EDC panel instructions

Contents

Rev	date	Change date	Remark	Edit
1	27.10.20	first version		Bold
2	22.03.21	Control point switchover Command set extension		Kuehne

3	26.05.21	OpenLoopCommand		Bold
4	30.03.22	Test programme TCPClient		Bold
5	25.04.23	Option force limit I/O Screen-shots updated PlugIn control		Kühne

Note: Subject to change in the interest of technical development and errors excepted.

Introduction

The EDCPanel software is a simple user interface for controlling testing technology equipped with EDC controllers.

The programme allows a connected axis to be moved/positioned to set up or perform simple tasks such as calibration or loads via a user interface. The individual measuring channels can be selected and displayed.

The individual components of the status display can be used to query and analyse the status of the entire system as well as detailed information on errors etc.

The system settings allow the management of different configurations, e.g. to operate systems with different controller settings.

The open communication interface based on TCP/IP makes it possible to control the software remotely in master-slave mode. The measured values as well as the status of the system and error states are transmitted. Thanks to the large selection of commands, it is possible to realise not only simple commands such as stop, hold or target approach, but also complex movement commands such as sine or movement commands with change of control.

As the interface protocol is compatible with that used within the LabMaster software, the EDC-Panel software can also be loaded as a plug-in module for LabMaster in order to be able to carry out modular multi-axis tests in LabMaster.

Main window

The view of the main window can be selected from three different variants depending on the application.

Minimum

Illustration 1

This view is intended for use of the EDCPanel as a plug-in in LabMaster and only contains the mandatory functions for manual operation of the axis. The EDCPanel is then docked directly in the main window of LabMaster (seeIllustration 3) and thus enables the operation of the axis controlled with the EDCPanel directly in LabMaster.

A measured value display for force / torque and position / angle is integrated in this view, but cannot be configured. When used as a plug-in in LabMaster, the channels transferred to LabMaster can be displayed on its measured value display in a configurable manner.

The areas of the operating elements can be shown and hidden individually, whereby the window automatically adjusts in size (see e.g.Illustration 2).

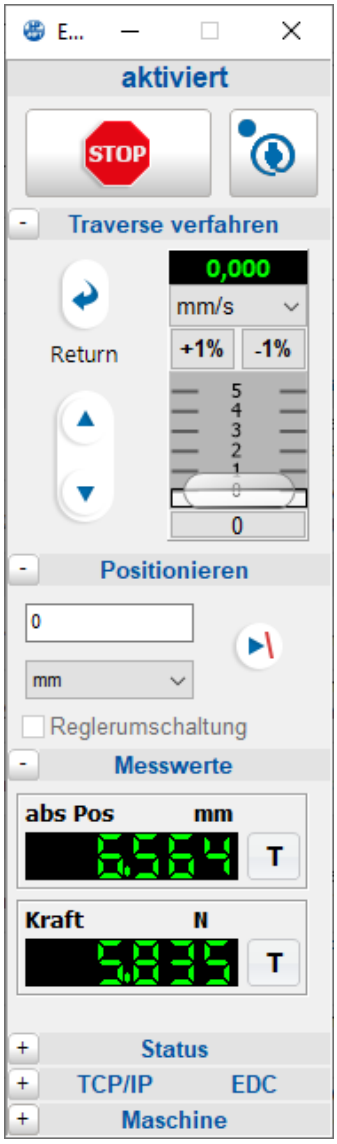


Illustration 2

The configuration used is selected in the *Machine* selection box. The buttons for *connecting / disconnecting* with the EDC or *starting / stopping* the TCP/IP server are located directly below.

The displays in the *Status* area have the following meaning:

- Direction: shows the preferred direction of the machine, measured values in this direction have a positive sign
- Controller: shows the currently active control channel of the EDC
- TCP/IP: status of the server (grey = offline; yellow = online; blue = connection established)
- EDC: Status of the EDC connection: (grey = offline; blue = online)
- Drive: Status of the machine drive (grey = not ready; blue = ready)

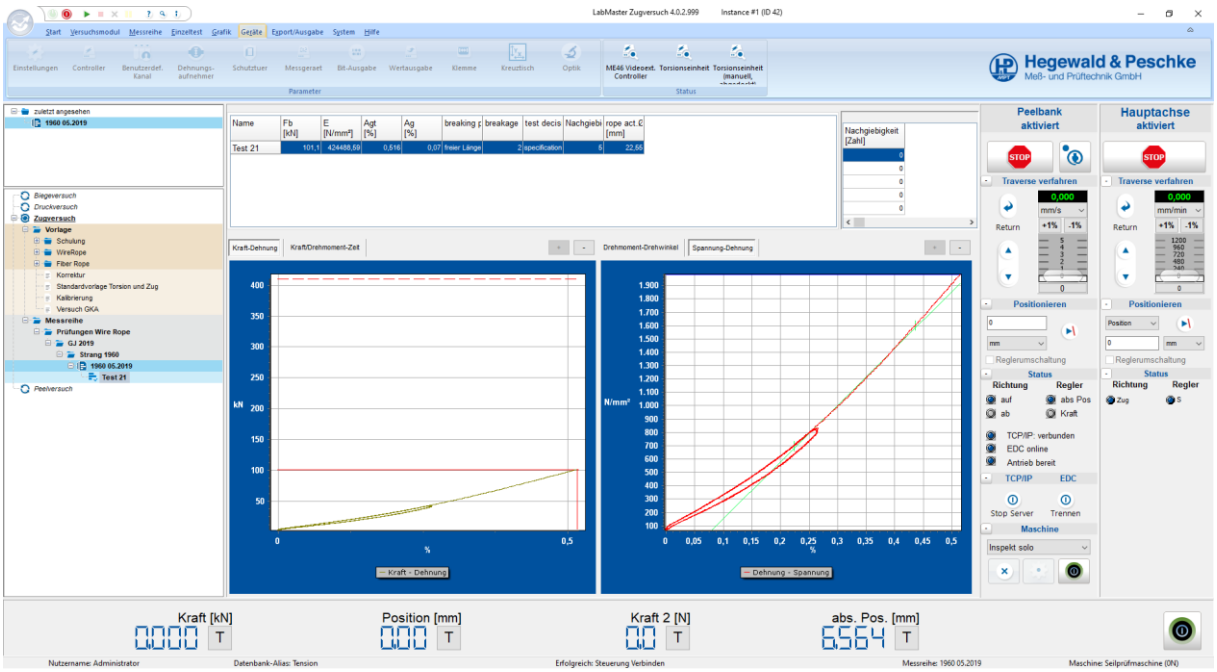


Illustration 3

The Traverse movement area contains the operating elements with which the machine can be *moved* in both directions at a speed set using the slider (*Up / Down* buttons). The selection of the speed unit is located above the slider. The desired speed can also be entered directly as a number in the field above.

The Return button is used to move the machine to the zero point of position / angle.

It also contains the buttons for *Stop*, *Drive on/off* and *System settings*.

The *Position* function allows you to move to a specified target value for position / angle or force / torque. The movement to the target takes place at the speed selected under Traverse movement using the control channel that results from the selection of the unit. If there are different control channels for movement and target, it is possible to switch the EDC to the control channel of the target when the target is reached. In this case, the target is

approached exactly. Otherwise, the target is only monitored and, depending on the speed, also overrun. The machine then remains in the control channel that was selected for the movement.

Medium

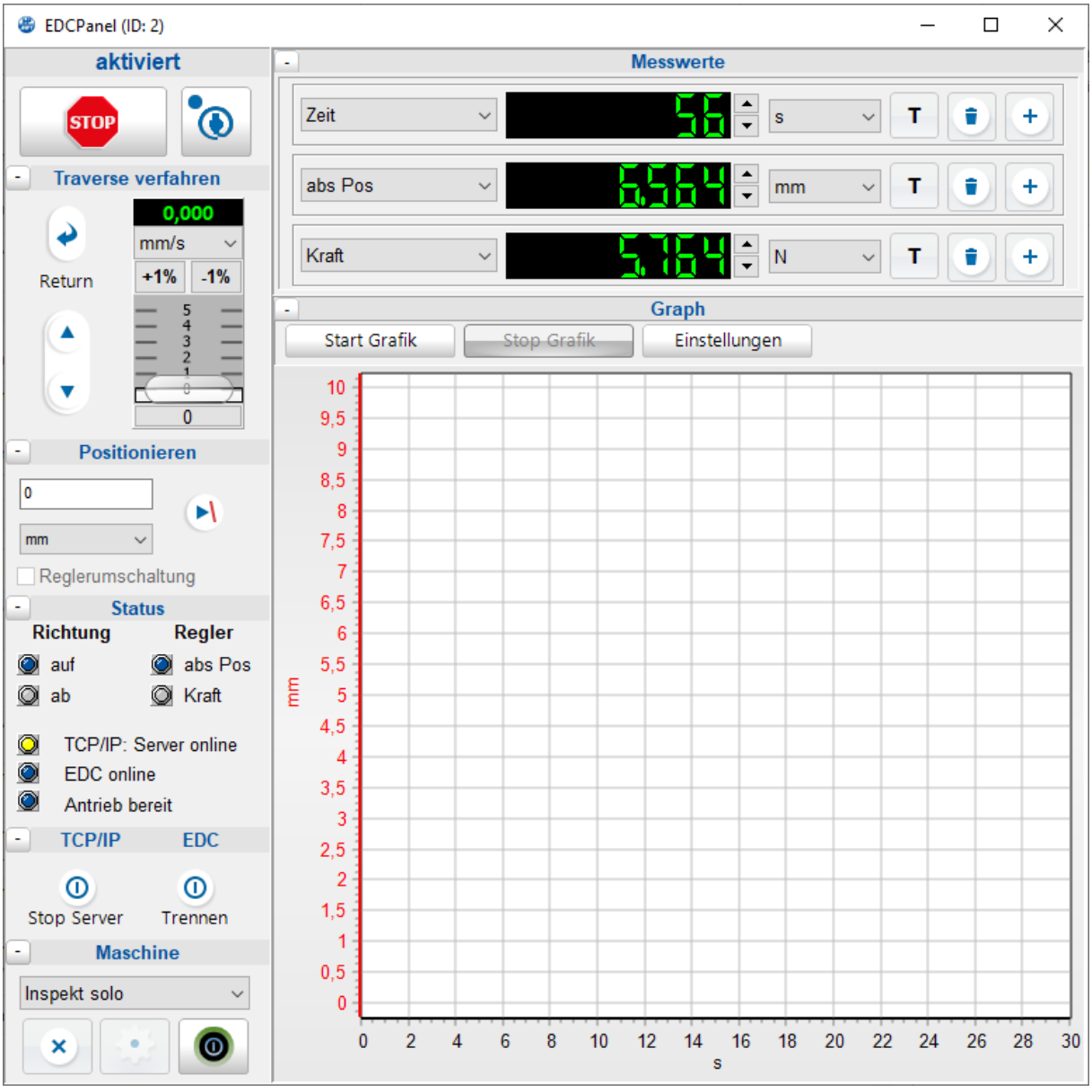


Illustration 4

This view is intended for operating the EDC panel as a stand-alone application. In addition to the basic functions that are already available in the minimum version, there is a separate area for the measured value display and a graphical representation of the measured values . The measured value display can be extended as required up to the display of all available

measuring channels. The display unit and number of decimal places can be set separately for each display. There is also a button for taring the respective channel.

