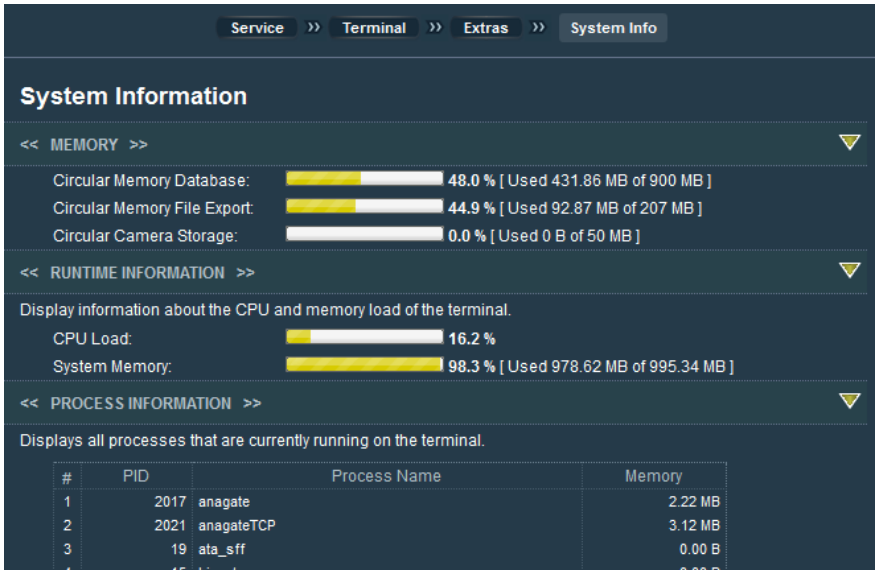
⚠ Please refer to the *ana::lyte* and *ana::pro* manuals for further details on how to work with *ana::pro*.

As soon as all service tasks in ana::pro have been finished, using the option *Exit windows and reboot* will trigger a complete system reboot and *moni::tool* will be started automatically.

System Info

Under this point information on the currently used terminal resources are provided, namely the used shares of various memory types as well as the CPU load.

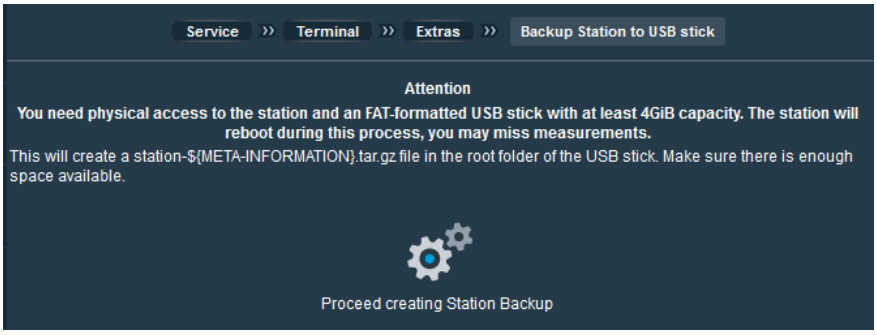
Furthermore, all running processes can be listed.



Backup Station to USB Stick

With physical access to the *con::cube* this function facilitates creating a backup of the whole monitoring station on a USB memory stick. The memory stick must be FAT-formatted with at least 4 GB capacity.

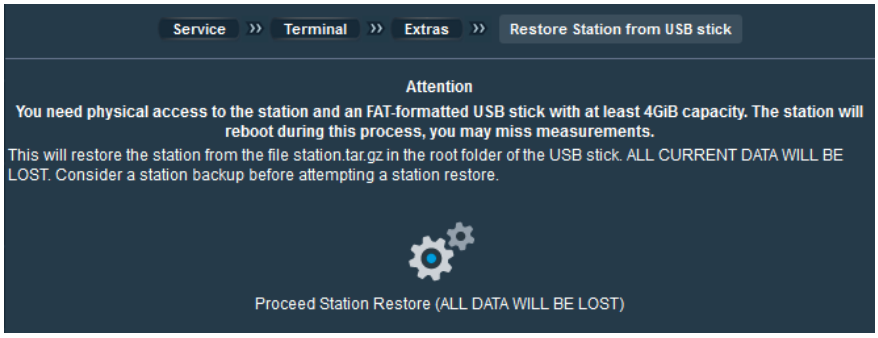
As a result of the backup procedure an *xxx.tar.gz*-file will be created in the stick's root folder.



Restore Station from USB-Stick

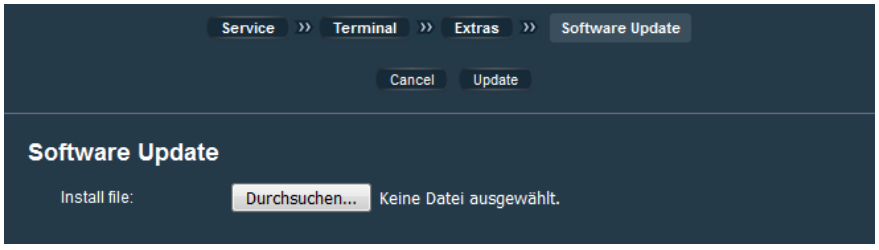
When a station's settings have been saved using the function described above, it can be restored with this respective function. The name of the backup-file has to be ***station.tar.gz*** and must be placed in the **root-directory of the USB-Stick**.

⚠ All settings will be overwritten during the restore process and a station backup should hence be considered.



Software Update

This menu item provides the possibility to run update-scripts (for example to update the firmware of sensors connected to the terminal).



The file containing the update-script can be selected by pushing the *Search*-button, *Update* will start the software guided upgrade procedure.

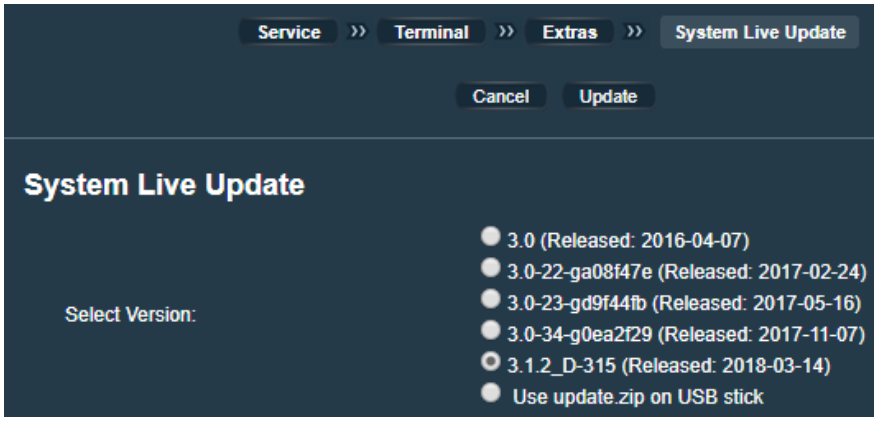
Update of moni::tool software for Windows OS (V2.0) via this function is possible also.

System Live Update

This function provides a comfortable way for updating the *moni::tool* software when the *con::cube* is connected to the Internet.

All available updates will be listed and after selecting the appropriate packages and pushing "*Update*" they will be downloaded and installed automatically.

Alternatively also a USB-Memory-stick containing the update packages can be used accordingly.

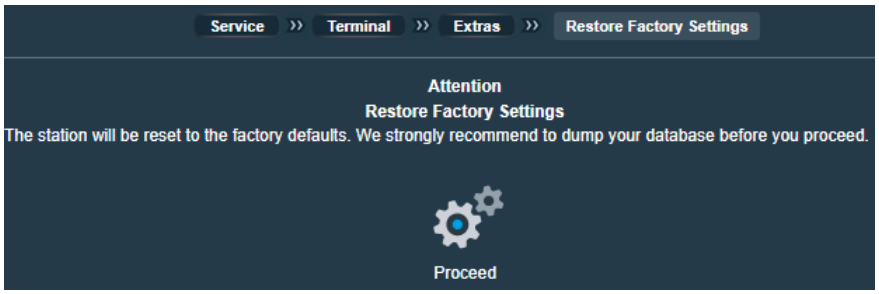


⚠ The Update of *moni::tool* software for Linux OS (V3.0) can be performed within this function only. The name of the update package has to be *update.zip*.

Restore Factory Settings

This menu item provides the possibility to reset the internal *moni::tool* database to factory default (i.e. the original settings the terminal was delivered with).

Restoring to factory settings includes measurement results as well as all settings related to parameters and sensors etc. and will hence lead to a loss of all data in the database as well as to a loss of all sensor information.



When the restore has been completed the s::can terminal should be rebooted before any configurations are performed.

⚠ It is strongly recommended to perform a *Backup Station to USB-Stick* as described above before factory settings are restored.

Task Manager

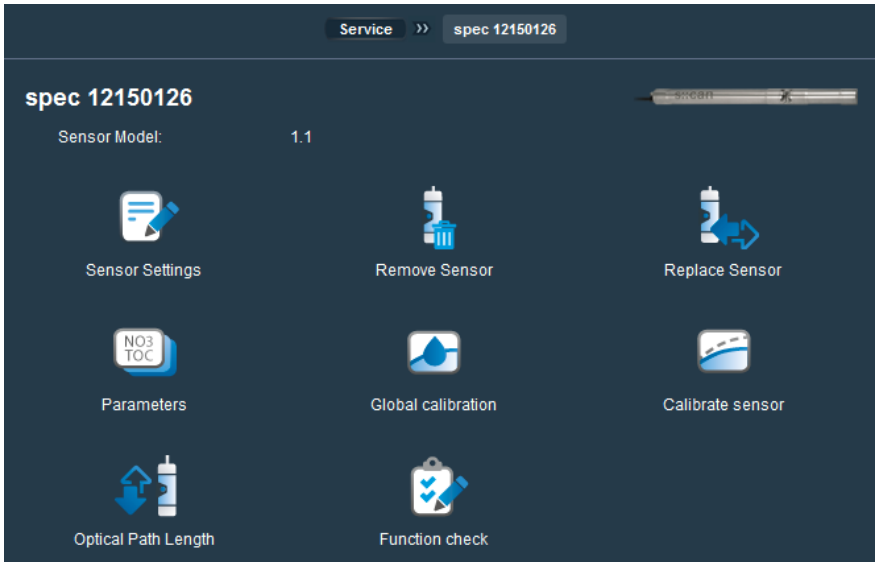
This menu item is available only for *moni::tool* versions up to V2.5 and will start the Windows XP task manager on the terminal itself (which is not available under the Linux operating system).

The launched task manager will be visible only on the screen of the s::can terminal and it will block the other applications from being visible.

6.5 Sensors

The sensor area of the *Service* tab provides an overview of all installed sensors.

Further configuration and maintenance options for each individual sensor are launched by pushing the respective sensor's icon.



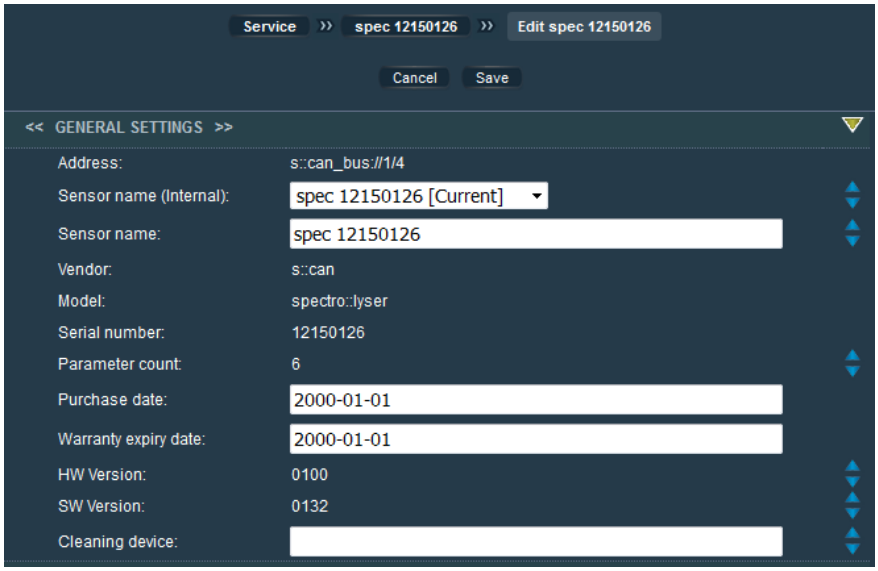
6.5.1 Sensor settings

This menu provides the possibility to edit the sensor settings that can be modified by the user.

Changeable settings include the *sensor name*, the *purchase date* and the *warranty expiry date*. Settings controlled by the system will also be displayed for information.

If a spectrometer is used in combination with a terminal which has an own cleaning valve (e.g. *con::nect*), entering a text in the field *Cleaning device* creates a new cleaning device (see 6.10) using the URL of this sensor.

Individual settings exist for the different sensor types. Please refer to the according sensor manuals



6.5.2 Remove sensor

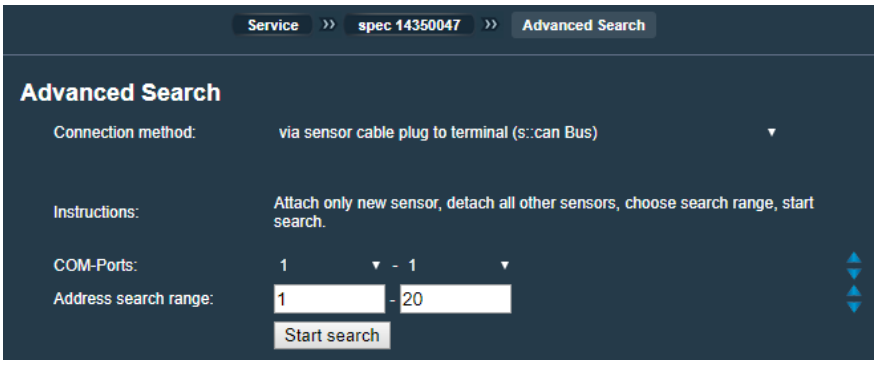
This menu removes the sensor and all associated parameters from the monitoring station configuration. Data collected by this sensor remain stored in the database and can still be displayed in Time Series and Fingerprint.

After selecting the menu item a user message will be displayed and *Delete all* has to be confirmed for removing the sensor.

6.5.3 Replace Sensor

This menu supports the exchange of a sensor with the same type of instrument. When using *Replace Sensor* all settings that can be configured in *moni::tool* remain unchanged and will be applied to the new sensor.

Time series of readings will be continued after sensor replacement as sensor name and parameter name are identical.



If it is not possible to exchange the sensor *moni::tool* will show the differences between the old and the new sensor and offer the possibility to install the replacement sensor as a new one (button *Install as new Sensor*).

*Available **Quick reference guides**:*

Manuals	*con::cube*	note_moni-toolV2_sensorinstall+replace

Please refer to the s::can Customer Portal!