The display of a live graphic can be activated in the graphics window. The axes, the measured values displayed and the time interval after which the graph is overwritten can be configured in the *graph settings*.

Maximum

The maximum variant contains additional information that is intended for developers who implement the use of the EDC panel in master-slave mode with their own software.

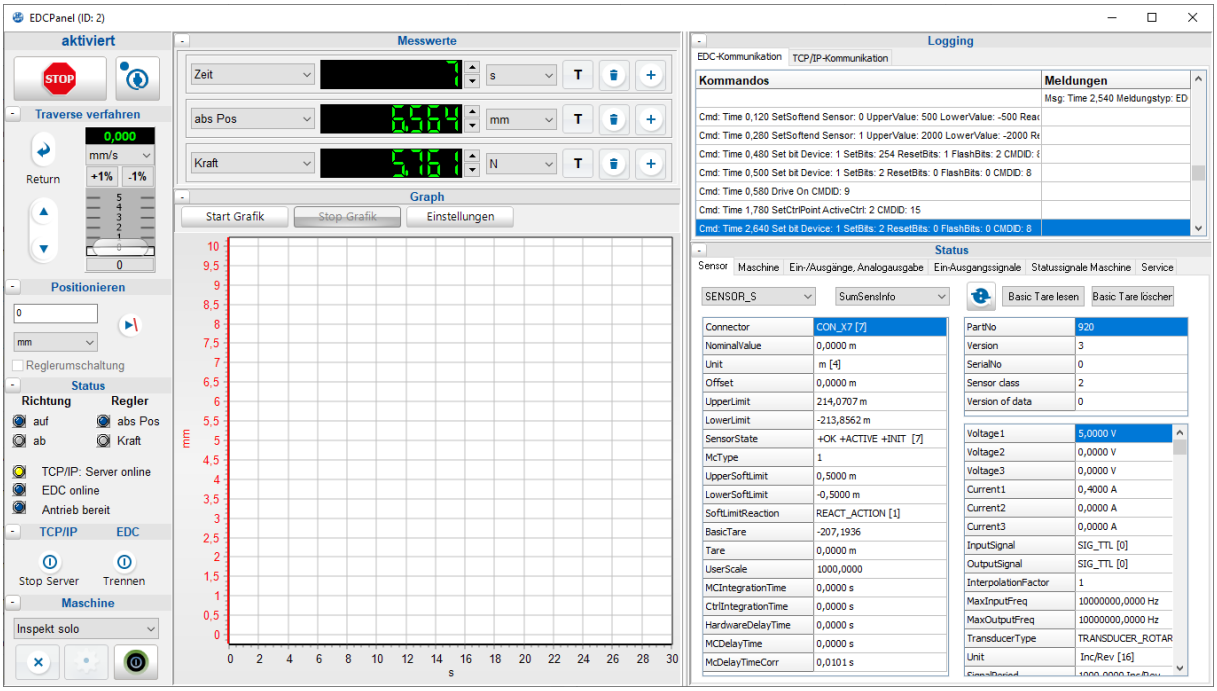


Illustration 5

In the *Logging* area, the information that can optionally be written to log files is recorded as a live display. The page is divided into two sections. In the EDC communication window, the commands sent to the EDC and their parameters as well as the EDC's response to the commands are recorded in tabular form. Communication between the master and slave is logged on the TCP/IP communication page. The *Status* area contains information from the EDC setup with which it has just been initialised.

Control points

A distinction is made between three operating points in the EDC panel:

- Internal control software
- Internal control RMC (only for EDCi)
- External control unit

You can switch between operation via the RMC manual control unit and directly via the software. The active control point after starting the software can be preset in the system settings. With EDCi series control units, a distinction is made between two operating points for the internal control.

Internal control

Internal control refers to manual operation of the EDC panel on the PC or via the RMC manual control unit. The operator has access to all operating elements of the software or the RMC. With EDCi, a distinction is made between RMC and software control. This distinction does not exist with older EDCs.

Commands from external software are not executed when the internal control unit is active. This must first request the control unit.

External control unit

The external control unit enables remote control of the EDC panel via external software. The command set described in the chapter*Interface description TCP/IP connection EDC-UDC* is available for this purpose.

The external control can be activated immediately after starting the programme by selecting it in the system settings. Alternatively, the external software can also request the control unit by command. This is only successful if the internal control has been deactivated beforehand or the *Force takeover control* option has been activated. The RMC and the operating elements provided by the software are locked for the operator when external control is active.

System settings

Any number of machine configurations can be defined and managed in the system settings. From version 2.0, the parameters of the EDC panel are saved in the following xml files:

File	Contents
Channels.xml	Available channels with their display properties and optional values for softends and controller settings separately for each machine configuration
Globals.xml	Global programme settings
GraphicSettings.xml	Settings for the graphic display
MachineSettings.xml	Parameters for the machine configurations separately for each machine configuration
Units.xml	Available units and their conversion factors
Unittypes.xml	Available types for units such as force, torque

To store the xml files, a path for the EDC panel is created under %ProgramData% when the software is installed. If no data can be found in the corresponding location when the programme is started, the EDCPanel creates new files that are filled with default values. The parameters can then be configured under System settings.

Machine

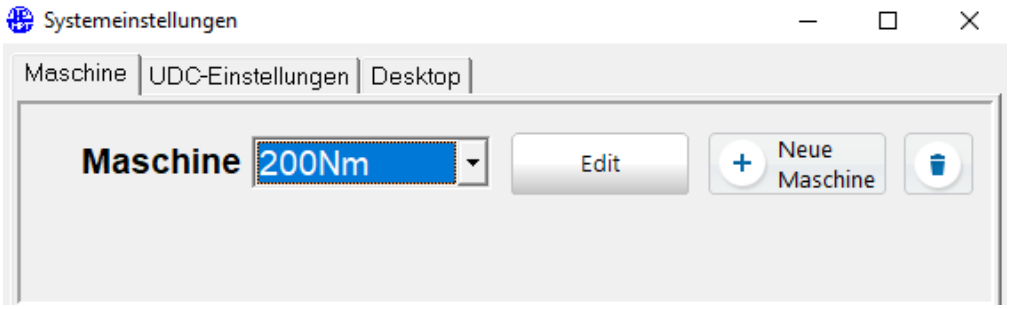


Illustration 6

On the first page, you can switch between the configurations via the *Machine* selection list. Furthermore, new configurations can be created or existing ones deleted here. The name of the configuration entered here is also shown in the header when using manual operation with display.

The settings under *Machine* are divided into several sections. The *Interface* page contains the settings for selecting the EDC and its interface to the PC as well as the selection of the

EDC configuration to be initialised (*Setup*). The *Scan Hardware button* searches all serial interfaces as well as LAN and USB ports for connected EDC controllers, determines their communication ID (*Device ID*) and reads out their setups. In the case of a USB or LAN connection, this is established to the EDC with the set device ID. If "00000000" is set for the device ID, a connection is established to the EDC control unit found first. If *hardware* was previously searched with *Scan*, a recognised EDC control unit can be selected in the list field. After the connection is established, the system checks whether the device ID of the EDC controller matches the default setting next to "USB/LAN". The setting is also effective if the connection is not via USB or LAN.

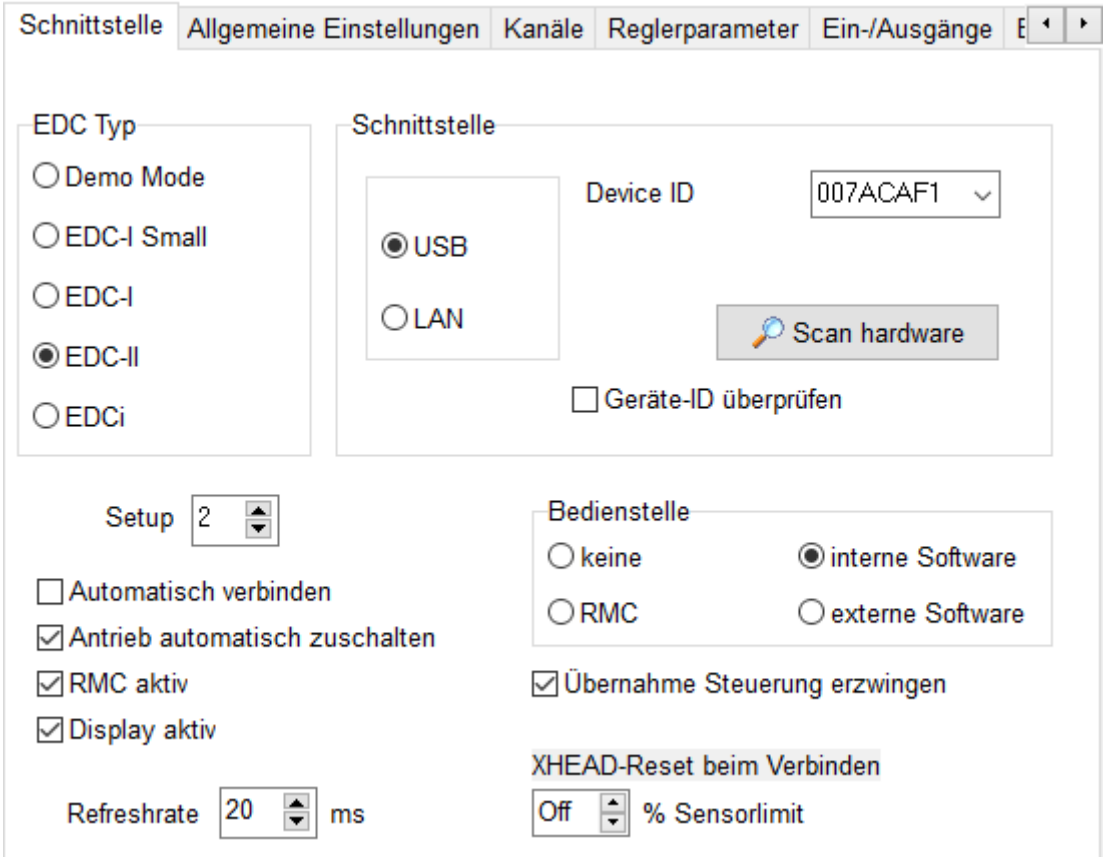


Illustration 7

Note: Observe the installation instructions for the communication drivers.

In the case of a serial connection, this is established via the specified serial port with the set baud rate.

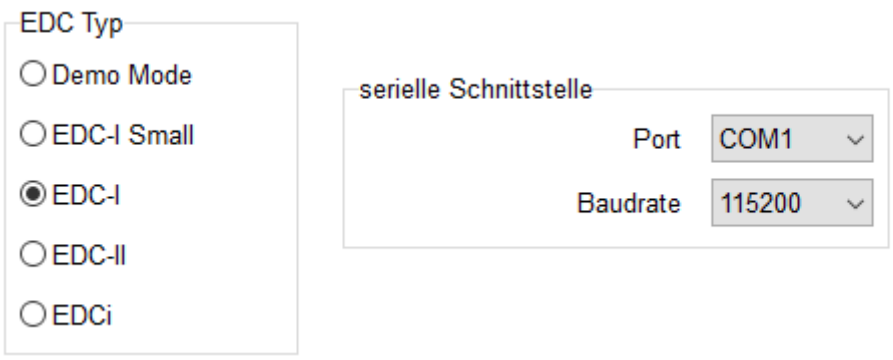


Illustration 8

The selection in the *Control centre* field determines which control centre should be active after the software is started. If the *Force takeover control* option is activated, you can switch directly from external control to internal control / RMC by command.

The *refresh rate* specifies the time interval at which the EDC should send measurement data to the PC. If a manual control with display is connected to the machine, the *Display active* option must be used. If no RMC is connected to the EDC, the *RMC active* option must be switched off.

If the *Automatically connect* option is used, an attempt is made to establish a connection to the EDC immediately after the EDC panel is started. If the *Automatically connect drive* option is activated, the drive is activated immediately after connection.

The parameters under the *general settings* have the following meaning:

- Limits for the travel speed in manual and automatic mode
- Direction with positive sign for force and position or angle and torque
- Maximum speed in jog mode
- Softends for position or angle when force hold mode is activated
- *Observe direction for softends*: this adjusts the parameters of the command for setting the softends to the setting of the positive direction
- *Invert control elements*: enables the buttons for manual operation to be adapted to the positive direction setting
- *No error on limit move command*: specifies that no error status is set when the limit of a move command is reached
- The *options for open-loop mode* allow the use of the open-loop command to move the machine. This may be necessary if the machine does not have a position

measuring system. In this case, the machine can be moved without control, whereby a percentage value of the control value is output. This can be limited in the *Max. Output power parameter*. In addition, an automatic switchover to force control can be activated when a certain force is reached.

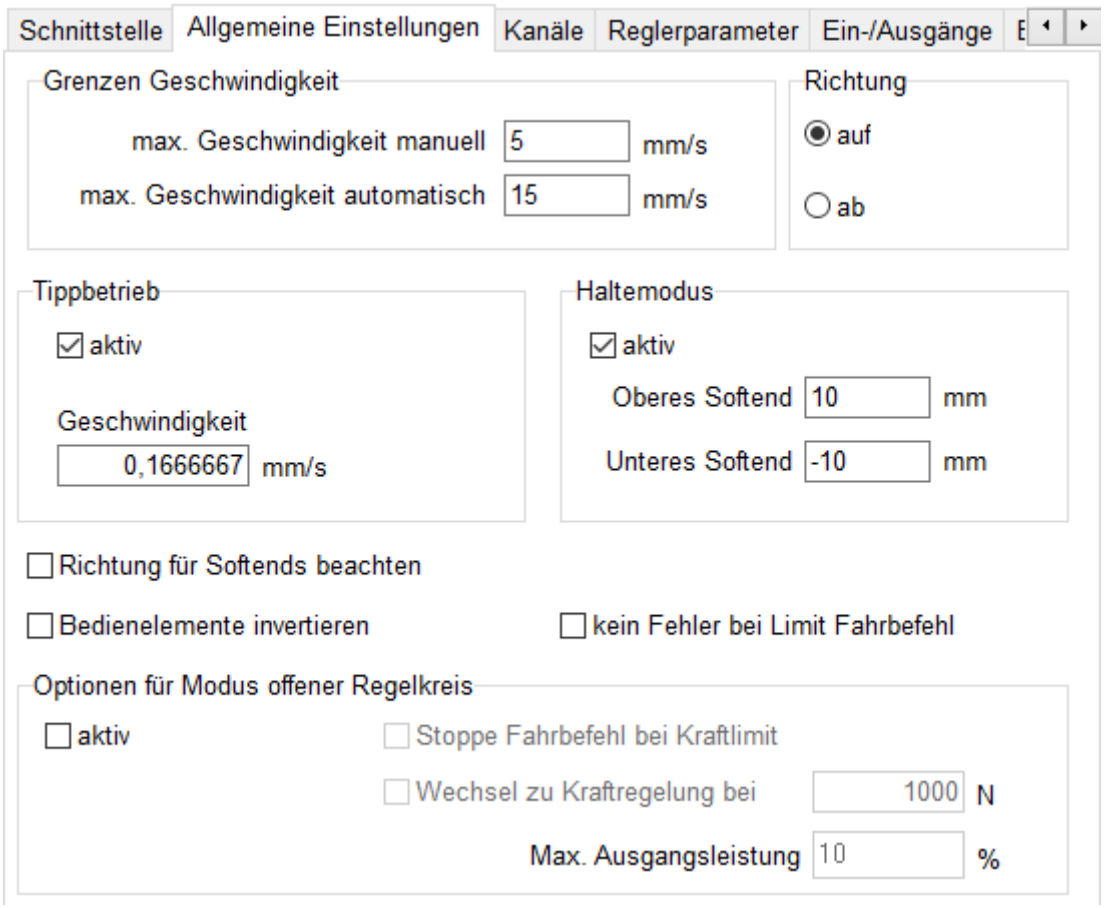
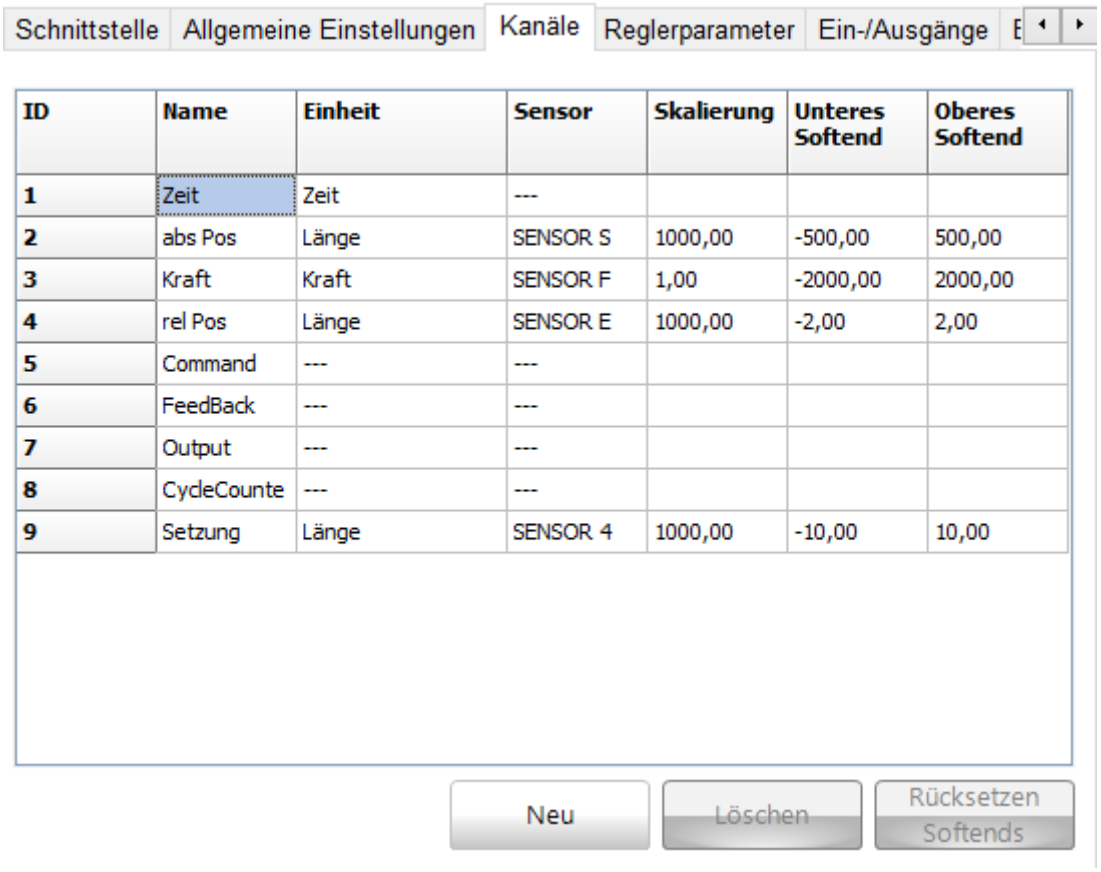


Illustration 9

On the *Channels* page, the measuring channels defined by the software are linked to the sensor settings of the EDC and configured. The *Scaling* factor is used to convert the basic unit of the sensor defined in the EDC into the basic unit of the EDC panel.