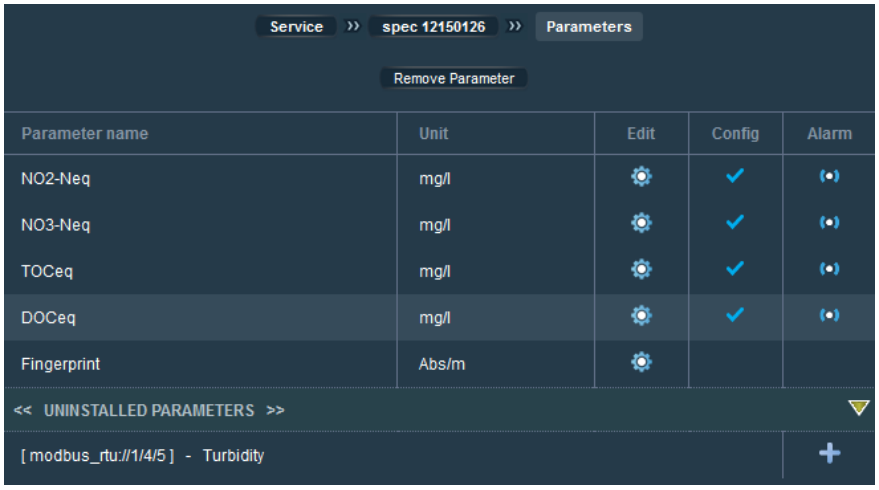
6.5.4 Parameters

The information managed in this menu is closely linked and partly identical with the one described under Service / *Terminal* / *Parameters*. Please refer to chapter 6.4.5 for further details!

The main window provides a list of all currently configured parameters measured by the respective sensor as well as the entry points for further configuration steps on the right of the table.

Furthermore, parameters which can be measured by the sensor but are currently not configured are listed in the lower part of the window. They can be installed by pushing the blue + on the right side.

Installed parameters can be selected by pushing their name and then be removed by pushing *Remove Parameter*. Thus they become available in the “uninstalled” list afterwards.



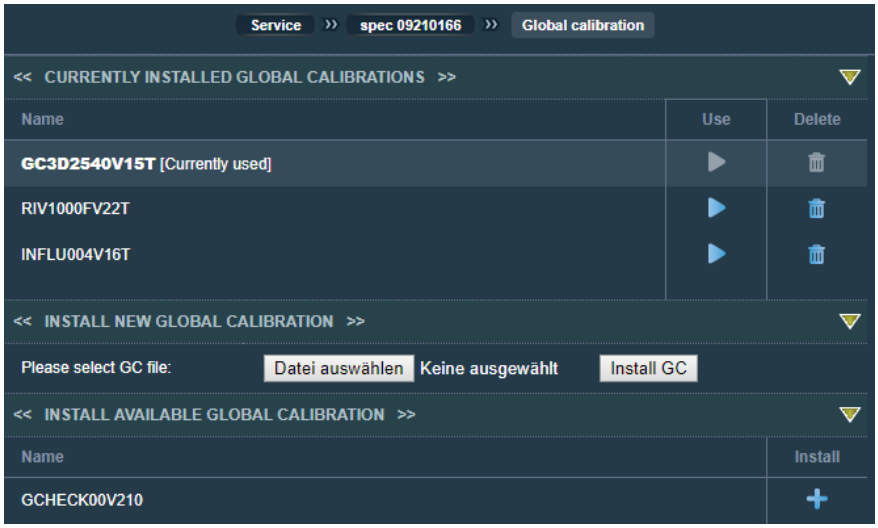
The table below provides an overview on the possibilities for customizing the various parameter settings as well as the manual page on which more detail are provided in the *Terminal* / *Parameters* chapter.

Edit		Configure general settings (name, unit, resolution)	p. 49
Config		Configure vali::tool (sensitivity, noise detection)	p. 49
Alarm		Configure alarm settings (upper/lower limit, warning level)	p. 51

6.5.5 Global calibration

This point is valid for spectrometer sensors only.

A global calibration defines for a certain setting how measurements from a spectrometer are transformed into parameter readings. This transformation is influenced by the spectrometer type, by the path length and by the medium. Accordingly, global calibrations exist for a variety of settings.



The *Global calibration* menu provides the possibility to select the global calibration used by the spectrometer from calibrations actually stored on the sensor.

For activating one of the listed calibrations it is selected by pushing the blue triangle symbol on the right side.

Also new global calibrations can be uploaded to the spectrometer. Push the button *Browse / Choose File…* for searching the global calibrations that are available on the terminal. The file name will be displayed beside the button.

When the correct one has been selected it can be uploaded to the spectrometer by pushing the *Install GC* button.

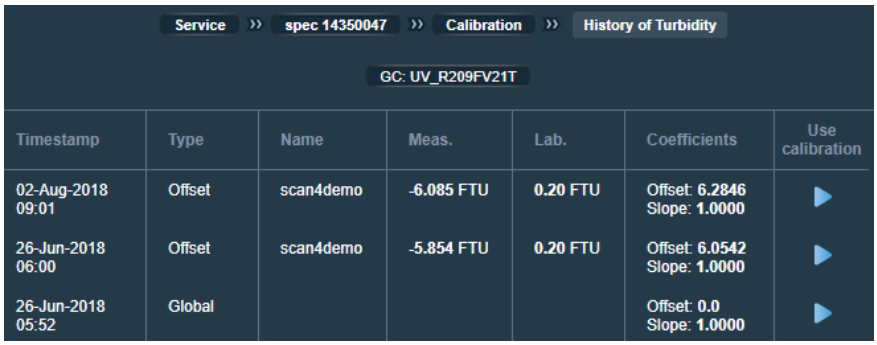
6.5.6 Calibrate sensor

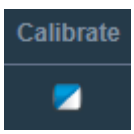
The parameter calibration is closely linked to the taking of samples and relevant information can hence also be found in chapter 6.3. Since calibration approaches depend also on the type of sensor, please refer also to the respective sensor manual.

The main window shows a list of all parameters being measured by this sensor. Pushing the blue triangles will open more information about actual used calibration coefficients for this parameter.

The *History* icon opens a logbook showing date and time of all performed calibration procedures up to now, including the used sample readings and the corresponding laboratory values (see lower figure on the right).

Historic calibrations can be activated by pushing the *Use calibration* icon if this function is supported from the sensor.





Pushing the blue *Calibrate* icon opens the calibration window which allows actually performing calibrations on the basis of samples with sensor readings and corresponding laboratory measurements. Respective details have been provided in the chapter on *Sample & Calibration* (6.3). There also the various types of calibration are explained. For detailed information about the calibration of a specific sensor please refer also to the manual of this measuring device.

When the calibration window is displayed several options exist and a series of steps need to be made:

- The leftmost button in the headline allows toggling the graph view;
  - As *Status View* the continuously updated readings of the parameter will be displayed numerically and graphically which can be helpful to assess the stability of the measurement (e.g. of electrodes). If a quality number of the sensor is available additionally (e.g. *ammo::lyser*) this will also be displayed as a number and in a yellow time series;
  - As *Samples View* the currently selected samples for performing a calibration are displayed (sensor and laboratory measurements);
- A new measurement can be started by pushing the *Trigger Measurement* button;
- The button above the graph shows the currently selected calibration type and allows changing it.

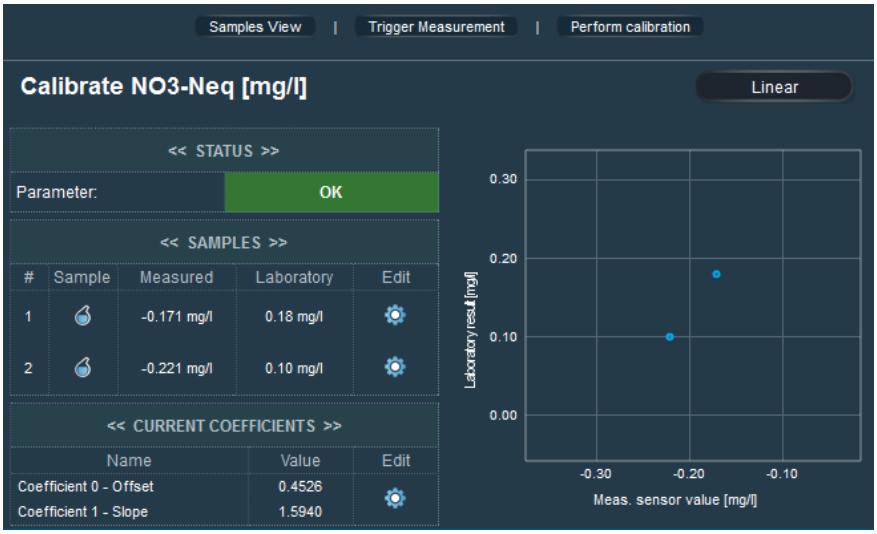
Basically two different groups of calibrations need to be distinguished as is described in the following sections.

Calibration with 1 or 2 samples (*Offset*, Slope, *Linear*)

The calibration types *Offset, Slope* and *Linear* make use of the sample values stored on the respective sensor (please refer to chapter 6.3).

The current values of the stored samples are displayed and can also be edited (e.g. for entering laboratory values). By pushing the *Sample* icon a new measurement can be triggered and stored onto the sensor. Please note that the displayed value is the Raw value (e.g. mV in case of electrodes) or based on the active global calibration (in case of spectrometer), respectively.

The result of the laboratory analysis can be entered into the field *Laboratory* by editing the respective sample. *Save* stores the sample values onto the sensor and hence replaces the old sample values.



Pushing *Perform Calibration* starts the selected calibration procedure with the two samples as displayed (select *Samples View* for checking graphically). When calibrating is finished a user message will inform if it was successful. In case of an error the reason will be displayed to the user in red letters (e.g. *Please enter at least lab values for 2 samples*).

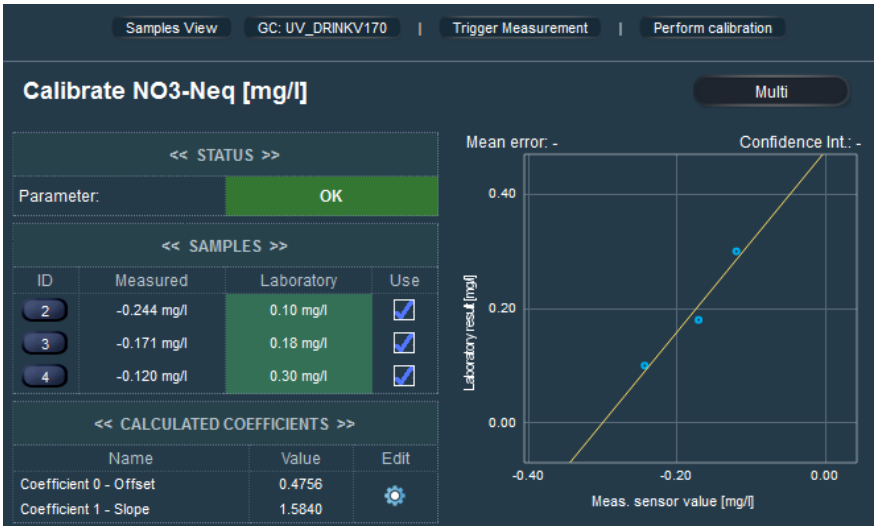
Under *Current Coefficients* the actual values are displayed.

⚠ Some sensor types also support writing coefficients directly onto the sensor by pushing the *Edit* icon (none of the coefficients can be *NaN*). Please refer to the sensor manual for respective information.

Calibration with more than 2 samples (*Multi*)

Please note that only parameters from i::scan and spectrometer probes support *Multi* calibration.

The *Multi* calibration type makes use of several samples based on a statistical method. While the calibration types *Offset* and *Linear* use measurement results stored directly on the sensor itself and are therefore limited to two samples (as explained above), the *Multi* calibration builds on results stored in the *moni::tool* sample database (see also chapter 6.3).



The calibration type *Multi* uses a linear fit to obtain a calibration line with the smallest possible error between the stored sensor readings and the laboratory values. Accordingly, the calibration screen for *Multi* looks slightly different and offers some additional features.

In the *Samples* area the ID and values of all stored samples valid for the respective sensor / parameter are displayed. Pushing the ID number switches to the respective *Sample list* results as explained in chapter 6.3.3.

When preparing the calibration it is possible to decide which samples shall actually be used by ticking the individual check boxes. This has no impact on the samples database but influences only the on-going calibration. As samples are checked or unchecked the resulting correlation between laboratory results and raw values in the *Sample View* is updated automatically. Also the resulting values for *Mean error* and *Confidence Interval* above the graph are updated automatically. Once the calibration results are considered satisfying, pushing *Perform calibration* finalizes the process.

*Available **Quick reference guides**:*

Manuals	*con::cube*	note_moni-toolV2.2_sensorcalibration

Please refer to the s::can Customer Portal!

6.5.7 Optical path length

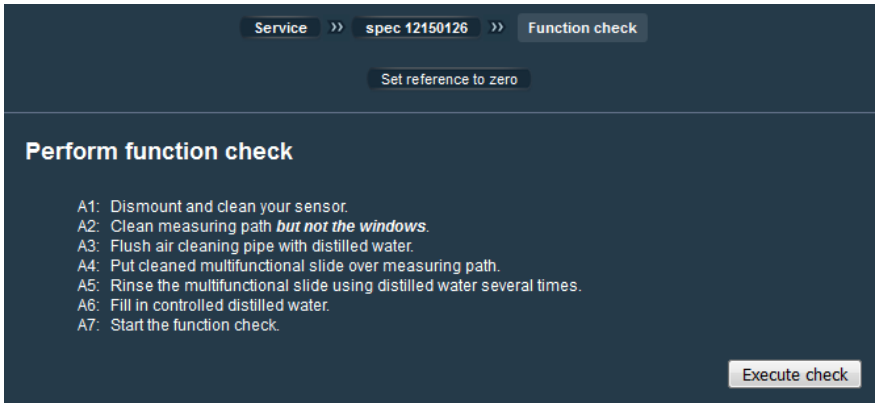
This function can be used to change the optical path length for spectrometer sensors. This function is available only when working on a *con::cube* terminal locally. Please get in contact with your s::can support in case of the need to change OPL.

6.5.8 Function Check

A detailed step-by-step explanation is displayed on the main screen.

Pushing the *Execute* button starts the checking procedure.

Please refer to the manual of the sensor for further details on possible results and consequences of functional checks.

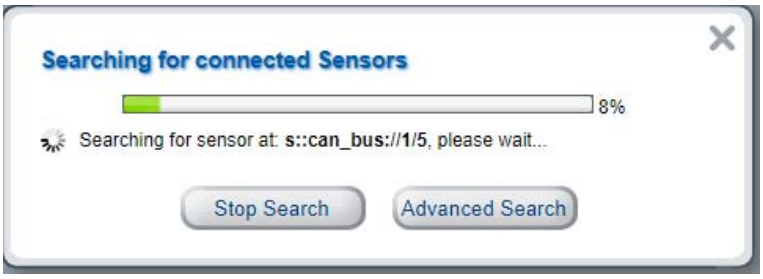


This function is available only for spectrometer probes and *i::scan* sensors.

6.6 Sensor installation

The installation of a new sensor is initiated by pushing a sensor symbol titled *Add new sensor* on the main *Service* window.

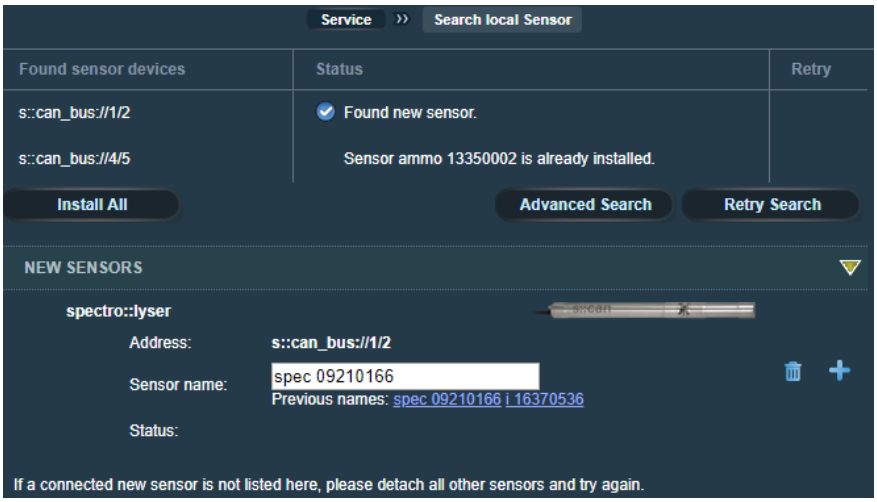
moni::tool will automatically initiate a search for connected devices.