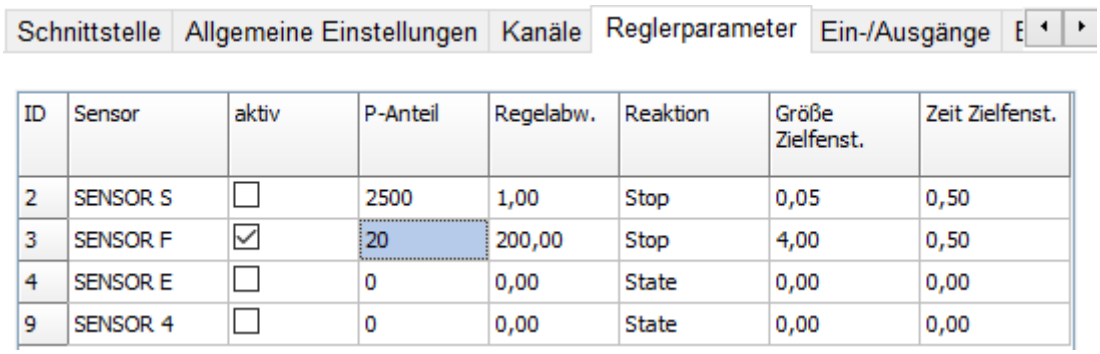


ID	Name	Einheit	Sensor	Skalierung	Unteres Softend	Oberes Softend
1	Zeit	Zeit	---			
2	abs Pos	Länge	SENSOR S	1000,00	-500,00	500,00
3	Kraft	Kraft	SENSOR F	1,00	-2000,00	2000,00
4	rel Pos	Länge	SENSOR E	1000,00	-2,00	2,00
5	Command	---	---			
6	FeedBack	---	---			
7	Output	---	---			
8	CycleCounte	---	---			
9	Setzung	Länge	SENSOR 4	1000,00	-10,00	10,00

Illustration 10

The *controller parameter* settings can be used to overwrite the values saved in the EDC setup. This option can be activated separately for each sensor. The following parameters can be overwritten:

- *P component*: Controller gain
- *Control deviation*: if this value is exceeded in the difference between the setpoint and actual value, the action set under Reaction is executed
- *Reaction*: Action when the permissible control deviation is exceeded
- *Size and time of target window*: If the EDC setpoint generator has reached the target when executing a move command, the actual value of the currently active controlled variable should have reached a range of +/- the size of the target window within the time specified here. If this is achieved, the corresponding move command is acknowledged with the message "Target reached", otherwise the message "Target window not reached" is displayed. In the latter case, however, the permissible control deviation has not been exceeded and therefore the machine is not stopped. The previously active control channel remains active with the last specified setpoint.



ID	Sensor	aktiv	P-Anteil	Regelabw.	Reaktion	Größe Zielfenst.	Zeit Zielfenst.
2	SENSOR S	☐	2500	1,00	Stop	0,05	0,50
3	SENSOR F	☑	20	200,00	Stop	4,00	0,50
4	SENSOR E	☐	0	0,00	State	0,00	0,00
9	SENSOR 4	☐	0	0,00	State	0,00	0,00

Figure 11 11

Note: Figure 11 shows the basic settings loaded from the EDC setup for all parameters. This is only the case if a connection to the EDC has already been established. However, this must be disconnected again to open the system settings. However, the software still has the last settings from the setup in its memory. Otherwise, all parameters are set to "0" and will be overwritten when activated or must be set again manually before saving.

On the *Inputs/Outputs* page, the states of the digital outputs of the EDC can be defined after initialisation or after disconnection.

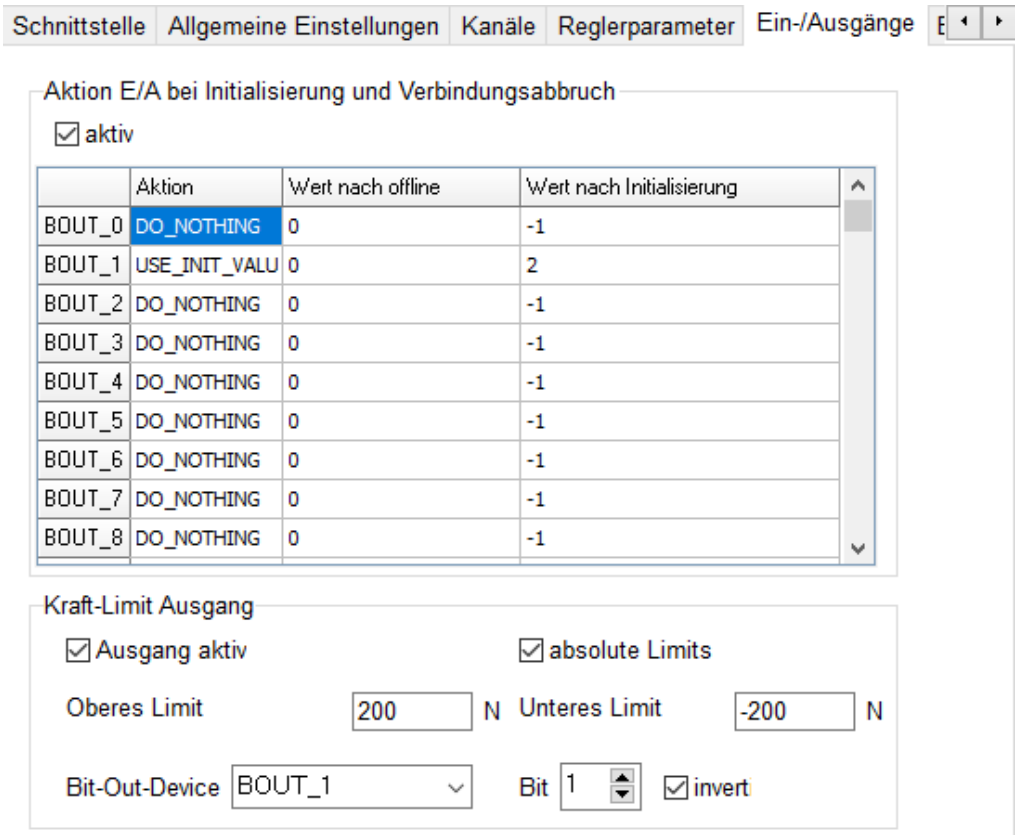


Illustration 12

The *force limit output* option offers the option of switching a digital output when a limit is reached in the force channel. This could, for example, prevent a grip from opening if the force is too high.

On the *External manual operation* page, manual control of the machine can be enabled via the RMC of a second EDC. This option is used for machines with more than one driven axis, whereby only EDCs are used. This allows the RMC of another axis to be used to flex the RMC of the EDC panel in set-up mode. This movement can be stopped automatically via an additional force limit.

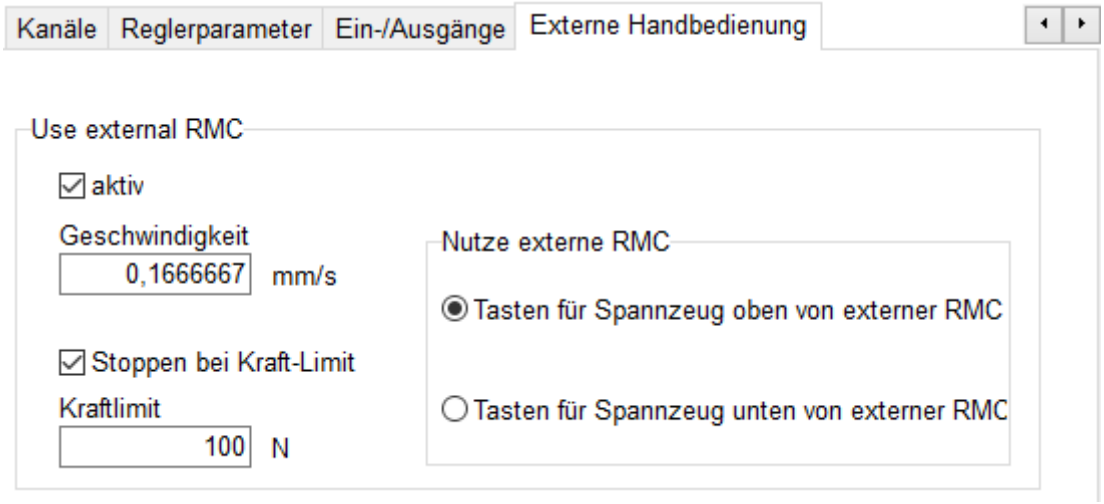


Illustration 13

UDC settings

The settings for the TCP/IP connection for master-slave operation are defined on this page. The EDCPanel acts as a server and waits for a connection request from a client.

Name	Meaning
TCP/IP port	Communication port
Execute stop if connection is terminated	The axis is stopped if the connection to the client is terminated
Start server automatically	Sets the TCP/IP server to standby mode immediately after programme start
Watch dog communication activated	

Identifiers for message start and end	Enables the client software to define the identifiers for the start and end of messages
Channels	Determines which measured values are transmitted to the client application in the data packet