Once the search is completed, a list showing all connected sensors will be displayed and the sensors that need to be installed can be added using the blue *plus* symbols in the table.

Detected new sensors can be ignored / not installed by pushing the recycle bin icon. The search can be interrupted using *Stop Search*.

Basically it is possible to install several sensors at the same time if they are of different types. When installing several sensors of the same type they most probably have to be installed separately.



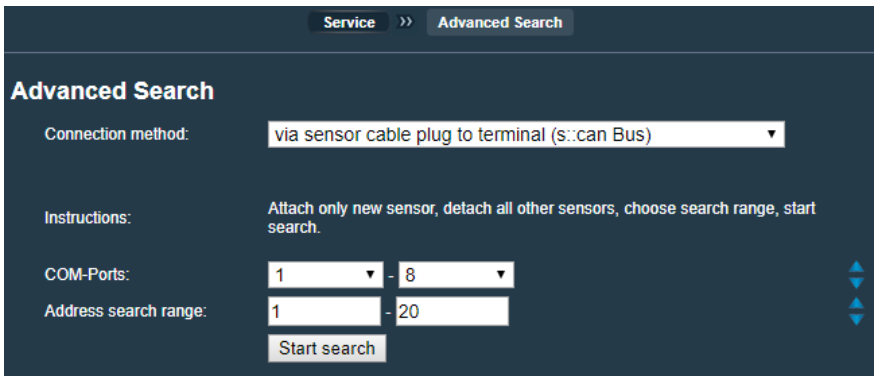
6.6.1 Installation of s::can sensors

As soon as *moni::tool* has completed the searched it lists up all found sensors (see image above). The Status column informs if the sensor is already installed or if it is new (*Found new sensor*). Pushing the blue *plus* icon or the *Install All* button installs the respective new sensors. Once *moni::tool* has finished the initialization of the new sensor (Status: *Installing sensor, please wait...*) it will switch back to the *Service* tab and show the new sensor in the system overview.

The automatic sensor installation will search for s::can sensors or s::can terminals connected to COM 1 (plug for spectrometer), to COM 4 (sys-plugs) and to COM 5 (terminal connection) (all modbusRTU only).

If the installation shall be performed manually (e.g. because a sensor has not been found during the automatic installation procedure) a manual / advanced installation is launched via *Advanced Search*.

After selecting the correct connection under *Connection method* (for s::can sensors this will be the default setting (*via sensor cable plug to terminal (s::can Bus)*) the correct COM-Port and Address have to be entered.

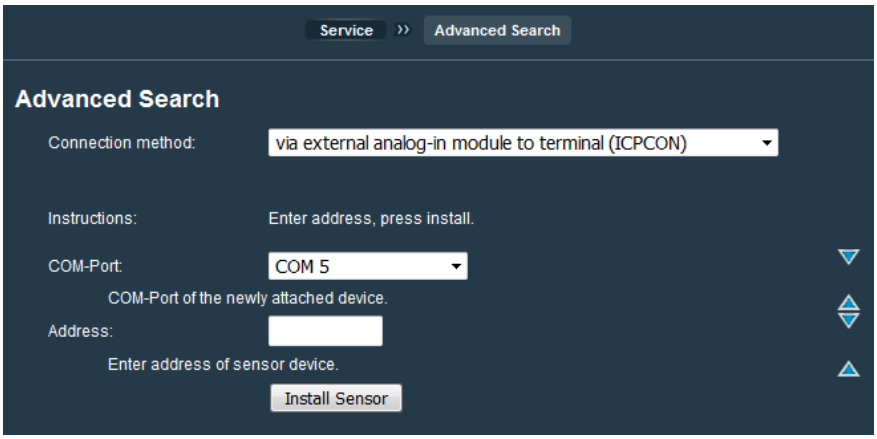


If this information is not available *moni::tool* will search the complete range as indicated on the screen and no other sensors should hence be connected. Pushing *Start* triggers the search.

6.6.2 Installation of non-s::can-sensors via an analog interface

Analog sensors producing a 0/4-20 mA output need to be connected to an analog input module of the *con::cube*. This can either be an internal module (D-315-in-mA) or an external analog input module (D-322-8) connected via COM-port.

Analog input devices are installed by implementing the following steps:

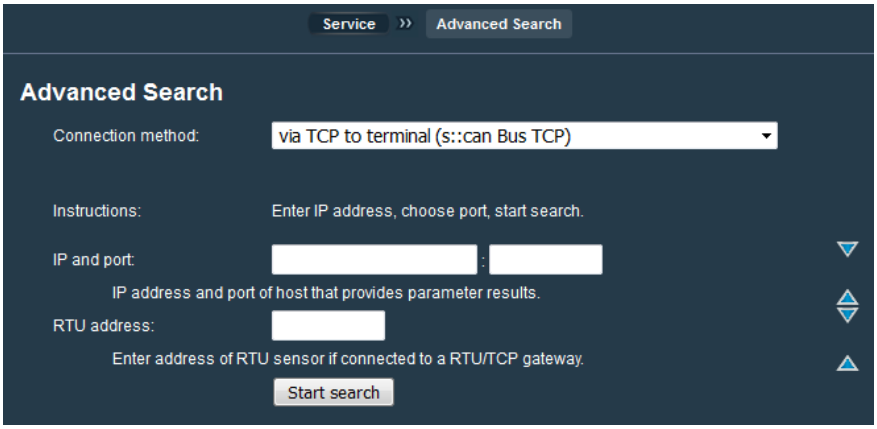


- Selection of the correct *connection method* in *Advanced Search* (see image above; either *via internal analog-in module to terminal* or *via external analog-in module to terminal (ICPCON)*);
- Selection of the correct channel of the internal module or entering the correct COM-port and address of the external module (if no channel is displayed it needs to be checked if an internal module is installed in the menu *Service / Terminal / con::cube / Internal Modules*; see chapter 6.4.2,p. 41);
- Pushing *Install Sensor*;
- Entering a sensor name and pushing *Save*;
- Configuration of a parameter that shall display and record the data from the analog sensor in the menu *Service / Terminal / Parameter* (see chapter 6.4.5).

If multiple analog input devices need to be installed the steps above have to be repeated for each device.

6.6.3 Installation of Sensors via TCP

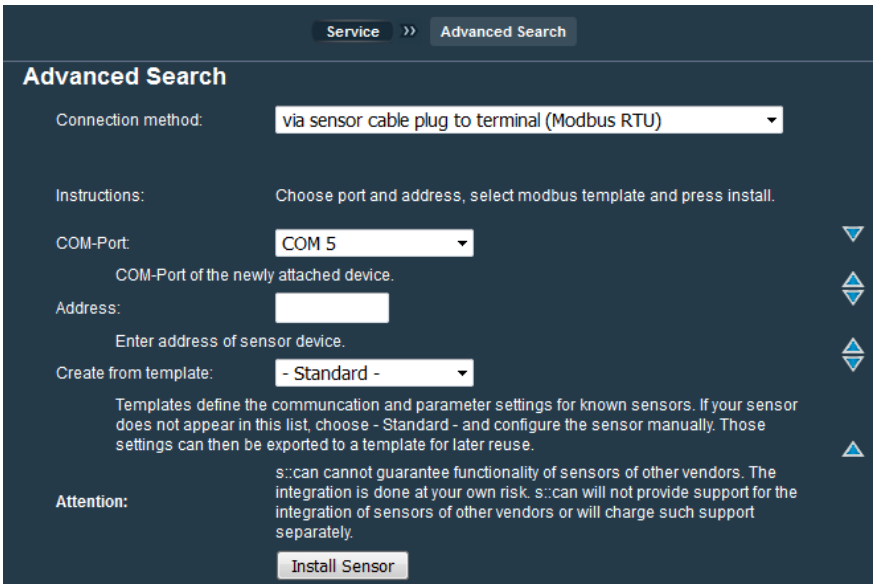
Parameters from other s::can terminals (e.g. another *con::cube or con::stat*), s::can sensors using Modbus TCP or sensors connected to a Modbus RTU/TCP gateway can be linked into *moni::tool* over an Ethernet (TCP) connection by the following steps:



- Selection of the correct connection method in Advanced Search (see image above; via TCP to terminal (s::can Bus TCP));
- Entering the correct address of the sensor (a correct address for TCP communication always contains an IP address and a port number; an RTU address is needed in case of having an RTU sensor connected to a RTU/TCP gateway, only); For con::cube and con::stat the default TCP port used is 502.
- Pushing *Start search*;
- When the sensor has been found: checking sensor name and pushing *Save*.

6.6.4 Installation of non-s::can sensors via Modbus

It is also possible to connect third party sensors to the s::can terminal either via Modbus RTU or via Modbus TCP. In both cases the protocol "mapping" of the sensor has to be configured manually by the customer. The installation itself will be performed by the following steps:



- Selection of the correct connection method in Advanced Search (via sensor cable plug to terminal (Modbus RTU) or via TCP to terminal (Modbus TCP));
- Entering correct COM-Port and Address of the sensor (for RTU connection) or the correct address of the sensor (a correct address for TCP communication always contains an IP address and a port number; an RTU address is needed in case of having an RTU sensor connected to a RTU/TCP gateway, only);
- Selecting the correct template for defining sensor communication (note that all templates are stored on the terminal and that templates can be selected, created or uploaded using the menu Service / Terminal / Settings / Templates; see chapter 6.4.4);
- If no suitable template is available for the sensor the option "– Standard –" should be selected. Protocol "mapping" details can then be entered later on (additional information is available on the s::can Customer Portal);
- Pushing *Install Sensor*;
- When the sensors has been found: checking sensor name and communication settings (for RTU: Baud-rate, Parity, timeout, retries, waiting time; for TCP: timeout) and pushing *Save*.

⇨ *Available **Quick reference guides**:*

Manuals	*moni::tool*	note_moni-tool_Install3rdParty_Modbus_Sensors.pdf

Please refer to the s::can Customer Portal!

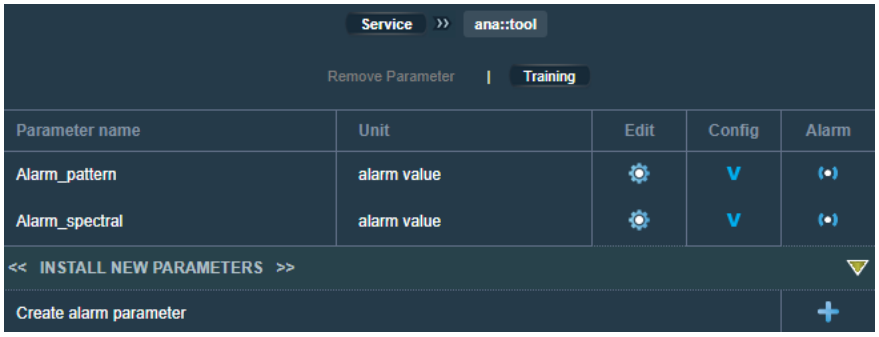
6.7 ana::tool

The *ana::tool* section of the *Service* tab provides access to all functions of the *ana::tool* software and to the respective virtual parameters.

6.7.1 General

The upper part of the main window lists all currently configured virtual parameters handled by *ana::tool* (for information on virtual parameters, please refer also to chapter 6.4.5).

As for all parameter three columns provide access to various settings and options (this is identical with the possibilities provided under *Service/Terminal/Parameters*!).



The lower part of the main window allows creating additional alarm parameters based on the principles of spectral or pattern alarms. This function corresponds to the *Add Parameter* function under *Service/Terminal/Parameters* as described in chapter 6.4.5.

Edit allows modifying the displayed parameter name, the displayed unit, the used resolution as well as the valid measurement range.

→*Service Level / Expert*

- Configure internal parameter name
- See configuration history information

Config focuses on adapting the automatic training function. Besides generally activating and deactivating it, also the used data interval as well as the interval in which automatic trainings will be triggered can be set.

→*Service Level / Expert*

trainAuto	0 1 (default)	OFF ON	Used to switch automatic training on and off.
trainAutoInterval	21 600 - 2 592 000 sec. 86 400 sec. (default)	equal to 6 hours - 30 days equal to 1 day	Time interval between automatic trainings.
trainDataInterval	0 - 2 592 000 sec. 1 209 600 sec. (default)	equal to 0 hours - 30 days equal to 14 days	Determines amount of historical data used for the training (interval in seconds).

Information on the settings can be displayed by pushing the blue arrows on the right side.

Furthermore, also special configurations can be used by either uploading a configuration file or by entering a respective command directly (all service levels).

Alarm is used to configure the conditions for actually triggering an alarm with the respective alarm parameter. The way the collected information is used is defined mainly by the proxy-indicator *sensitivity*. It can be set to one, two or three, where a higher number means that relatively many situations will be considered as alarms. Accordingly, it is very likely that all real alarms will be detected while at the same time the share of false alarms which need to be rejected (see 5.2) will increase.

The following configuration settings can be used for manually customizing the pattern alarm.

Name	Possible Settings	Effect	Function
sensitivity	0 1 2 (default) 3	Alarm switched off Low sensitivity Medium sensitivity High sensitivity	Used to control the sensitivity of the alarm parameter.

→*Service Level / Expert*

AutoSelectAlarmLimit	0 1 (default)	Upper alarm limit is controlled by option *upperAlarmLimit*. Upper alarm limit is determined automatically depending on the selected sensitivity	Determines whether the sensitivity settings override the settings for the upper alarm limit.
upperAlarmLimit	-infinity < upperAlarmLimit < +infinity 3 = default		When the alarm value exceeds the *upperAlarmLimit* the parameter triggers an alarm.
upperWarningLimitFactor	0 ≤ upperWarningLimitFactor ≤ 1 0.9 = default		When the alarm value exceeds the *upperWarningLimitFactor* the parameter triggers a warning.

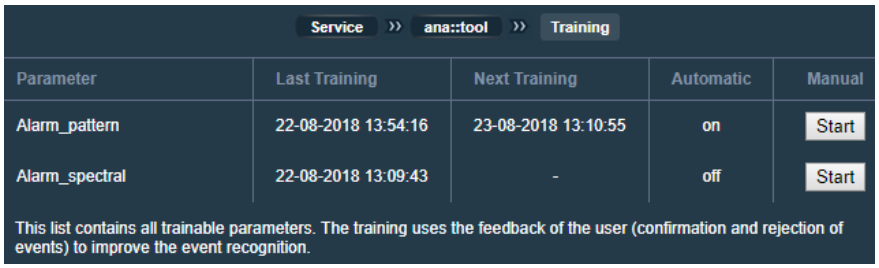
Besides, also the possibility for uploading information on a special alarm configuration exists.

6.7.2 Training

As mentioned already under *Alarm* (see chapter 5.5), virtual alarm parameters should usually be trained regularly on the basis of historical data. This is necessary to train alarm parameters in order to adapt the alarm response to a particular application on the basis of feedback given to the system by confirming, rejecting or adding missed events

Pushing *Training* opens an overview of all currently installed trainable alarm parameters.

Furthermore, for each parameter time and date of the last successful training as well as that of the next scheduled automatic training are shown.



Per default all alarm parameter algorithms are automatically trained once a day. However, the training frequency can be modified under *Config* as explained above.

If for an alarm parameter automatic training is disabled, the date and time fields for the next scheduled training will show only a dash and in the *Automatic* column it will be marked *off*.

Trainings can also be started manually by pushing *Start* in the row of the respective alarm parameter.

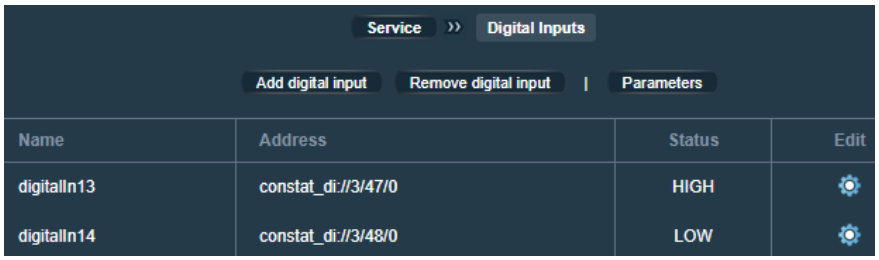
Parameters that are currently being training are marked with *training in progress* in the *Next Training* column.

The comprehensive functionalities of ana::tool require well-trained staff for being used according to their full potential. Please contact your s::can partner for information on respective training courses!

6.8 Digital Inputs

The s::can terminal can be equipped with internal digital inputs or an external digital input module can be connected to it. Each digital input can be used as an independent sensor.

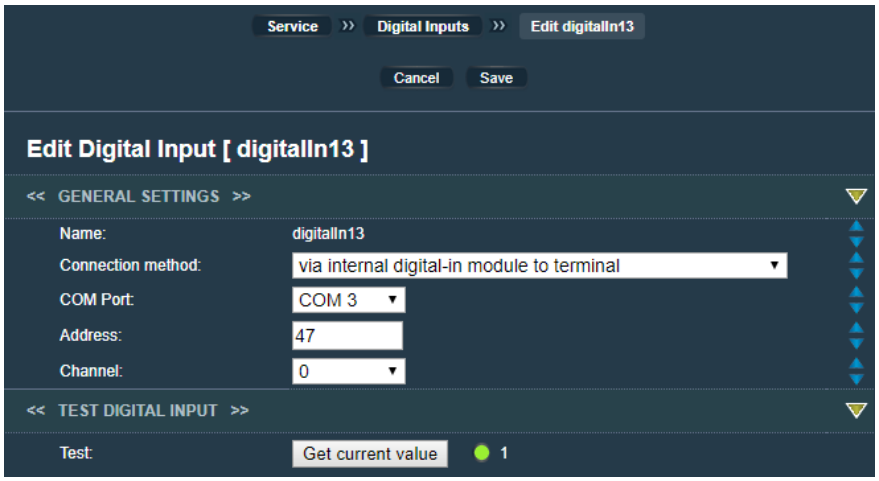
Internal digital inputs will be initialized automatically during initial start-up and will be listed by pushing the *Digital Inputs* icon beside the *Terminal* icon in the *Service* tab main window.



If internal digital input modules of the *con::cube* are not detected automatically or when external digital input modules shall be used, the inputs have to be initialized manually using the *Add digital input*-function.

The following steps are needed:

- Selecting the correct connection method (e.g. *via internal digital-in module to terminal*)
- Entering information on the used COM Port, Address and Channel (please refer to the manual of the external input module for the correct settings).
- All settings and modifications will be stored when pushing *Save*.



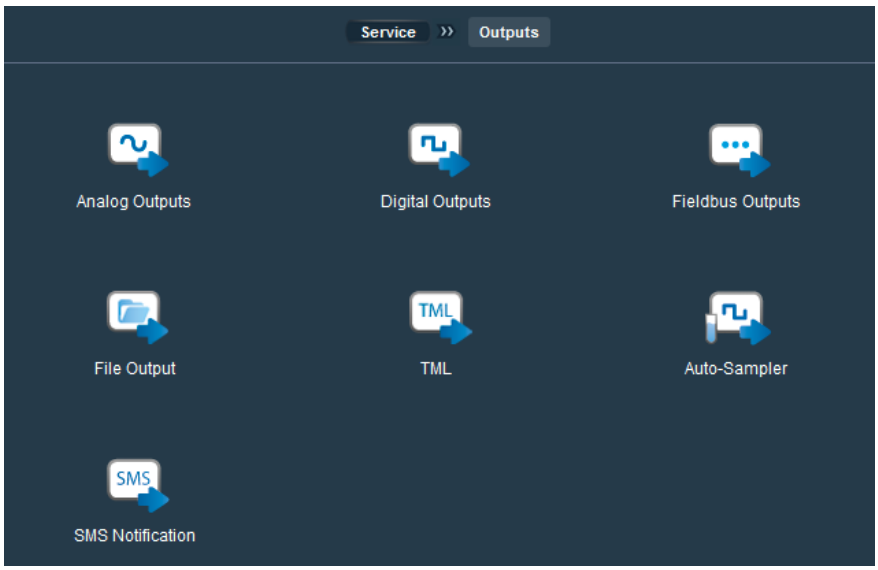
All settings can be modified later by pushing the Edit icon.

Once the digital input has been configured, it is possible to test the current state of the device by pushing *Get current value*. The state of the input is then shown by the LED icon and the number next to it where grey and 0 mean the state is LOW while bright green and 1 mean the input state is HIGH.

For each digital input a parameter can be definned by pushing *Parameter*, which will be used to read the input status.

A defined input can be deleted by selecting it in the overview and pushing *Remove digital input*.

6.9 Outputs

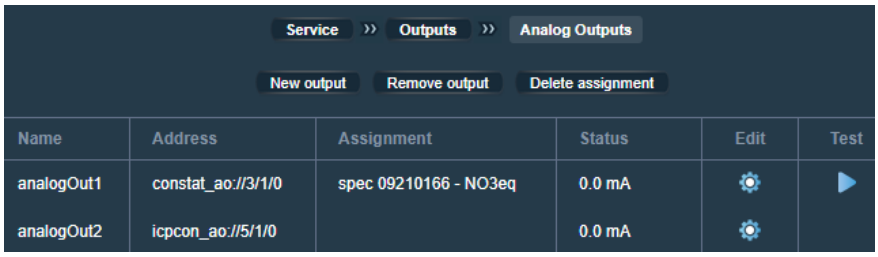


Depending on the used terminal, *moni::tool* provides several possibilities to transfer measurement results and other data and information to external devices.

6.9.1 Analog Outputs

The s::can terminal can be equipped with internal analog outputs or an external analog output module can be connected to it.

When *Analog Outputs* is selected, an overview shows all configured analog output channels.



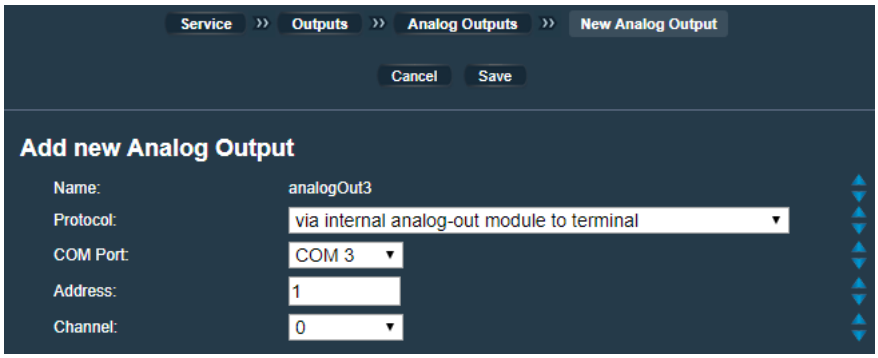
If the internal modules are not detected automatically the function *Service / Terminal / con::cube / Internal Modules / Search internal modules* (see chapter 6.4.2, p. 41) can be used.

→*Service Level / Expert*

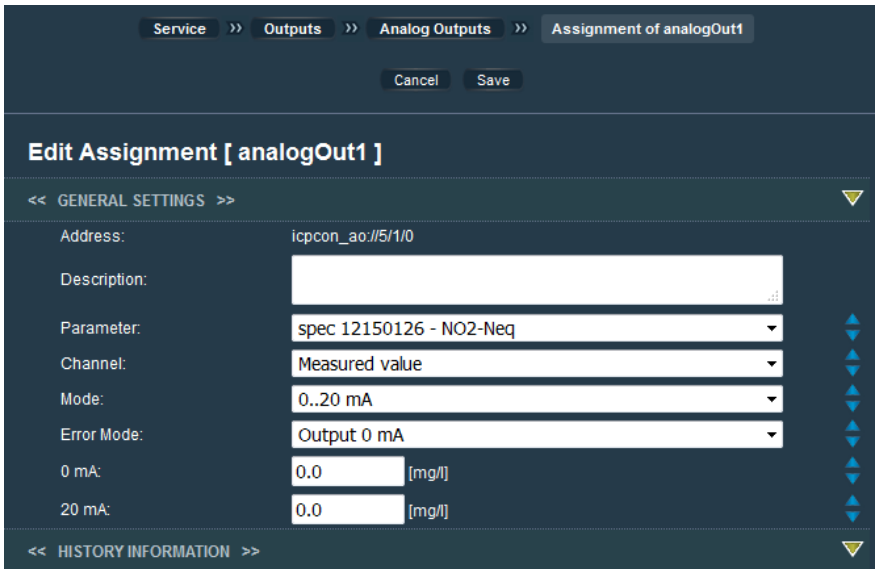
- Outputs can be added by selecting *New output*.

In the configuration window firstly the protocol (internal or external module) has to be selected.

Afterwards the COM Port, the Address and the Channel have to be entered (refer to the information provided with the external output module for the correct settings).



To each output channel a single parameter can be assigned by pushing the edit icon (blue cogwheel) on the right hand side of the respective channel. The following configuration settings must be made:



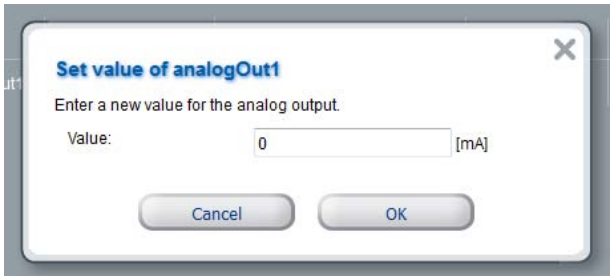
- *Parameter* defines which signal is being linked to the analog output (available parameters can be selected from a drop-down selection list);
- *Channel* defines which parameter information is linked to the analog output (e.g. measured or clean value);
- *Mode* defines if the output signal range is 0 - 20 mA or 4 - 20mA;
- Under *Error Mode* the mA signal for indicating an error (parameter value = NaN) can be selected. Possible settings are 0 mA, 3.5 mA, 4 mA, 20 mA or Hold (produces 4 mA if no valid measurement is available);
- The parameter results that correspond to an output of 0/4 mA and 20 mA can be set in the following two fields.
- All settings and modifications will be stored when pushing *Save.*

→*Service Level / Expert*

- In the field *Description* additional information can be entered
- Configuration history can be viewed

Assignments can be cancelled by marking the respective line and pushing *Delete assignment* in the *Analog Outputs* overview.

For testing the functionality of the analog data transmission a mA value can be set for each configured output by pushing *Test*.



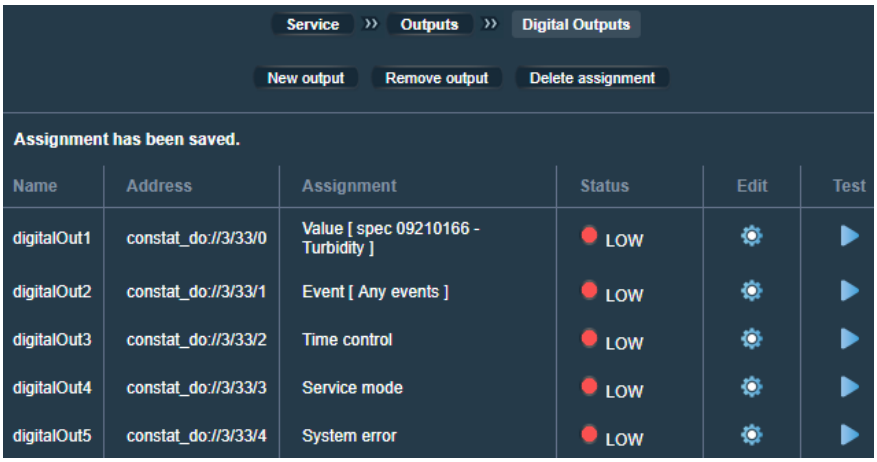
→*Service Level / Expert*

- An analog output can be deleted by selecting it in the overview and pushing the button *Delete Output*.

6.9.2 Digital Outputs

The s::can terminal can be equipped with internal digital outputs or an external digital output module can be connected to it.

When *Digital Outputs* is selected, an overview is displayed that lists all available digital output channels, together with information on their assignment and their current status.



→*Service Level / Expert*

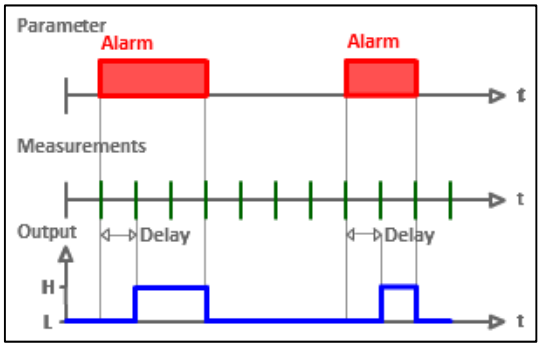
- Outputs can be added by selecting *New output*.
- In the configuration window of the new output firstly the protocol (internal or external module) has to be selected. Afterwards the *COM Port*, the *Address* and the *Channel* have to be entered (Please refer to the information of the external module for the correct device settings).

Status functions can be assigned to the output channels by pushing the *Edit* icon (blue cogwheel) on the right hand side of the channel.

To each channel one of the following functions (*Mode*) can be assigned which will be used to set the status of the digital output, which can be either *High* or *Low*.

The following modes are possible:

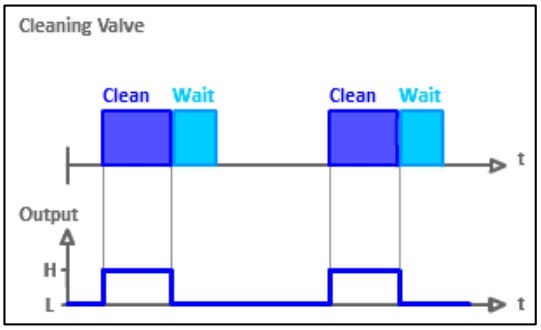
Mode	Output = LOW	Output = HIGH	*moni::tool* screen
System error	any error reported, system status = Error	Status of all devices (terminal, sensors, parameters) is Ok	*Status* tab blinking when status = LOW
Service mode	Service mode of monitoring system is active and measurements are interrupted	Service mode is not active, query of readings is running	*Service* tab blinking when status = LOW
Event	No alarms are pending	At least one alarm is pending	*Alarm* tab blinking when status = HIGH
Cleaning	Cleaning not active (i.e. cleaning valve closed)	Cleaning active (i.e. cleaning valve open)	Cleaning active indicated by *Activity*
Time control	During “interval” time	During “duration” time, 1. cycle starts whenever the service mode is left	
Value	Parameter reading within limits (within limits +/- hysteresis when it was outside limits before)	Parameter reading outside limits	



For the mode *Event* the type of the observed event can be specified. When *Any events ...* is selected all possible event types will be monitored.