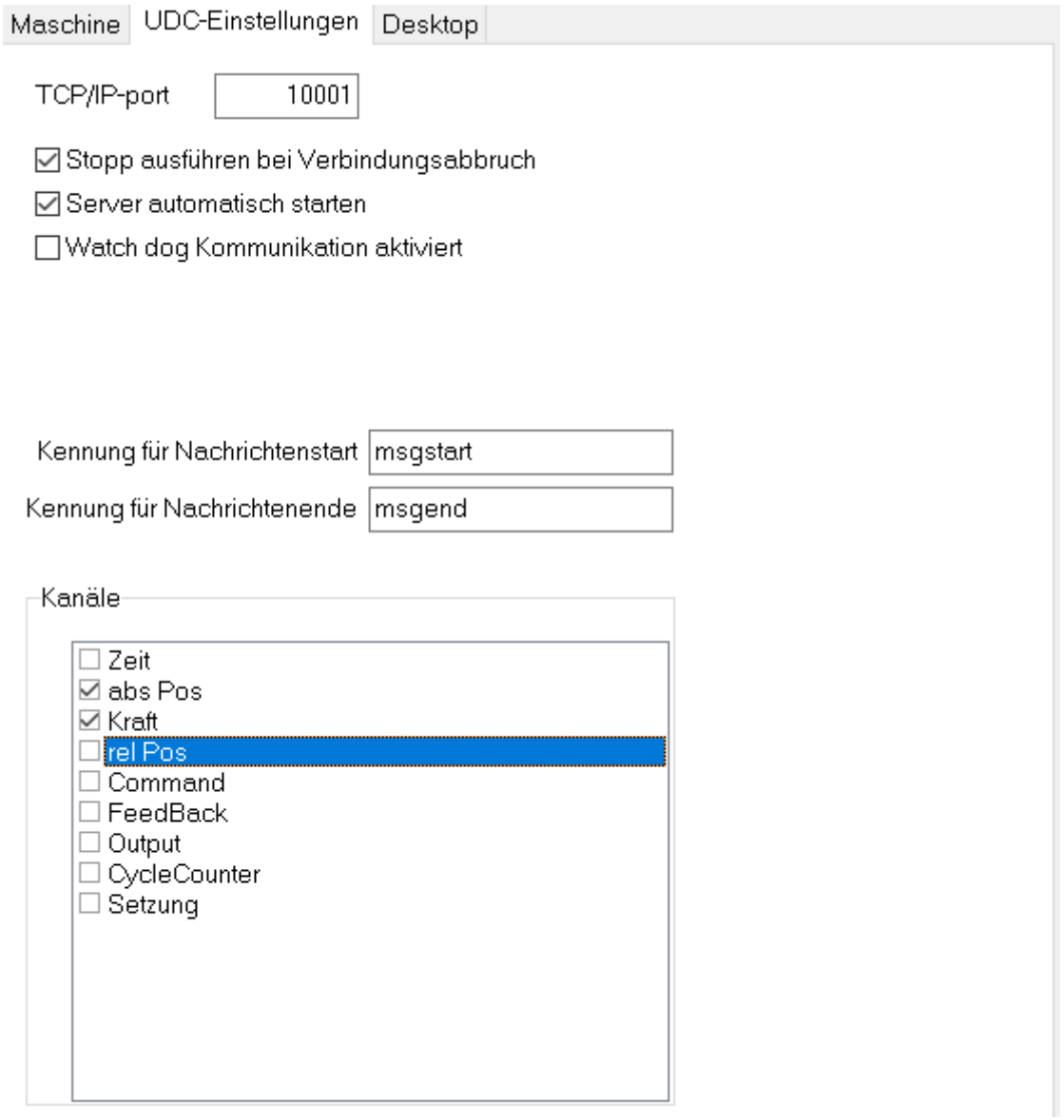


Illustration 14

Desktop

The view for the EDCPanel is defined on the last page. If the EDCPanel is to be embedded in another application (e.g. LabMaster) in master-slave mode, a plug-in ID and the messages for docking, undocking and closing the programme can be defined here. Regardless of the

desktop view variant selected (seeMain window), the EDC panel is automatically switched to the Minimum view when docking to the master application as a plugin. When undocking, the view set here is restored. The EDCPanel's internal measured value display can optionally be hidden when used as a plugin, e.g. if the transmitted measured values are also displayed in the master application.

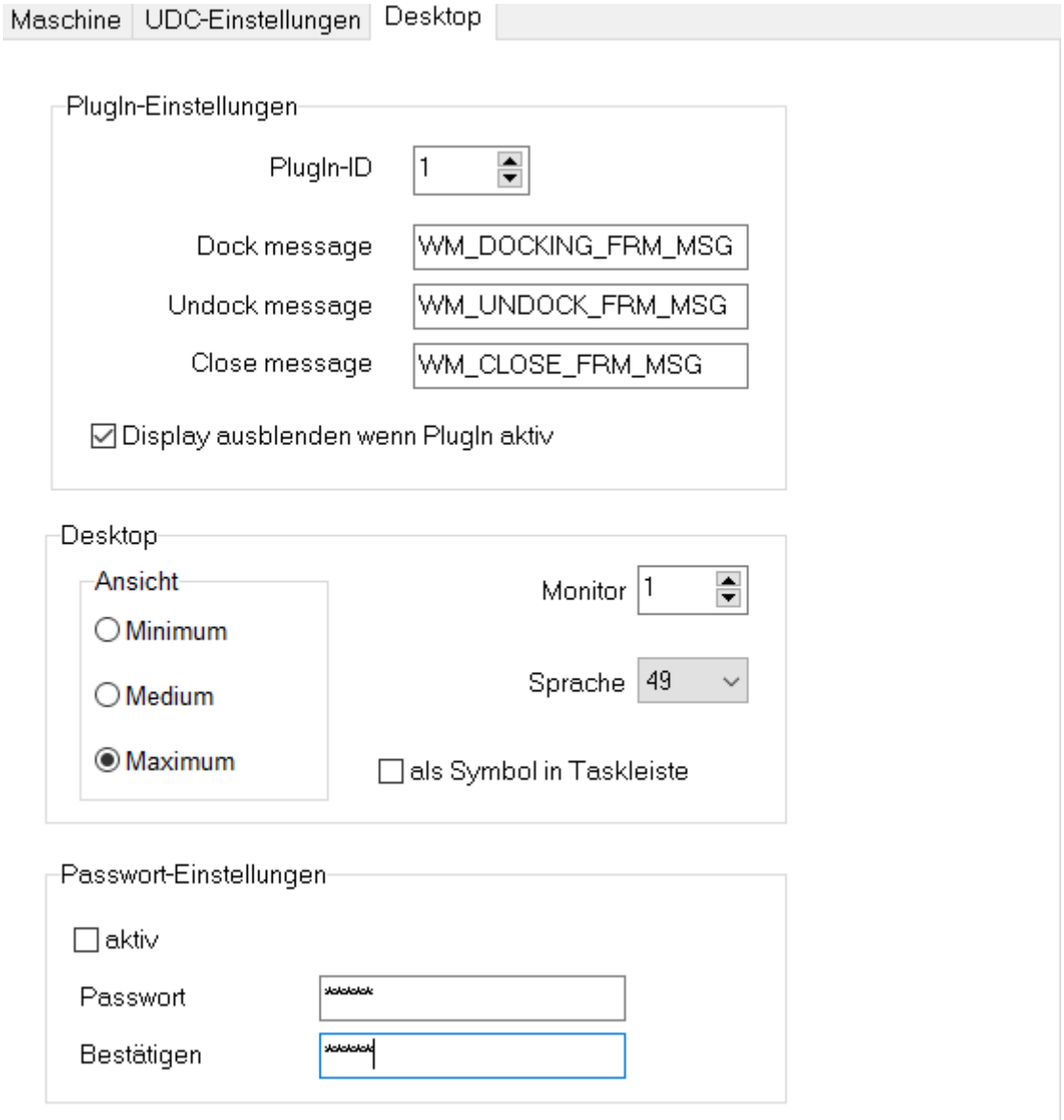


Illustration 15

The EDCPanel is currently available in German and English. The selection is made via the respective country code. In addition, the start screen can be defined when using multiple screens. If the EDCPanel is only to operate in the background in master-slave mode, it can be automatically minimised to the taskbar. Restoring the main window can be secured with a password. This password is also requested when the system settings are called up.

Logging

Various log files can be configured via the Logging button in the System settings dialogue. A log level can be set for each one, which determines how detailed events in the programme should be logged. The following options are available:

- hpllNothing: Logging not activated
- hpllErrors: only error messages
- hpllMessages: all messages including error messages
- hpllDebugging: like hpllMessages with some additional information
- hpllFunctions: all executed functions

Settings can also be made to limit the size of the log files and to organise the storage of older files.

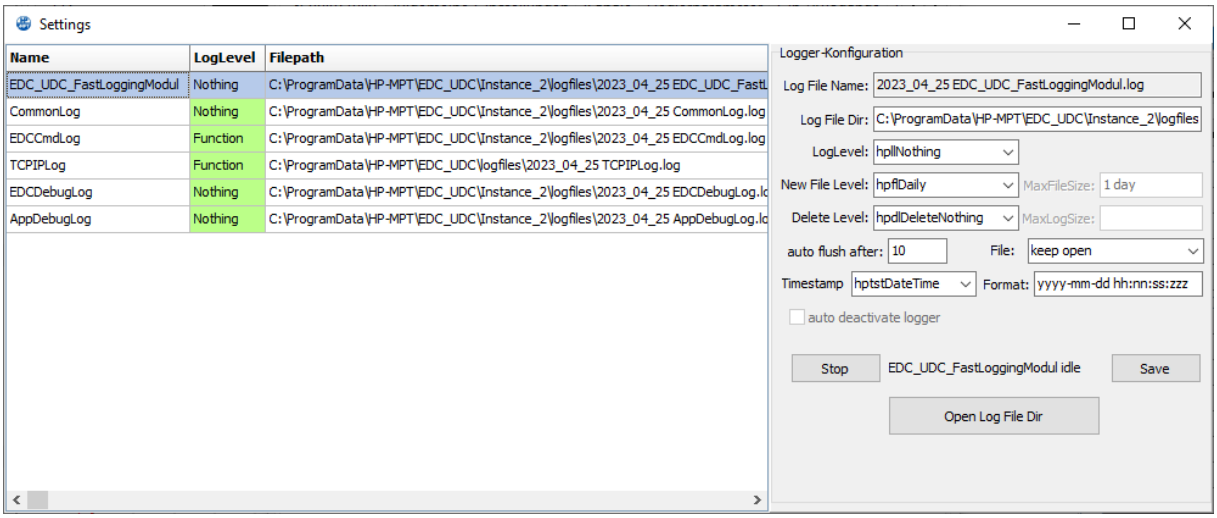


Illustration 16

The following log files are created:

- CommonLog: general information
- EDCCmdLog: Communication with EDC (commands and messages)
- TCPIPLog: TCPIP communication with master application
- EDCDbgLog: DebugInfo from EDC
- AppDebugLog: DebugInfo of the application

The communication logs for EDC and TCP/IP are recorded as a live display regardless of the settings of the two log files when the main window view is maximised.

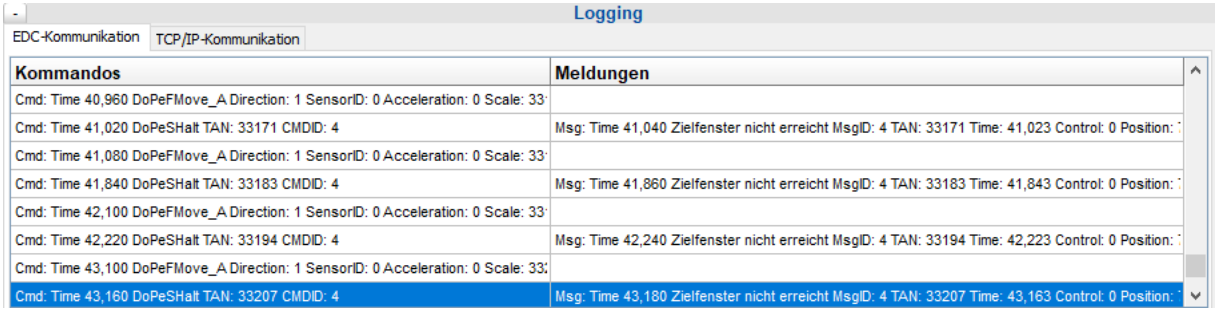


Illustration 17

Graphic settings

The graphics window has four axes that can be displayed individually. The display unit and scaling can be defined for each axis in the Axis settings section. With automatic scaling, this adapts to the currently displayed measured values so that they fit exactly into the window. With the *inverted* option, the direction of the axis is rotated.

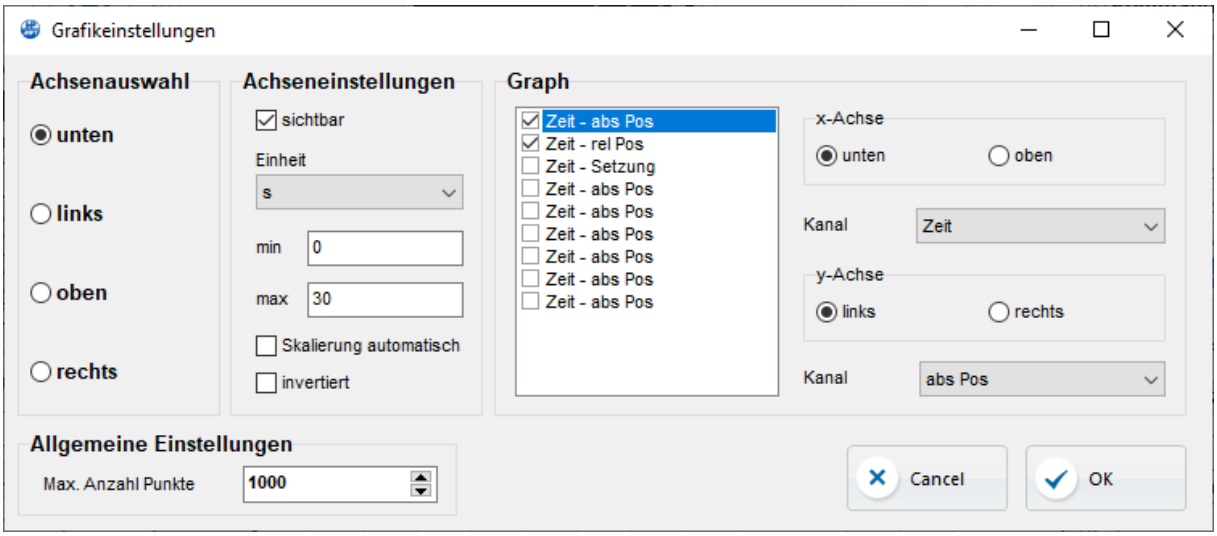


Illustration 18

The Graph section is used to specify which measurement data is to be displayed graphically, whereby a maximum of 9 curves can be displayed simultaneously. For each curve, you must specify which of the vertical axes is to be used as the Y axis and which of the horizontal axes is to be used as the X axis. Depending on the display unit set for the selected axis, the appropriate channels are offered for selection.

The graphical display uses a ring buffer, the size of which is specified via the *Max. Number of points parameter*. This shows a section of the incoming data, which represents a certain period of time depending on the size of the buffer and the set refresh time (see*System*

settings). For example, a refresh time of 20msec and a buffer size of 1000 measured value data records results in a time window of 20 seconds.

The graph display is started and stopped in the main window using the Start graph and Stop graph buttons.

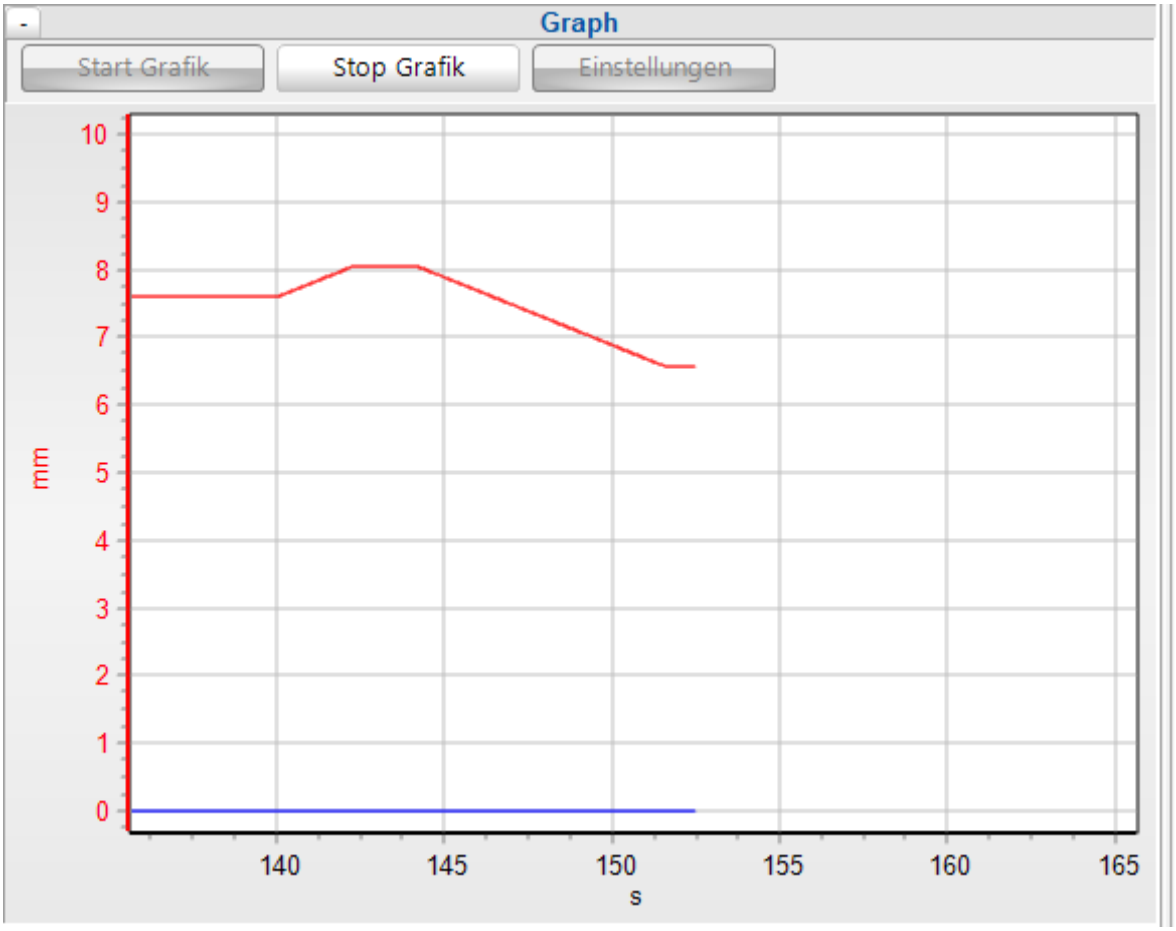


Illustration 19

Status displays

In the maximum view of the main window, the data of the setup initialised in the EDC is displayed in the *Status* area. The data of all connected sensors is displayed on the first page. The buttons *Read basic tare* and *Delete basic tare* can be used to read out the tare value stored in the control unit or set it to zero. A Refresh button is used to update the data display.

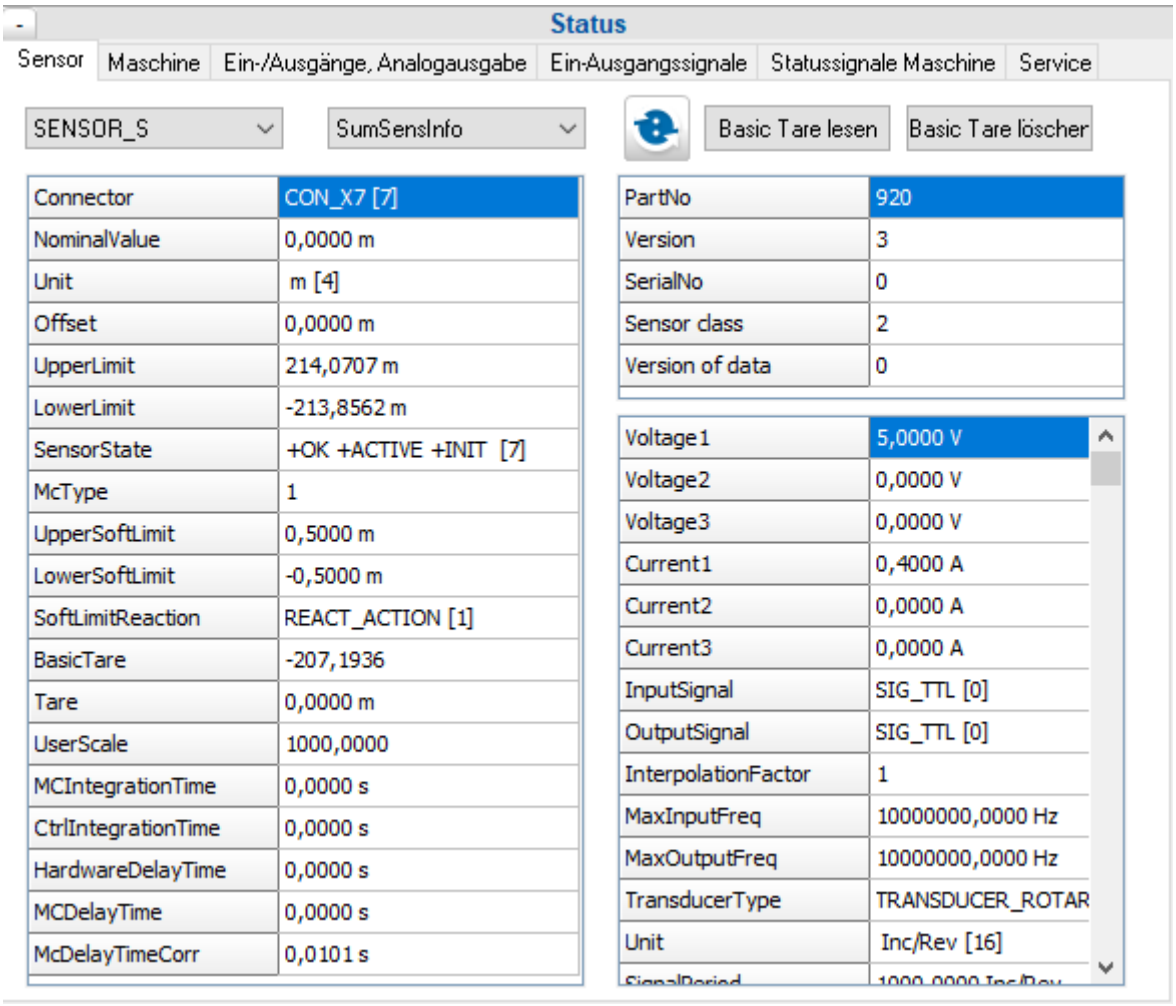


Illustration 20

The sensor information contains different views for each sensor.