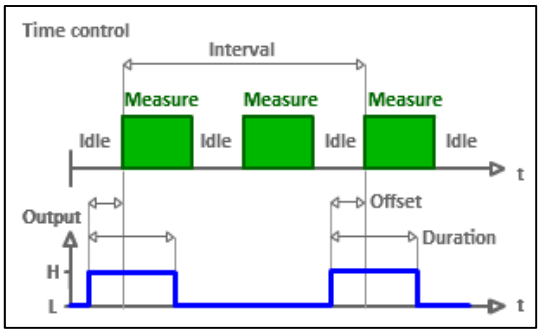
Also the number of measurements which will delay the triggering of the output has to be selected.

The logic for setting the output to high is shown on the right.



When the mode *Cleaning* is selected also the used *Cleaning device* has to be selected.

The logic for setting the output to high is shown on the right.



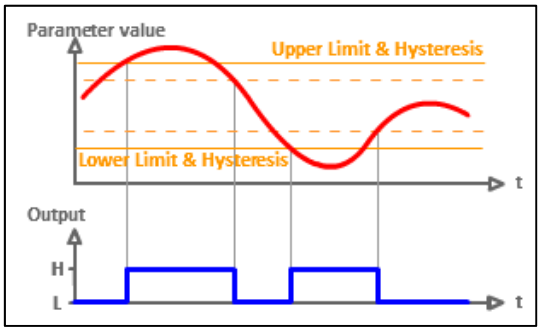
Time control is linked to the automatic measuring interval and will be configured by three parameters (refer to example on the left).

The *Interval* defines den number of measuring cycles before the output becomes active. The *Duration* defines for how long the output is HIGH. Outside this phase the status is LOW. Entering a *Start Time Offset* causes a time shift of the HIGH interval related to the start of the measurement.

The time control starts with the first measurement after saving the settings or after leaving the Service Mode.

The mode *Value* assigns the digital output to a parameter reading. If the value of the parameter transgresses a configured limit, the state of the output will change.

The following properties have to be set:



- *Parameter*: The parameter the output is assigned to;
- *Channel*: Whether the measurement result (value from sensor) or the clean result (value from *vali::tool*) will be used (the latter requires *vali::tool* and is not available for virtual parameters);
- *Upper Limit*: The value above which the output status is switched;
- *Upper Hysteresis*: The hysteresis defining when the output switches back again once the reading has surpassed the upper limit;
- *Lower Limit*: The value below which the output is switched;
- *Lower Hysteresis*: The hysteresis defining when the output switches back again once the reading has been below the lower limit.

→*Service Level / Expert*

- In the field *Description* additional information can be entered
- Under *Output Inversion* the checkbox can be used to invert the logic, i.e. instead of High the output will be Low.
- Options under *Pulse Settings* can be used to transform a constant *High* signal into a pulsed one.
- Configuration history can be viewed

All settings and modifications will be stored when pushing *Save*.

Individual assignments can be selected and cancelled in the overview by pushing *Delete assignment.* In this case the digital output will be assigned to “System error” with the next re-start.

The specifications of the assignment can be edited by pushing the blue cogwheel in the respective line. Pushing the *Test* icon switches the output status between *High* and *Low*.

→*Service Level / Expert*

- An analog output can be deleted by selecting it in the overview and pushing the button *Delete Output*.

6.9.3 Fieldbus Outputs

moni::tool supports a fieldbus slave function which allows transferring up to 32 parameter readings, one fingerprint plus status information to an external receiver.

The following protocols are supported:

- Modbus TCP(per default active);
- Modbus RTU (RS485) or Profibus or SDI-12 (per default deactivated).

Only one of these protocols is supported at a time. For using Modbus RTU on con::cube the COM5 interface is needed which then cannot be used for connecting additional sensors.

For Profibus and SDI-12 optional internal modules are needed on the con::cube. These modules (Profibus and SDI-12) support the transfer of up to 8 parameter readings.

⚠ For Profibus a special firmware for transferring up to 16 parameter readings is available.
Please get in contact with your s::can support for further information on this option.

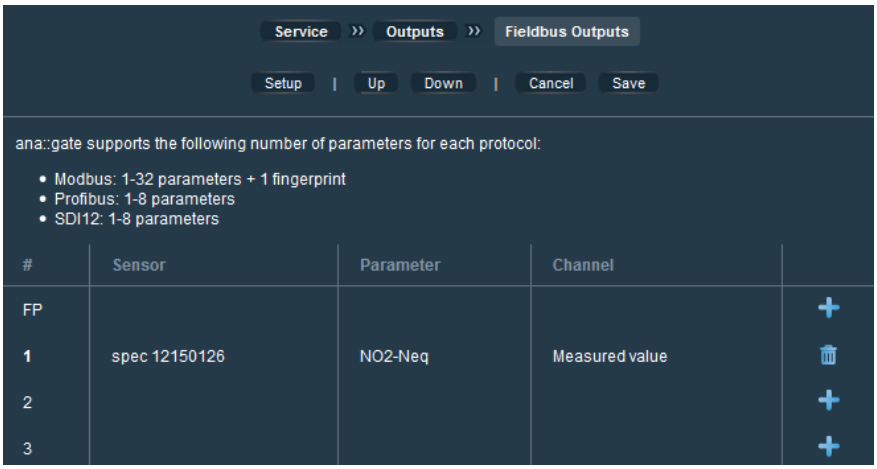
Configuring the protocol is initiated by pushing the *Setup* button.

The *Fieldbus Outputs* main window lists all available outputs.

Configuring a new assignment is initiated by pushing the blue *plus* icon which will open a dialogue window. There the parameter and the type of reading (measured or clean value) to be transferred can be selected. The process is finalized by pushing *Save*.

To change the positions of already defined parameters the buttons *Up* and *Down* can be used after selecting the parameter by pushing its name.

Pushing the trash icon on the right side of the parameter name removes a parameter from the list.



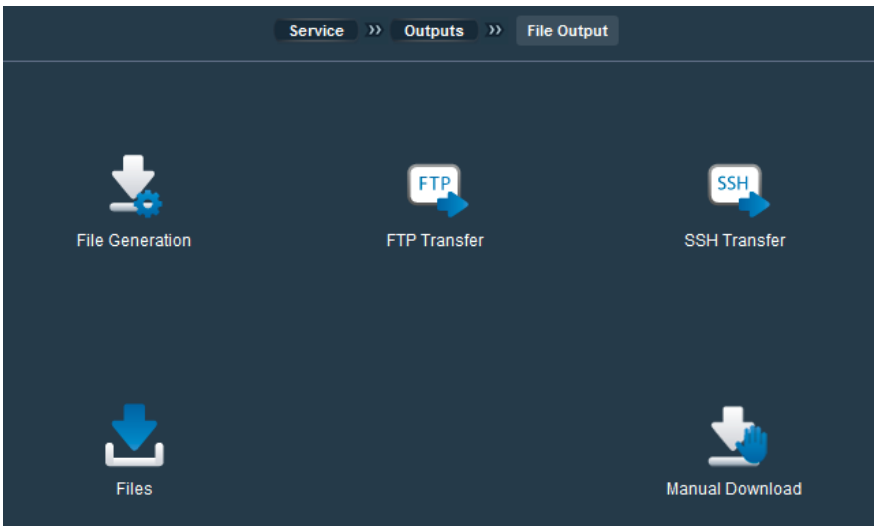
For more detailed protocol information please refer to the *con::cube* manual.

⇨ *Available **Quick reference guides**:*

Manuals	*con::cube / MODBUS*	MODBUS_Application_Protocol_V1_1A
Manuals	*con::cube / MODBUS*	MODBUS_Protocol_con-cube_BASIC_Datatypes
Manuals	*con::cube / MODBUS*	MODBUS_Protocol_con-cube_BASIC_Mapping

Please refer to the s::can Customer Portal!

6.9.4 File Output



The *File Output* function provides the possibility to configure how information stored in the *moni::tool* database is extracted to ASCII compatible files and how these files are transferred from the s::can terminal.

Information about the monitoring station and the connected sensors stored in the *moni::tool* database includes

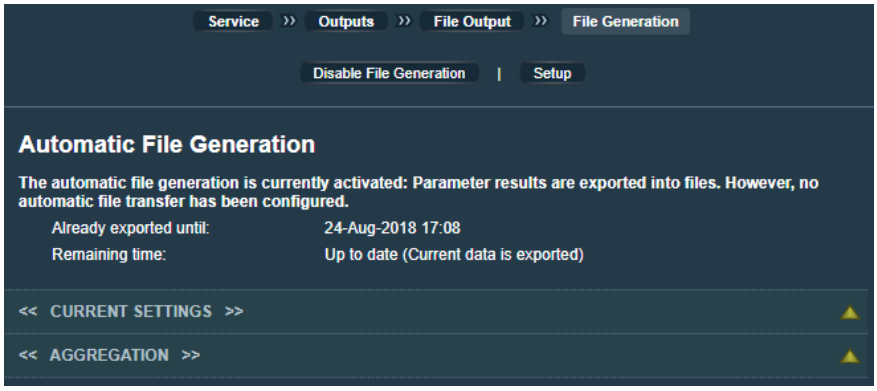
- measurement results,
- information related to sensor and station status,
- data validation results as well as
- detected events.

Furthermore, all maintenance activities that are logged by *moni::tool* are stored in this SQL-type database.

The data storage capacity of an operation terminal is of limited size and can only store a certain amount of results. The available capacity is used as circular memory in that respect means that measurements do NOT stop when the maximum size is reached but that the oldest data will be removed automatically for making space available for the new measurement results. This concerns the database as well as storage for data transfer and storage of images.

Running a typical standard configuration (1 spectrometer, 3 other sensors, 120 sec. measurement interval) the *con::cube* D-315 can be operated 4 months without data loss The current status of the circular storage is shown in *Service/Terminal/Extras/System Info*.

File Generation



Exporting data from the database can be triggered manually or automatically. In the overview window of file generation the actual status of the settings is displayed.

When *File Generation* is activated, the software will write measurement results into files automatically, hence ensuring that the results are available for immediate download and data transfer.

→*Service Level / Expert*

- See also aggregation status

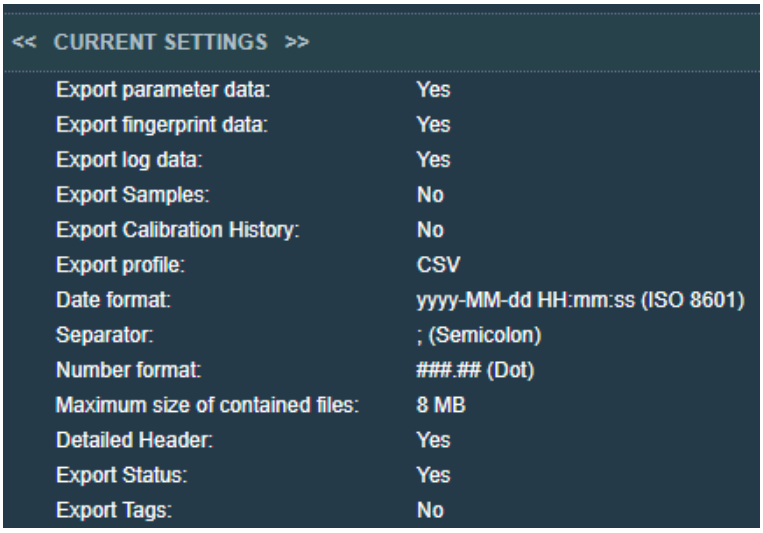
⚠ Before it is possible to change the settings using the *Setup* button, automatic file generation must be disabled. After that all existing files within the transfer folder must be removed (since all readings are still stored within the database and can be re-exported anytime using the new setup, no data will get lost by removing these files).

For transferring results by FTP or SSH automatically, *Automatic File Generation* must be activated (button *Enable Export*), which is the default setting. The automatic export function can be stopped via the button *Disable Export*.

After (downloading and) deleting the files which have been produced with the old settings, *File Generation* must be started again and *Setup* can be initiated and the available options for generating files as listed below can be configured.

Options:

- Types of data (parameter, fingerprints, logbook, samples);
- Timestamp from when the file export will start;
- Data file profile (either CSV or *ana::pro* or a customized profile);
- Date format (ISO 8601 or *ana::pro*);
- Separator between columns (tab, comma or semicolon);
- Decimal separator (Number format *dot* or *comma*);
- Maximum file size of a single file (1MB, 2MB, 8MB, Per day);
- Detailed header (includes also the calibration coefficients, i.e. a new file will be created whenever any coefficient has changed);
- Export of status data;
- Export of tag information;



Using the Aggregation options will reduce the amount of exported data compared to the number of real measurements performed.

→*Service Level / Expert*

- Function for aggregating exported data.

After saving the new setup by pushing *Save* the writing of the files needs to be started by pushing *Enable File Generation*. As all data from the selected starting point will be immediately exported the system can be busy for some time. *Remaining time* shows when the most recent data will be exported. After that new information will be checked and added to the respective files in a 60 seconds cycle.

The generated files will be stored within the transfer folder of the terminal and can be downloaded anytime manually via Service / Outputs / File Output / *Files* or automatically transferred (see following sections regarding FTP- and SSH transfer).

⚠ The structure of the files names generated automatically and the file contents depend on the used version of *moni::tool*. The differences between version 3.0 (or older) and 3.1 are pointed out below.

Filename:

The file name is created from the station name and the time stamp of the first measurement in the file as well as an extension depending on the file content. The file name extension depends on the generated data type.
Example for filename: stationname_YYYY-MM-DD_hh-mm-ss***_xx.extension***

_xx:

- ***_fp*** for files containing fingerprint results of a spectro::lyser
- ***_log*** for files containing logbook entries

In addition since *moni::tool* V3.1:

- ***_par*** for files containing parameter results
- ***_ch*** for files containing the calibration history
- ***_sample*** for files containing the sample data

Extension:

- ***csv*** with moni::tool V3.1 fix; for older versions when export profiles *CSV* or *customized* have been configured

Only *moni::tool* V3.0 or older:

- ***par/fp*** when export profile *ana::pro* has been configured

A new file will be generated each time

- an entry of the file header changes
- the maximum file size is reached (in the case of a fingerprint file reaches the maximum file size first, also a new parameter file is created at the same time)
- a re-start of *moni::tool*

Fingerprint Result Files

The fingerprint results files show in their first line the serial number (always 8 places) as well as the identifier for the length of the optical path (mm * 10) of the used *spectro::lyser*.

Optical path length	0.5 mm	1.0 mm	2.0 mm	5.0 mm	15 mm	35 mm	100 mm
Identifier in file	_5	_10	_20	_50	_150	_350	_1000

Also the sensor type (format 0xXXXX), the type name (max. 20 places), the used *Global Calibration* and the sensor name are shown.

The second line of the files state the column titles (date, time, status, wavelength) while actual readings are stored from the third line on. Measurements are recorded as absorbance values for the respective wavelengths. The description *Source* within the Status column identifies the type of the stored fingerprint (0 = measured, 1 = turbidity compensated),

12150126_350_0x1.1_spectro::lyser_UV_DRINKV170											
Measurement interval=120[sec] (Export-Aggregation disabled)	Tag	Status (Source:0)	200 nm	201 nm	202 nm	...	220 nm	221 nm	222 nm	223 nm	...
16.12.2016 17:34	*Tag-name*	Ok 0x0000.0000	NaN	NaN	NaN		31.892	31.620	31.496	31.767	
16.12.2016 17:36		Ok 0x0000.0000	NaN	NaN	NaN		31.788	31.335	31.328	31.592	
16.12.2016 17:38		Ok 0x0000.0000	NaN	NaN	NaN		31.748	31.576	31.356	31.592	

Parameter Result Files

The parameter result files show in their first line, starting at the 4. column, the name of the sensor that measured the respective parameter.

For each parameter up to four columns are reserved.

- Measured value (Header line: name of the parameter (e.g. DOCeq), unit (e.g. [mg/l]), upper and lower limit of measuring range limit and the precision of display.
- Status of the parameter and calibration coefficients (offset and slope) used for the parameter in case of a local calibration (if the parameter is not locally calibrated the values are 0.0000 and 1.0000).
- Cleaned value of the parameter as calculated by *vali::tool* (Header line same as above).
- Status of the cleaned value.

In case a parameter status is not Ok the error code will be displayed.

The values themselves are stored from the third line on, where the first column states the time of measurement.

Timestamp	testing		spec 12150126	spec 12150126	spec 12150126	spec 12150126
Measurement interval=120[sec] (Export-Aggregation disabled)	Status	Tag	DOCeq - Measured value [mg/l] (Limit:0.00-17.14)	Status [DOCeq - Measured value]	DOCeq - Clean value [mg/l] (Limit:0.00-17.14)	Status [DOCeq - Clean value]
16.12.2016 17:34	Error 0x0010		0.34	Ok 0x0000.0000.0000.0000	0.34	Ok 0x0000.0000
16.12.2016 17:36	Error 0x0010		0.34	Ok 0x0000.0000.0000.0000	0.34	Ok 0x0000.0000
16.12.2016 17:38	Error 0x0010		0.34	Ok 0x0000.0000.0000.0000	0.34	Ok 0x0000.0000

In the second column the system status at the time of measurement and in the third column an optional remark (Tag), which was created by the operator is stored.

Calibration-History Files

Information of all stored samples and of all performed calibrations are stored within these files. The first line contains the header of the columns when using *moni::tool* V3.0 or an older version. Since *moni::tool* V3.1 the first line contains the information for „filetype“, „parametername“, „sensormodel“ and „serialnumber“.

In the first four columns the time stamp, the user, the calibration type and the entered remark for the calibration are stored. The following six columns are reserved for the calibration coefficients, at which only the first two for *Offset* and *Slope* are used actually. The last four columns contain the measured value and the laboratory value for sample 1 and sample 2.

After each modification (e.g. sampling, performing a calibration) the complete history will be exported into a new file.

[filetype:calibration history]	[parametername: NH4-N]	[sensorname: ammo 13350002]	[sensormodel: ammo::lyserV2]	[serialnumber: 13350002]									
Timestamp	entered by	Type	Name	Coeff 1	Coeff 2	Coeff 3	Coeff 4	Coeff 5	Coeff 6	Raw value 1 [mV]	Lab value 1 [ppm]	Raw value 2 [mV]	Lab value 2 [ppm]
10.04.2018 14:10	Administrator	Linear	TestKal2	NaN	NaN	NaN	NaN	NaN	NaN	184,85	2,7	153,71	2
10.04.2018 11:21	Administrator	Linear	Administrator	NaN	NaN	NaN	NaN	NaN	NaN	39,93	6	-24,29	800
21.03.2018 11:15		Unbekannt		NaN	NaN	NaN	NaN	NaN	NaN	39,93	6	-24,29	800

Proben Dateien

These files contain information oft he samples taken. The first line contains the header of the columns when using *moni::tool* V3.0 or an older version. Since *moni::tool* V3.1 the first line contains the information for „filetype“, „paramername”, „sensormodel“ and „serialnumber“. Furthermore the used Global Calibration and the optical path length are stored for spectrometer probes.

In the first three columns the time stamp, the sample-ID and the sample description are stored. Then for each parameter two columns are reserved containing the value measured by the sensor and the value of the reference measurement. The fingerprints belonging to the samples are stored in a separate file.

[filetype: samples]	[sensorname: spec 11280357]	[sensormodel: spectro::lyser]	[serialnumber: 11280357]	[GC: GCHECK00V210]	[OPL: 50]	
Timestamp	Sample ID	Description	CODkhpeq [mg/l] Sample	CODkhpeq [mg/l] Labor	NO3-Neq [mg/l] Sample	NO3-Neq [mg/l] Labor
10.04.2018 13:58	1	Probe 1	0,81	110	-3,03	1,5
10.04.2018 14:00	2	Probe 2	1,21	170	-3,18	2,3

Log File

The first line contains the header of the columns (Timestamp, Log level, Log message) when using *moni::tool* V3.0 or an older version. Since *moni::tool* V3.1 the first line contains the information for „filetype“.

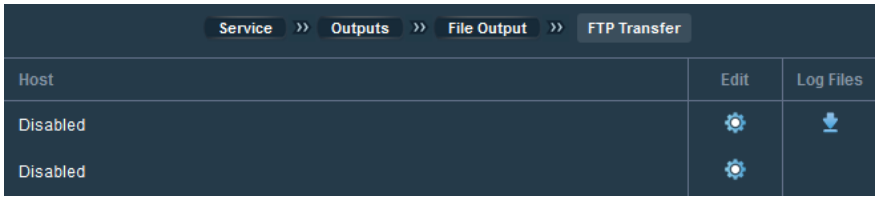
The logbook entries themselves are stored after the header of the columns.

[filetype:logbook]		
Timestamp	Log level	Log message
15.03.2018 17:50	CRITICAL	CModbusHelper::ReadHoldingRegisters(): eMBMReadHoldingRegisters() failed (url= modbus_rtu://4/5, name= scangeneric, start= 32768, n= 19, ret= MB_EIO).
15.03.2018 23:48	SYSTEM	gc.exe: Database size limit for garbage collection not yet reached (size= 30 MiBytes, limit= 900 MiBytes). Only 'VACUUM' and 'ANALYZE' executed.
16.03.2018 09:27	SYSTEM	Could not retrieve modem status
16.03.2018 09:36	USER	CSecurity::GetLock(): User "Administrator" logged in (read-write access).
16.03.2018 09:38	SYSTEM	Could not retrieve modem status
16.03.2018 09:39	SYSTEM	Measurement suspended.

Log levels are defined as follows:

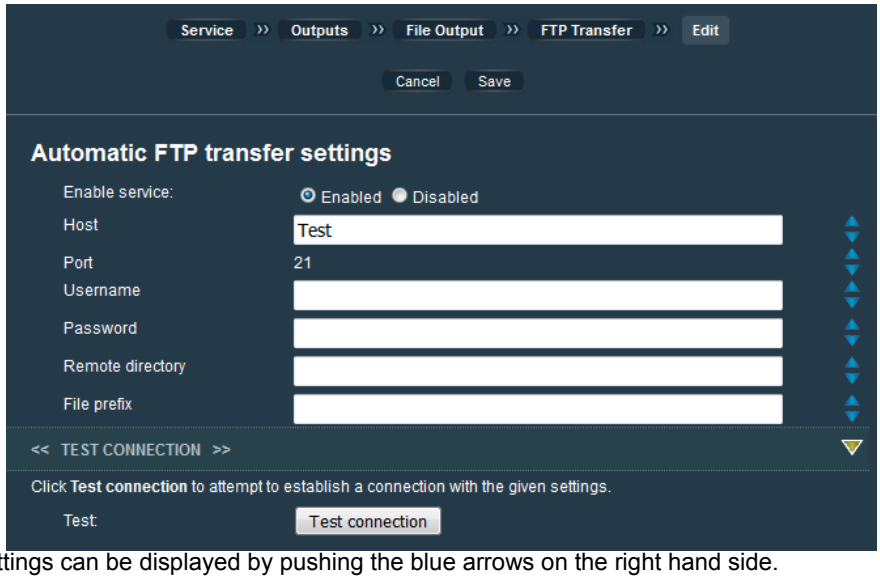
System: Information regarding the complete system (SW maintenance and integrity checks)
User: Information regarding user login and logout
Critical: Information regarding errors in measurements, configuration and security related issues
User-Interface: Information regarding the user-interface (frontend).
TML: Information regarding data-transfer using TML
Training: Information regarding training of alarm parameters in ana::tool

FTP Transfer



For using this function *Auto-Export* must be activated under *File Generation*.

This menu provides the possibility to define up to three FTP transfer processes for transferring the generated files to another computer automatically. Each of the FTP processes can be configured by pushing its *Edit* icon. The Log-Files for already active processes can be stored externally by pushing the according icon.



As a first set-up step the automatic FTP transfer function needs to be enabled for the respective host.

After that the configuration screen provides several fields to enter the required login data for the target FTP server (Host IP address, Username and Password). Also a target directory can be set.

Furthermore a file prefix can be entered to add a fixed text block to the filename. This can be used to make it possible to recognize the origin of the files when data is being transferred from multiple stations to the same FTP server.

Additional information on the individual settings can be displayed by pushing the blue arrows on the right hand side.

The name of a file transfered via FTP consists of prefix, date, time, filename and -number (if several files will be transfered; x is the number, starting with 2) as well as the original filename extension (PREFIX_DDMMYY_hhmmss_x.EXT). Since *moni::tool* V3.1 the file type is displayed before the extension (PREFIX_YYMMDD_hhmmss_x_TYPE.EXT).

If all required data have been entered correctly, the connection can be tested by pushing *Test connection*.

Pushing the button *Save* will store all settings and start the automatic FTP transfer.

Once the process has been initialized, data will be transferred after the end of each measurement if new readings are available within the transfer folder. Such new readings will be put into a new file with the current system time in the file name and which will then be transferred.

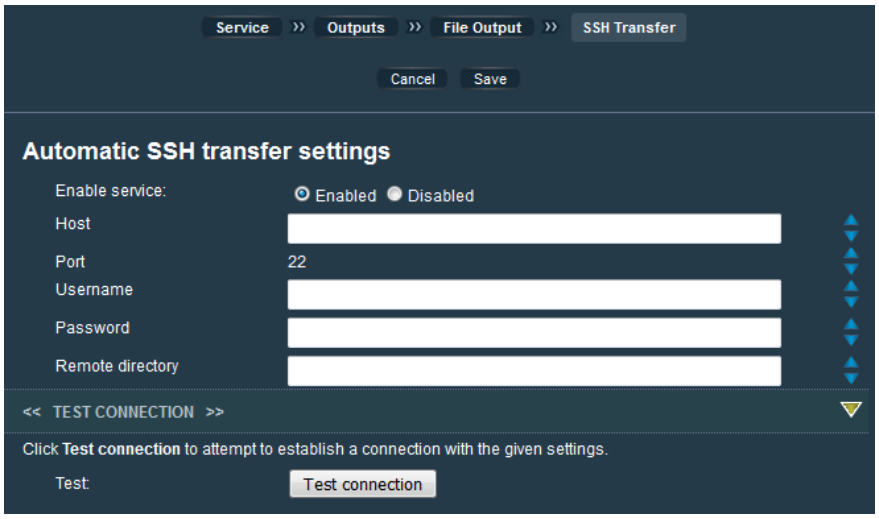
Only after a successful transfer the respective timestamp will be marked as transferred and will hence not be sent again. If *moni::tool* detects a failure during the transfer (e.g. the FTP-Server is not available) all readings which have not been transferred will be re-transmitted automatically when the FTP service is available again. Details on the file transfer process are recorded in a log-file which can be downloaded by pushing the *download* icon and then be used for analyzing the reasons of failures.

The FTP Transfer can be interrupted by editing the respective host entry and selecting *Disable* on top of the screen.

SSH Transfer

For using this function *Auto-Export* must be activated under *File Generation*.

This menu item provides the possibility to configure the SSH transfer process which will transfer result-files to another computer.



The configuration window provides several fields for entering the required data for login on the target SSH server (Host IP address, Username and Password). Furthermore a Remote directory to which the result files will be transferred can be defined. The target directory needs full right access (read, write, execute).

If all data are set up correctly, the connection can be tested by pushing the button *Test connection*. If the LED is red either host or user or password is not correct. A green LED indicates a working connection to the SSH Server.

Pushing the button *Save* will store all settings and start the automatic SSH transfer.

SSH Transfer uses the function “rsync” which will synchronize the *moni::tool* transfer folder with the target folder on the SSH Server after the end of each measurement. The function “rsync” must be available for the user on the SSH-Server. Rsync will automatically transfer any differences between these two folders so that a 1:1 copy of the *moni::tool* transfer folder is built on the server. SSH transfer can hence be used for producing an automatic backup of all *moni::tool* results on a remote server.

When the transferred data shall be removed or deleted from the server target directory, FTP transfer should be used. Otherwise SSH would continuously transfer all data from the terminal.

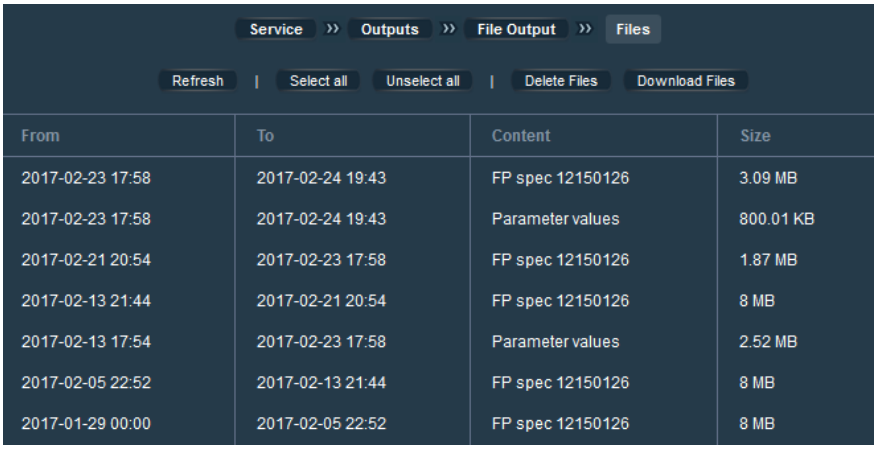
The SSH Transfer can be interrupted by editing it and selecting *Disabled* on top of the screen.

Files

In this menu all files generated automatically so far are listed. Pushing *Refresh* will actualize the display.

All (pushing *Select all*) or single files (pushing individual lines) can be selected.

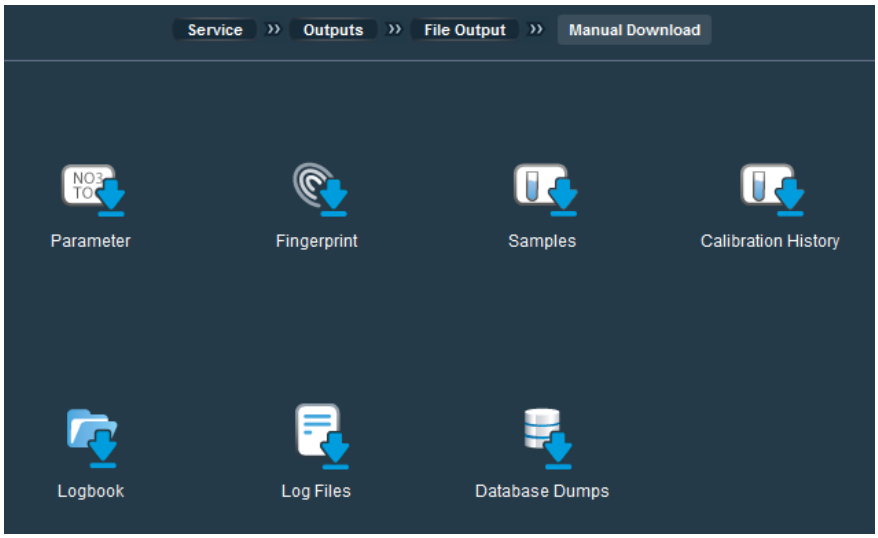
Selected files can be downloaded within one zip archive (*Download Files*) to an external medium (connected USB-memory-stick or local PC when using a remote browser) or they can be deleted (*Delete Files*).