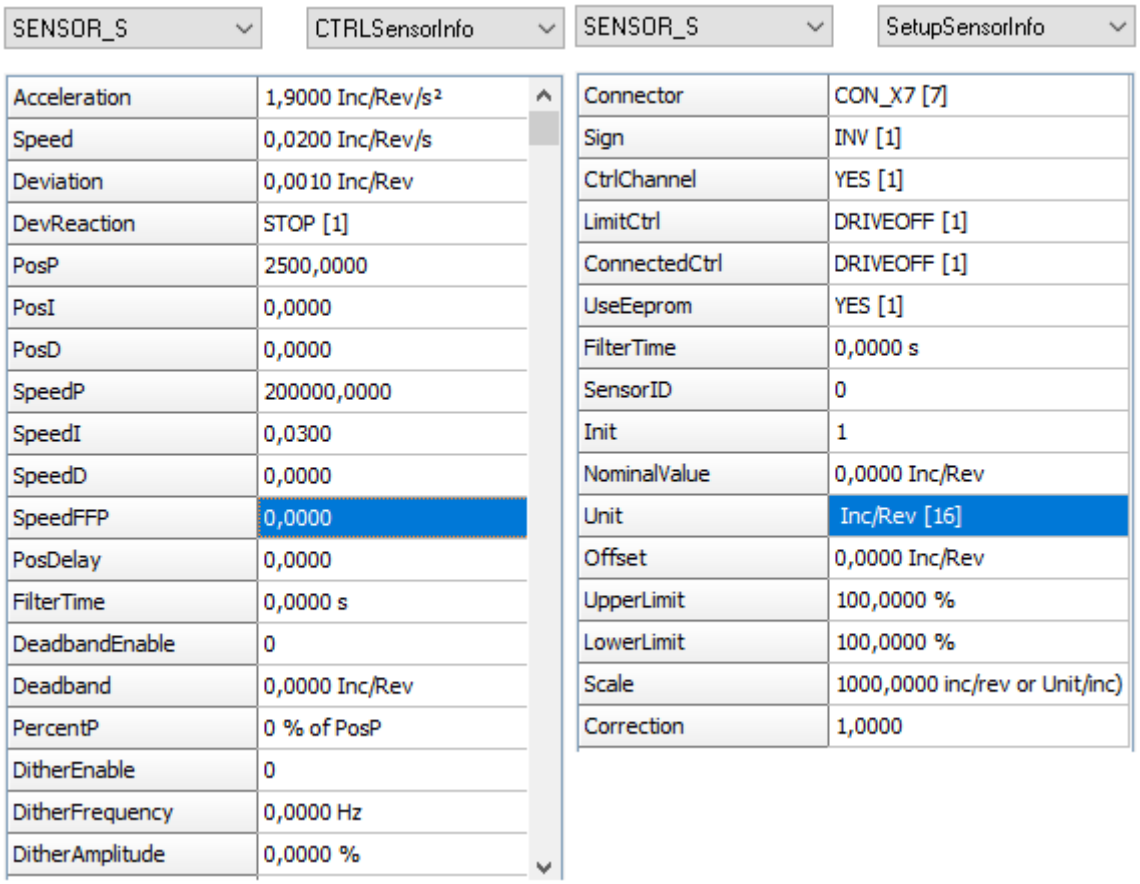


Illustration 21

On the *Machine* page, the general machine parameters and information on the version of the control software from the setup are displayed.

The *Reset traverse position* button can be used to reset the counter of the incremental distance measuring system to zero. This may be necessary for torsion machines, for example.

If an analogue output of the EDC is to be used to output a measured value, this function can be tested here in the *voltage output* area.

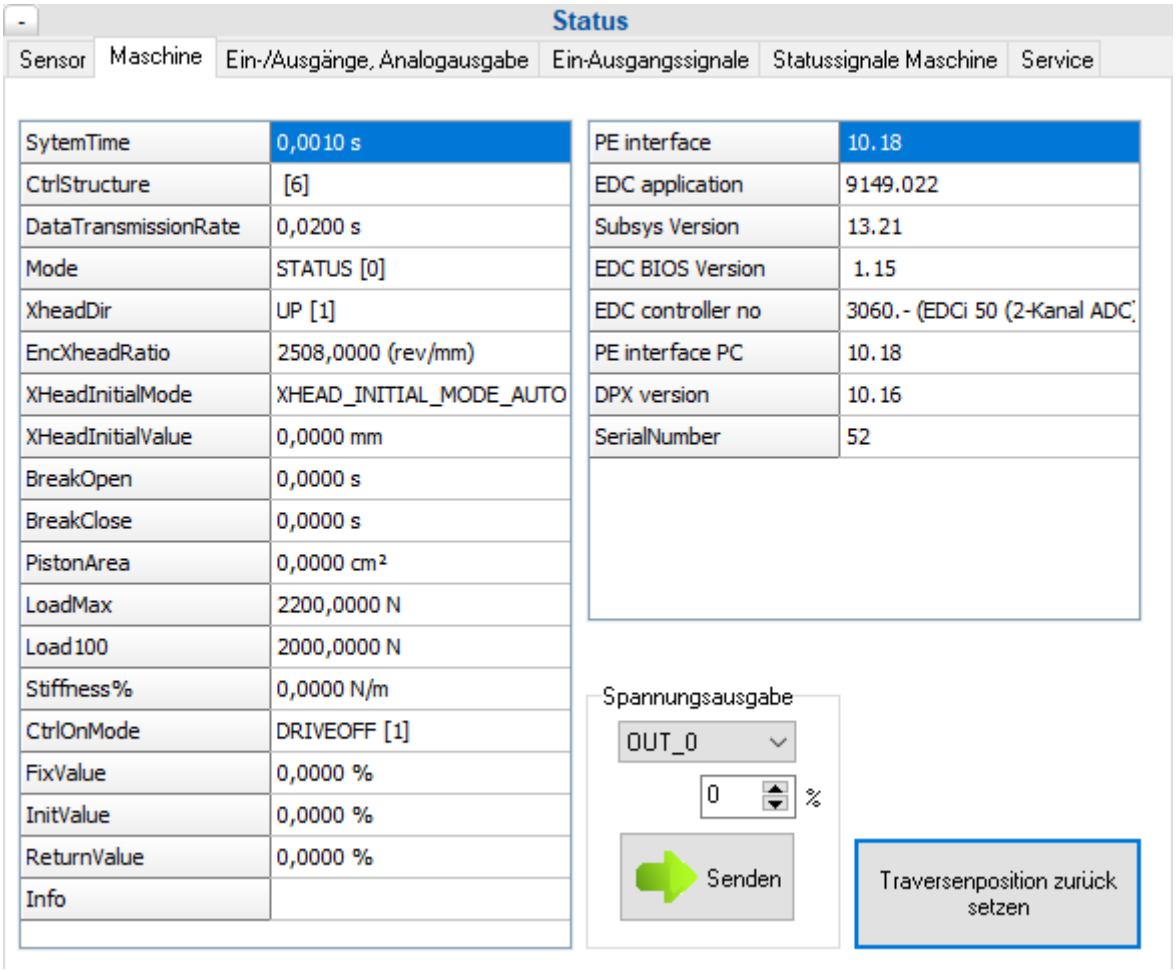


Illustration 22

The Inputs/Outputs, Analogue output page shows the configuration of the inputs and outputs of the EDC.

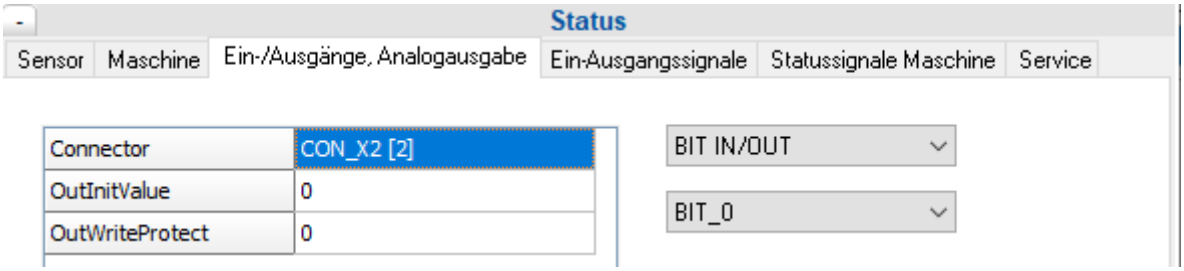


Illustration 23

The next page shows the current status of the inputs and outputs.