From	To	Content	Size
2017-02-23 17:58	2017-02-24 19:43	FP spec 12150126	3.09 MB
2017-02-23 17:58	2017-02-24 19:43	Parameter values	800.01 KB
2017-02-21 20:54	2017-02-23 17:58	FP spec 12150126	1.87 MB
2017-02-13 21:44	2017-02-21 20:54	FP spec 12150126	8 MB
2017-02-13 17:54	2017-02-23 17:58	Parameter values	2.52 MB
2017-02-05 22:52	2017-02-13 21:44	FP spec 12150126	8 MB
2017-01-29 00:00	2017-02-05 22:52	FP spec 12150126	8 MB

Manual Download

As an alternative to automatic *File Generation*, this menu provides the possibility to select specific data to be exported from the *moni::tool* database into ASCII files for download on demand.



Parameter

This item can be used to export parameter results from the database to a text file on the basis of a list of criteria:

- Period (Start and End date / time using calendar icon and entry fields)
- Parameters (*Select all* button or selecting parameters individually using the tick box - also historic parameters can be selected)
- Export profile (Pull-down menu)
- Separator for columns
- Export Aggregation (Enabled / Disabled)

Fingerprint

This item can be used to export fingerprints from the database to a text file on the basis of a list of criteria:

- Period (Start and End date / time using calendar icon and entry fields)
- Parameters (*Select all* button or selecting fingerprints individually using the tick box - also historic fingerprints can be selected)
- Export profile (Pull-down menu)
- Separator for columns
- Export Aggregation (Enabled / Disabled)

Samples

This item can be used to export all sample data from the database to a text file on the basis of a list of criteria:

- Export profile (Pull-down menu)
- Separator for columns

Calibration History

This item can be used to export the calibration history of parameters from the database to a text file on the basis of a list of criteria:

- Period (Start and End date / time using calendar icon and entry fields)
- Parameters (*Select all* button or selecting parameters individually using the tick box - also historic parameters can be selected)
- Export profile (Pull-down menu)
- Separator for columns

Logbook

This item can be used to export the system logbook from the database to a text file on the basis of a list of criteria:

- Period (Start and End date / time using calendar icon and entry fields)
- Log levels (*Select all* button or selecting parameters individually using the tick box)
  - *System*: Information regarding the complete system (SW maintenance and integrity checks)
  - *User*: Information regarding user login and logout
  - *Critical*: Information regarding errors in measurements, configuration and security related issues
  - *User-Interface*: Information regarding the user-interface (frontend).
  - *TML*: Information regarding data transfer using TML
  - *Training*: Information regarding training of alarm parameters in ana::tool
- Export profile (Pull-down menu)
- Separator for columns

In all the cases described above pushing the respective *Download* button on the top of the screen will download an archive file (*_.zip*) to the web browser. Depending on the web browser setup either a file dialog will pop-up (e.g. for the *con::cube* browser) or files will be downloaded automatically. In the case of a file dialog the archive file will be compiled and transferred after selecting the target drive.

Please note that depending on the amount of selected data this process might last several minutes.

Log Files

This item can be used to download log-files which have been created by the system on specific issues automatically.

When Log Files is selected a list of available files is displayed.

Single files can be downloaded by pushing the respective *Download* icon.

Service ›› Outputs ›› File Output ›› Manual Download ›› Log Files

Type	File name	Last modified	Size	Download
FTP Log	History_1	2016-11-02 20:04	4.21 KB	
Other Log	monifront	2016-11-02 21:48	4.91 MB	
Other Log	firstboot_20160726-002932	2016-07-26 00:29	81.72 KB	
Other Log	run.log_upgrade-dbv13	2016-07-26 00:29	4.46 KB	
Other Log	ana-gate-TCP	2016-07-26 00:33	1.03 KB	
Apache Log	access	2016-07-26 00:37	0 B	
Apache Log	error	2016-07-26 00:37	0 B	

Database Dumps

This item can be used to copy the complete database content into a single file, referred to as a *dump*. The copying process is triggered by pushing the button *Dump now* on top of the screen. After a dump file has been created it is shown in the respective list.

Since *moni::tool V3.1* this function is not available anymore but replaced by the function *Backup to USB stick*.

6.9.5 TML

TML is a protocol which can be used to transfer data from the database on the s::can terminal to an external database or server. Only data requested from outside is supported, that means the s::can terminal only responds to an external request for data by sending an appropriate TML stream.

For communicating via TML port 8083 is needed. TML data can be obtained by http-requests using the GET or PUT method. The s::can terminal prepares the requested static data (system configuration) and dynamic data (measured parameters) for a specified interval defined by start and end timestamp (ISO8601). A data request has the form:

https://<station>/cgi-bin/data?start=2008-08-01T10:35:00Z&end=2008-08-01T11:35:00Z&hash=0123456789abcdef0123456789abcdefmoni

For further information please refer also to chapter 6.4.9 on security aspects of TML access and on the information provided directly in *moni::tool*!

6.9.6 Auto-Sampler

The *moni::tool* function “Auto-Sampler” allows taking samples automatically from a medium using a minimum of additional devices (just valves and bottles). All needed hardware, intelligence and the user interface will be supported by the *con::cube* running *moni::tool*. Practically, the *Auto-Sampler* function cascades a number of digital outputs, each representing the status “full” or “empty”. Consequently, only the state of the digital out-put with the status “empty” on the top of the list will be checked.

If the condition is fulfilled, a valve connected to this digital output will be powered on for the configured duration. A sample collection delay can be set to avoid getting multiple samples for the same event.

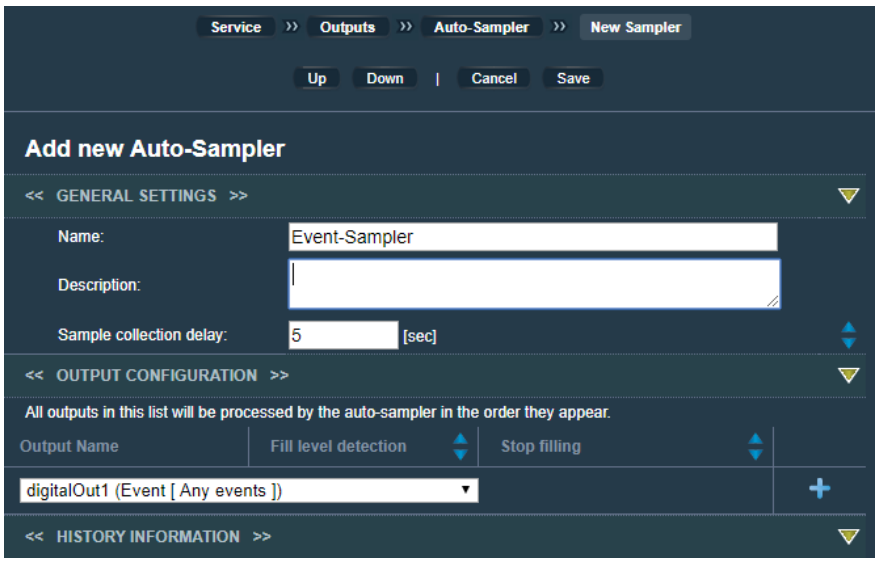
Additionally, automatic level detection can be used if the *con::cube* internal digital input module is available. In that case the filling of the bottle ends automatically when the level detector indicates that the bottle is full. Using a digital input for level detection will change a bottle’s filling status automatically from “full” to “empty” when the bottle is emptied.

⚠ Before an auto-sampler can be added it is necessary to make sure that at least one digital output assignment has been defined (see chapter 6.9.2)

When digital output(s) have been configured appropriately, the process New Auto-Sampler can be launched and some configuration settings can be made:

- Name of the auto-sampler
- Sample collection delay
- Digital outputs (“+” symbol)
- Duration for *Stop filling* or use *Fill level detection*.
  - The maximal filling duration is limited by *Sample collection delay*
  - If fill level detection is used, a digital input and its behaviour must be configured
- Pushing *Up* and *Down* can be used to bring the selected digital outputs in a correct order
- Save settings

Online help for the individual options can be displayed by pushing the blue arrows.



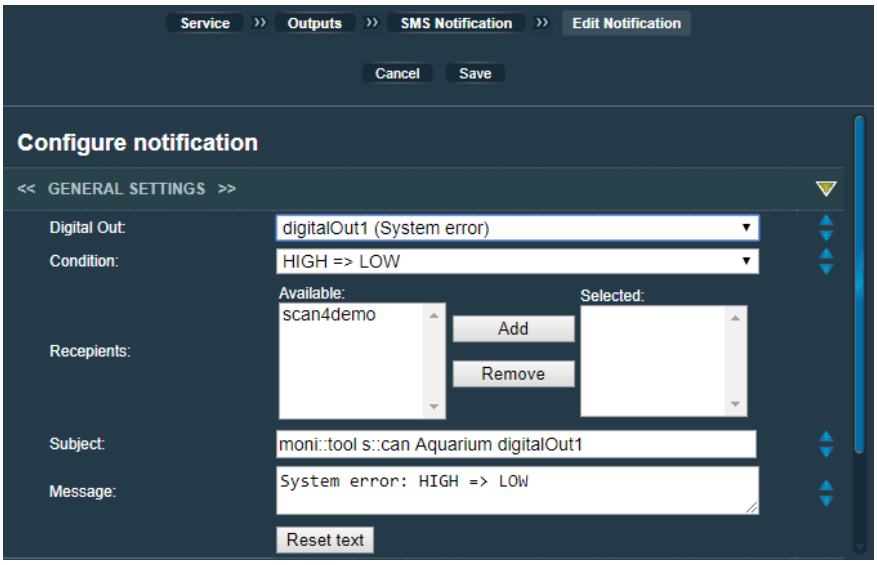
6.9.7 SMS Notification

With the help of the *SMS Notification* function a *con::cube* can use the SMS mobile service for informing about programmable conditions directly from the monitoring site in real time.

Requirements:

- *con::cube* with internal 3G modem (s::can item No. D-315-3G) running *moni::tool* V2.0 or higher
- *moni::tool* SMS notification license (s::can item No. S-11-SMS)
- SIM card with active SMS service
- User profile with configured mobile phone number

All modes which can be assigned to a digital output (see chapter 6.9.2) can be used to trigger SMS notifications. This provides a variety of possibilities for staying informed about the condition of monitoring systems. Typical examples for triggering SMS notification are parameter values exceeding a configured value, parameter alarms, system errors or activation of the *moni::tool Service Mode*.



moni::tool can also be controlled by SMS remote control using specific messages.

For details on how to configure SMS functions please refer to the respective reference guides!

*Available **Quick reference guides**:*

Manuals	*moni::tool*	note_moni-toolV2.2_SMS_Notification
Manuals	*con::cube*	note_concube_SMS_remoteControl_moniV3

Please refer to the s::can Customer Portal!

6.10 Cleaning Devices

Cleaning devices manage the automatic cleaning of sensors.

All existing cleaning devices can be configured with their individual settings. All configured cleaning processes are performed in sequence before the measurement. The process will start with the cleaning device that is displayed highest in the list. As the list is ordered alphabetically, the configured name of a cleaning device defines when it will be activated.

An example of a complete cleaning cycle is given below:

Duration Valve1 → Waiting Valve1 → Duration Valve2 → Waiting Valve2 → Measurement

The currently running process (e.g. *Waiting*) is shown as activity in the bottom menu of the *moni::tool* screen (see also chapter 0.2.5).

If the total duration of all actions as displayed above (i.e. cleaning + measurement) is longer than the measurement interval, the next cycle will not be started immediately but the beginning will be delayed until the next measuring time point. This means that fewer measurements are performed than specified by the measurement interval. The window for setting the Measurement Interval (see chapter 6.4.1) will be displayed when this is likely to occur.

New cleaning devices are installed and configured by pushing the *New cleaning device* button above the list which is available when an unassigned cleaning device address exists. This opens a window where the *Name* and the *Type* of the device can be entered.

→*Service Level / Expert*

- Add an individual *Description*
- Select the *Address* of the used valve (*constat_do://3/33/6* for *con::cube* cleaning output "Valve 1" and *constat_do://3/33/7* for *con::cube* cleaning output "Valve 2")

The configuration of already installed devices is shown in the list and can be changed by pushing the respective *Edit* icon. For configuring cleaning devices the following information needs to be provided:

Interval [sec]	Time between starts of cleaning	As cleaning is always triggered before the measurement, the minimum setting is the measurement interval. Interval 0 = device disabled
Duration [sec]	Time of active cleaning	Duration 0 = device disabled
Waiting time [sec]	Time between end of cleaning cycle and start of next process (next cleaning process or measurement)	

All settings listed above are available both for the measurement modes "Standard" and "Intensive". Different values for "Intensive" can be set for an intensified cleaning during certain conditions (see also chapter 6.4.1).

→*Service Level / Expert*

- Durations for pulsed operations (0 = pulsing disabled)
- Configuration history

All settings and modifications will be stored when pushing Save. Regarding the electrical connection of the cleaning device please refer to the s::can manual of the terminal.

Configured cleaning devices can be tested by pushing the *Test* icon in the respective row which will activate the automatic cleaning until the button *Cancel* is pushed.

Pushing *Delete cleaning device* removes the marked device which will then not be available until it is re-installed.

7 Help

⚠ This tab might have been replaced by using the Custom Screen function (see chapter 6.4.8, p. 54)

7.1 Version Info

Pushing *Version Info* will open a window showing several information.

Version:	version of the software that is installed actually
Contact:	official possibility to contact s::can
Terminal:	model and serial number of the operating terminal
LAN:	information of local network connection
WLAN:	information of local wireless network connection

As an option also information of modem and VPN connection are displayed.

This information can also be accessed directly at any time by pushing the s::can logo in the *Bottom Menu* (see chapter 0.2.6).

7.2 Software License

The Software *moni::tool* consists of basic functionalities that are available to all users and of additional features that are only available when the appropriate license has been obtained. In the License window the additional features are listed as well as the information on whether they are unlocked and until when.

It is also possible to select (button *Browse*) and upload (button *Enter license*) a new license key (file ending ".lic", you receive from s::can). Once the key has been loaded, the additional features are immediately unlocked.

⚠ License keys are specific for the serial number or MAC address of your terminal. They can be found in the License window right above the Choose License file field. If you want to acquire a new license key, which can be done via the s::can Customer Portal directly, these IDs need to be provided to your s::can sales partner.

7.3 License Agreement

Pushing *License Agreement* will open a window showing the license agreement.

7.4 New Features

Pushing *New Features* will open a window showing a list of all modifications until the currently used version of *moni::tool*. Via the *Edit* icon more detailed information about new features provided by new versions can be displayed.

7.5 Open Source License Information

Under this point the valid Open Source License Information can be viewed.

7.6 Function Index

This index lists all *moni::tool* functions, structured according to their respective tab and menu hierarchy and hence corresponds to the structure of this manual.

A search function is included allowing to find the correct position within *moni::tool*.

PART D – ADDITIONAL INFORMATION

1 Function Checks

Checking the correct function of the monitoring system is useful in situations such as

- initial startup,
- suspected monitoring system malfunction,
- modification of the monitoring system (e.g. integration of additional sensor or device) or
- changed measuring location.

Besides, a regular functional check with weekly to monthly intervals depending on the application (e.g. composition of the medium), the types of connected sensors as well as on the environmental conditions is recommended. The following table provides an overview of the activities that should be performed for checking the integrity of the individual system components as well as possible reasons / countermeasures for frequently happening situations.

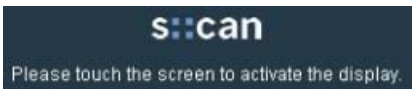
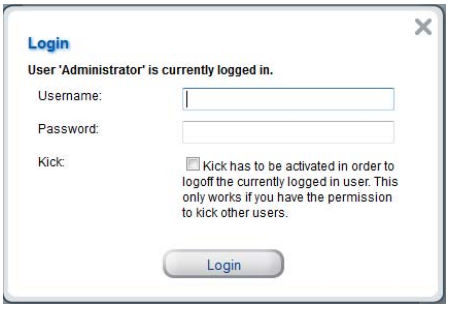
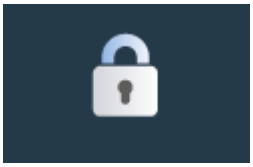
Component	Checks	Measures
Power supply	• The LED on the housing cover is on? • The *moni::tool* screen is displayed after touching the screen?	• Check power supply
Running system	• The server time and the time stamp of the last measurement displayed when pushing the clock at the center of the *Bottom Menu* are correct? • Current activity displayed on the right of the *Bottom Menu* is changing?	• Check *Measurement* setting (see chapter 6.4.1) • Check *Date & Time* setting (see chapter 0.2.4 and 6.4.6)
System status	• The *con::cube* LED is blue? • The icon of the *Status* tab is NOT blinking yellow?	• Open the *Status* tab and push the symbol of the affected component for more information (see chapter 4).
Alarm status	• The icon of the *Alarm* tab is NOT blinking yellow?	• Open the *Alarm* tab for confirming or rejecting pending alarms and for clearing all warnings (see chapter 5).
Service mode	• The icon of the *Service* tab is Not blinking yellow?	• If the *Service* icon is blinking, the measuring process has been interrupted (see chapter 6.2).
Automatic cleaning	• Air bubbles are visible when cleaning is activated? • Sound of the rotating cleaning brush is heard when cleaning is activated?	• Check the *Cleaning* settings (see chapter 6.10) and wait for next cleaning cycle • Use function *Test cleaning device* (see chapter 6.10).
Compressed air main	• No hissing sounds audible? • No uncontrolled air bubbles visible?	• Check tightness of tubes and fittings
Compressor and air storage tank	• Pressure of compressed air is correct?	• Check pressure setting • Drain condensed water from the compressor's storage tank (not necessary for s::can compressor B-32).
Monitoring by-pass	• All tubes and fittings are tight? • All sensors are supplied with medium? • No air bubbles within the tubes?	• Check valves and pumps • Check the geometry of the by-pass tubes (e.g. diameter, air traps)
In-situ installation (submersed)	• Mounting equipment of all devices is ok? • All sensors are submersed?	• Maintain and repair mounting material • Adapt installation
Data transfer	• Readings on the local terminal correspond to readings on the external system.	• Check settings for the used *Outputs* (see chapter 6.8 and 6.9)

For the plausibility of measurement results please refer to the respective information provided for the *Values* (chapter 1) and *Status* (chapter 4) tabs. In cases of doubt regarding the integrity of a sensor, please refer to the section *Sensor Integrity* in the manual of the respective instrument.

2 Troubleshooting

2.1 Known issues

The following situations have occasionally caused difficulties in the past.

	Reason	Reaction
	The screen saver has been activated.	Touching the screen with a finger or the mouse pointer will activate the display again.
	Another user is currently logged in (eventually directly on the terminal itself).	Login might be possible at a later time. If the user to be newly logged in has the appropriate right, the current user can be kicked out by ticking the respective check box and pushing *Login*.
	Parameter reading = *NaN* (“*Not a Number*”)	Use the *Status*-Tab for identifying the reason for the error in clear text. Pushing the text or the blue arrows on the right displays information on resolving the issue.
	If the issue remains:	Download the zip-file from the con::cube (*Status\Terminal\Logbook\Export all*…) and send it to your s::can Customer Support.
	Internal con::cube battery empty. Con::cube nicht mehr Verwenden um Datenverlust zu vermeiden.	Replace con::cube battery. Instruction is available on s::can Customer Portal.
	con::cube is protected against unauthorized usage by customer specific password (Web Authentification see section 6.4.9)	Connect to con::cube via Web-Browser and enter customer password.

2.2 Error and Status Messages

As has already been shown at various occasions (e.g. chapters 1.3.5, 4) *moni::tool* has the ability to monitor and document its own performance as well as that of the system's various hardware components. The results of this permanent quality control are communicated as status messages, both within *moni::tool* and in the various output files (see below and chapter 6.9.4).

Status messages basically consist of a status code which is the aggregated result of all status information available for the respective component as is shown in the example below.

Timestamp	testing		spec 12150126	spec 12150126	spec 12150126	spec 12150126
Measurement interval=120[sec] (Export-Aggregation disabled)	Status	Tag	DOCeq - Measured value [mg/l] (Limit:0.00-17.14)	Status [DOCeq - Measured value]	DOCeq - Clean value [mg/l] (Limit:0.00-17.14)	Status [DOCeq - Clean value]
16.12.2016 17:34	Error 0x0010		0.34	Ok 0x0000.0000.0000.0000	0.34	Ok 0x0000.0000
16.12.2016 17:36	Error 0x0010		0.34	Ok 0x0000.0000.0000.0000	0.34	Ok 0x0000.0000
16.12.2016 17:38	Error 0x0010		0.34	Ok 0x0000.0000.0000.0000	0.34	Ok 0x0000.0000

The specific meaning of the code format depends on the type of the respective system component as shown below.
The meaning of the different code sections within the code formats is as follows:

Symbol	Type
TTTT	System status
tttt	Terminal status
SSSS	Sensor status
PPPP	Parameter status
pppp	Parameter status (type-specific)
VVVV	*vali::tool* status
vvvv	*vali::tool* status (extension)

As documented below, status codes start with "0x", followed by dot-separated groups of four digits. The leading "0x" indicates that the following number is displayed in the hexadecimal system. Each of the following four-digits-groups represents the status of 16 positions called "bits". Each bit indicates the status of one specified condition (yes: condition is met, bit is set, value 1; no: condition is not met, bit is not set, value 0) and thus it is possible to aggregate the overall information and to show all covered aspects simultaneously. The results are sequences of 16 positions, each showing 0 or 1 for the respective condition (binary system).

However, displaying codes with up to 96 digits would not be convenient and hence the complete binary code is converted into the hexadecimal system. The result is a number consisting of groups of four digits for each covered status type because the highest hexadecimal number which can be expressed with 16 digits is FFFF. Accordingly, a system status code 0xFFFF would (theoretically) mean that all monitored system status properties are active while 0xFFFF.FFFF for a fingerprint parameter would indicate that all respective system status and all covered sensor status properties are active simultaneously.

For understanding the situation which led to a certain status code it is necessary to understand which positions have been set to 1 because the respective condition has been met. For this purpose the hexadecimal code needs to be transformed back into the corresponding binary number which can be done by scientific calculators or even by the calculator included in MS Windows.

Status type	Name in file header	Code format within file	Examples
System	*Status*	0xTTTT	*Ok 0x0000;* *Error 0x0010* (Bit 4 is set => System error)
Parameter	*"parameter name" – Measured value [unit] (Limit: LL – UL)*	0xtttt.SSSS.PPPP.pppp	*Ok 0x0000.0000.0000.0000;* *Error 0x0001.0000.0000.0000* (Bit 0 is set => no communication between sensor and terminal)
Clean value	*"parameter name" – Clean value [unit] (Limit: LL – UL)*	0xVVVV.vvvv	Ok 0x0000.0000; Error 0x0800.0000 (Bit 11 is set => maintenance necessary)
Virtual parameter	*"parameter name" – Result [unit] (Limit: LL – UL)*	0xtttt.PPPP.pppp	*Ok 0x0000.0000.0000;* *Error 0x0001.0000.0000* (Bit 0 is set => virtual parameter reports an error)
Fingerprint	*Status (Source:0)*	0xtttt.SSSS	*Ok 0x0000.0000;* *Error 0x0001.0000* (Bit 0 is set => no communication between sensor and terminal)

Example:

- "*Error 0x1801.00a0*" has been documented in the status file for "*NO3-Neq clean value"*.
- "Clean Value" has the code format V*VVV.vvvv*
- VVVV = 0x1801 (hex) = b0001 1000 0000 0001 (binary) => b12, b11 and b0 set.

In general the meaning of a status code 0000 is "OK" – no error has been detected and hence no bit has been set.

The interpretation of error codes generated by set bits as well as the consequent deeper analysis will usually be done by s::can service staff. However, for facilitating a basic understanding of the underlying process the following tables show the meaning of the individual status bits for various status types as well as the displayed user message and possible reactions.

System Status (*TTTT*)

Bit no.	System Message	Reason	Removal
b1	*Service mode is active - all measurements are stopped*	No error - service mode is entered and measurements are stopped.	Leave service mode.
b3	*ALARM*	At least one unconfirmed alarm is active.	Confirm all pending alarms.
b4	*System error*	Error - at least one internal check failed.	For more information see all additional status messages. After activating or deactivating vali::tool the effect will be visible until end of the next measurement.
b6	*Left free disk space is below ALARM limit*	Error – no free disk space available anymore - measurements are stopped.	Check and free disk space or perform database maintenance.
b7	*Left free disk space is below WARN limit*	Warning – little free disk space available – measurements will be stopped soon.	Check and free disk space or do database maintenance.
b8	*Invalid measurement interval*	The current measurement interval is too short for the current system configuration. Measurements might be skipped.	Raise the measurement interval in order to ensure continuous measurement. Reduce time used for automatic cleaning (cleaning duration or waiting time between end of cleaning and begin of measurement).

Terminal Status (tttt)

Bit no.	System Message	Reason	Removal
b0	*No communication between probe and controller*	Error - Communication between sensor and controller failed.	Check probe cable and connector. Dis- and reconnect sensor. Try the same sensor with another terminal and another sensor on the same terminal.
	Virtual parameter reports an error	Error – Virtual parameter do not work.	Check all inputs (if still available). Delete and reinstall virtual parameter.
b1	*Invalid Sensor*	Error - Serial number of sensor has changed.	Disconnect new sensor and connect the original sensor. Use “Replace” function to replace the old sensor by the new sensor. Install the new sensor (and remove old sensor).
b2	*Dependencies error*	Error – For example parameter still exists, but sensor has been removed.	See logbook for more information. Undo the last change before this failure occurred the first time. Remove parameter. Reset all settings to default (if allowed).
b3	*Manual measurement*	No error - Measurements are triggered manually.	Switch to automatic mode.
b6	*mA input signal is outside of the allowed input range*		Check the functionality of the input device. Check the selected input range.
b7	*Device update required*		Device software version (firmware) is outdated. An update has to be performed.
b9	*Feature not included*	Error - this feature needed is not included in the current license of moni::tool.	Please contact your s::can sales partner and acquire a license in order to use this feature.

Sensor Status General (SSSS)

Bit no.	*System Message*	Reason	Removal
Any other bit is set	All possible errors regarding the device status incl. the user messages, the reason for the error and notes for trouble shooting can be found in the manual of the specific sensor.	Sensor status not o.k. or the sensor is used outside the specification, for example within too hot or too cold medium or supplied with too high or too low voltage.	Check installation and perform sensor function check.
B13	*Device busy*	Measurement on sensor is still running.	Increase measurement interval or speed-up measurement duration on sensor.

Parameter Status General (PPPP)

Bit no.	*System Message*	Reason	Removal
b0	*General parameter error*	Parameter error	Check if another bit is set in addition. If no other bit is set: check installation, run sensor function check.
b1	*Parameter error, Hardware error*	Sensor/Electrode signal not OK. An electrode is missing, too old or defect.	Check the electrode (remove air bubbles), run sensor function check and/or replace electrode.
b2	*Parameter error, Configuration error*		Change the local calibration or switch back to the global calibration. Change the global calibration. Inform s::can service.
b3	*Parameter error, wrong medium*	The probe is outside of the medium or in incorrect medium, for example too high concentration.	Check water supply und check whether the probe is fully submersed. If medium OK, perform system check for further information.
b4	*Parameter error, Incorrect calibration*	Most probably something went wrong during last user calibration.	Check readings and lab values, repeat calibration. If the values are OK and repeating the calibration gives the same error: restart the probe (by disconnecting and reconnecting the probe). If the problem still persists, switch back to Global (factory) calibration and repeat local calibration.
b5	*Parameter not ready*	Parameter not activated on the sensor or sensor still warming up.	Activate parameter on probe or wait until start-up is completed.
b11	*Maintenance recommended*		Check installation and sensor condition, perform maintenance if necessary.
b12	*Marked as not trustable*	Do not use the parameter reading for calibration!	Check correlation to reference measurement.
b15	*Reading out of measuring range*	In case of parameter reading is given, this is a warning and so linearity to reference measurement maybe is not as expected anymore. In case of parameter reading is NaN, this is an error indicating not possible to generate a reading anymore.	Check whether sensor is in the medium and clean. If yes, perform functional check of the instrument and/or recalibrate sensor using samples with higher and / or lower concentrations.

The sensorspecific messages of the parameter status (pppp) are described in the according sensor manuals.

***vali::tool* Status (VVVV)**

Bit no.	System Message	Reason	Removal
b11	*Maintenance necessary*	Sensor / installation needs to be checked. Alternative: *vali::tool* setup is not adapted correctly or *vali::tool* sensitivity is too strict (too high).	Check sensor and installation according to the instruction in the manual. Perform maintenance if necessary. If sensor / installation is ok, adapt *vali::tool* setup to the application / sensor / parameter or reduce the sensitivity of *vali::tool*.

3 Using VNC for Remote Control

One of the software components available on a *con::cube* running *moni::tool* is a so called *VNC server*.

VNC stands for *Virtual Network Computing* and means a platform-independent system for transferring keyboard and mouse events as well as screen content between two computers, a *VNC server* and a *VNC client*, over a network.

The *VNC server* is the VNC program on the machine that shall share its screen and will passively allow a VNC client to take control of it and such a program is part of *con::cube*.

The *VNC client* (or *viewer*) is the program that watches, controls, and interacts with a VNC server, i.e. the client controls the server.

It is hence possible to control the *con::cube* remotely via a network or modem connection using the terminal program *VNC-Viewer* which is available for free over the internet. Once the VNC connection has been established, keyboard, mouse and monitor of the computer running the VNC client can be used for remotely operating the *con::cube*.

After starting the VNC-Viewer in the appearing dialog window the correct IP address of the *con::cube* (*Server*) which shall be controlled from remote has to be entered. Please be aware that the IP address of a *con::cube* can either be fixed or dynamically allocated.

In case of a slow connection a reduction of the color depth to minimum is recommended.

After confirming the connection details by pushing *Ok* the password for the VNC server has to be entered in the pop-up window. As a standard for the *con::cube* this is „scan“ (case-sensitive) but it might have been changed for security reasons.

For *moni::tool* V3 with OS Linux the Port 5900 has to be opened in the firewall configuration (see section 6.4.9)

After the VNC session has been successfully established the screen content of the *con::cube* will be displayed on the remote notebook / PC in an extra window. As long as the mouse pointer of this computer is located inside this window of the VNC application, all mouse-controlled actions will impact directly on the remotely controlled *con::cube*. However, any action taken outside the VNC window will still control the local computer. All available buttons and input boxes in *moni::tool* can hence be operated via the keyboard and mouse of the remote notebook / PC.

When the required activities (such as configuration settings, measurements readings) have been completed the VNC-Viewer may be closed using the button x top-right.

scan Messtechnik GmbH

Brigittagasse 22-24, 1200 Vienna, Austria
Tel.: +43 (0) 1 219 73 93 - 0
Fax: +43 (0) 1 219 73 93 - 12
office@s-can.at
www.s-can.at