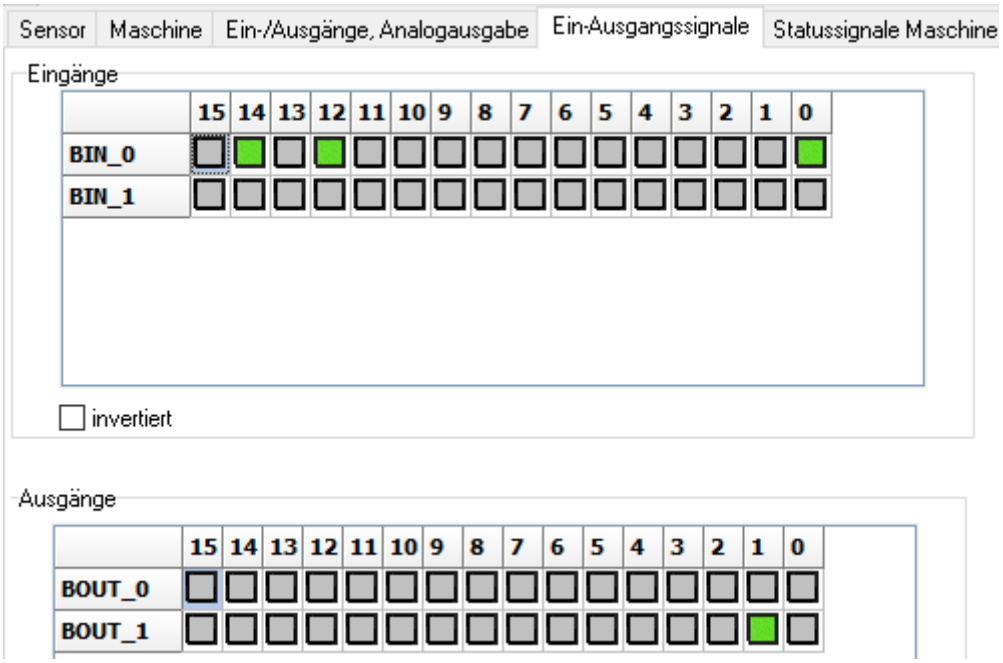


Illustration 24

The EDC also has a number of status signals, which are displayed on the page of the same name.

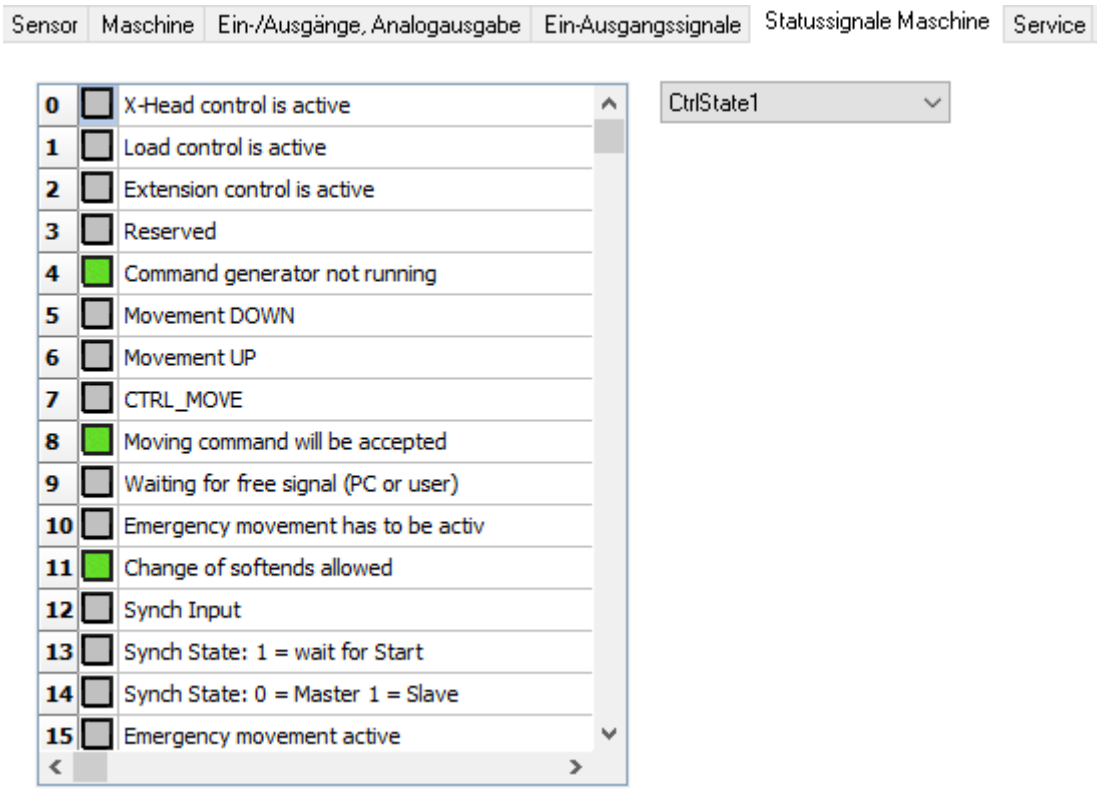


Illustration 25

Plugin control

The plug-in is controlled via Windows messages using the broadcast method:

```
PostMessage(HWND_BROADCAST,Message,PluginID,DockingHandle of the
panel)
```

As LabMaster controls the plugins via broadcast, the relevant plugin must be addressed via a unique plugin ID each time a message is sent. LabMaster determines the place where the plugin is to dock by passing the handle of a "placeholder component" (e.g. a panel in the main form). The plugin ID is coded in the WParam of the message.

The action to be performed is coded via the message:

WM_DOCKING_FRM_MSG:

Docks the plugin to a VCL component (e.g. a panel from LabMaster). The handle of the component to be docked to must be transferred in the Lparam.

e.g: PostMessage(HWND_BROADCAST, WM_DOCKING_FRM_MSG, iPluginID, panel1.handle)

Alternatively, the plugin application can be called with the command parameter:

-DOCKWND:<handle of the panel to be docked to>

can be docked.

WM_UNDOCK_FRM_MSG:

Undocking the plugin. The plugin then continues to run outside LabMaster.

WM_CLOSE_FRM_MSG:

Closes the plugin. The EDC panel is closed and terminated.

The messages described above must be registered in the operating system via the API function RegisterWindowMessage. The plain text names of the messages, e.g. "WM_DOCKING_FRM_MSG", can be set in the plugin (see***Desktop***

When closing LabMaster, all docked plug-ins must be undocked or closed by LabMaster. Otherwise, the plug-ins will continue to run without a visible window and can no longer be terminated "normally" by the user.

In addition to the messages listed above, which are sent from the host application to the EDCPanel, the following additional messages are sent from the EDCPanel to the host application:

WM_PLUGIN_DOCKED_MSG:

Feedback after successful docking to host application with
WParam = PlugIn ID
LParamHi = Coding for ext. manual operation (0 = not activated; 1 = use button for lower grip; 2 = button for upper grip)
LParamLo = ID=1 for identification as EDCPanel

WM_PLUGIN_UNDOCKED_MSG:

Feedback on the undocking of the EDCPanel from the host application with
WParam = PlugIn ID
LParam = ID=1 for identification as EDCPanel

WM_PLUGIN_STARTED_MSG:

Notification that an EDCPanel has been started with
WParam = PlugIn ID
LParam = ID=1 for identification as EDCPanel
This message ensures that an EDCPanel that is started later automatically docks to the host application that is already waiting.

WM_PLUGIN_TERMINATED_MSG:

Notification of the termination of the EDCPanel docked up to that point
WParam = PlugIn ID
LParam = ID=1 for identification as EDCPanel
This ensures that the host application can clean up the form to which the EDCPanel was docked.

These messages cannot currently be changed by the user.

Interface description TCP/IP connection EDC-UDC

The EDC-UDC software has a TCP/IP interface (Win socket) for communication with external devices/software. EDC-UDC works in principle as a server. Data is exchanged via telegrams in ASCII format.

Commands, status and error detection are defined within the syntax specified below. A TAN system (TAN = transaction number) is used to clearly identify the commands and assign the result.

Basic structure

Communication takes place using the following commands (string):

Keyword	**Keyword Description**
Msgstart	Reserved for later use
Msgend	Indicates the end of a data packet
acknowledged	Confirms a command
notacknowledged	Command was rejected
Getvalue	Client instructs the server to transmit a data packet
Sendcmd	Indicates the command set for a command from the client
server closing	Message from the server (closing the connection)
Stopaction	Optional command to the server to stop immediately

The separator between the individual parameters is the pipe character `(concatenation character) "| " (#124).`

Communication generally takes place from the client (polling); the data packet is only sent by the server at the start of the connection and when the connection is terminated by the server.

Establishing communication

The client establishes a connection to the server, which sends an acknowledged command when the connection is established.

```
Acknowledged|msgend
```

The client responds with `acknowledged`

```
acknowledged|msgend
```

Data requests

The client requests data from the server at time-synchronised intervals. This is done using the command `getvalue`

```
getvalue|msgend
```

The server responds with a data record with the following structure:

Position	**value**	**Type**	**Comment**
1	List of measured values	Floating point	Measured value, if not possible = -9999999999, decimal separator according to OS Physical units: [mm, N, s]
2	Status	Integer	Status of the device, always greater than 0
3	Error	String or integer	Error message of the device, preferably as an integer. 0 - no error 1 - Move Control Message (=error feedback of a movement, e.g. control deviation) 2 - Command error from EDC (command cannot be executed, e.g. incorrect parameterisation) 3 - Runtime error (error during command execution, e.g. emergency stop, end position, force limit) 4 - EDC error (error feedback from the EDC subsystem)

			5 - Internal error 6 - Event handling error (internal) 7 - Connection error 8 - Software error (internal)
4	TAN	Integer	TAN of the current command, 0 if no command is running
5	MsgEND	String	End Identifier of the message

The following statuses are defined:

Status	**Status Description**
0	stNone: no action or error / "basic status"
1	stInit: Initialisation, e.g. when connecting the machine control unit EDC
2	stReady: ready for new action/command
3	stBusy: Action is running; wait for feedback stDone or stError Comment: Command is active, TAN>0
4	stDone: Action successfully executed (TAN is reset to 0)
5	stError: Error status / action returns error
6	stOffline: EDC not connected

Example of a getvalue `telegram`:

`23.5;1.45;100.5;|2|3|0|msgend`

Current measured values: Force = 23.5 N, position = 1.45 mm; time = 100.5s

Status=2 =Ready,

Error message=3 =Runtime error

Command TAN =0=no command active.

The client responds with an Acknowledged message

Error:

Error messages are used to provide feedback when commands are not executed and in the event of faults in the system. An error message is sent until the error has been rectified.

TAN (TransActionNumber):

The TAN is defined when the client calls up a command with "sendcmd". It is sent back with each data record as long as the command is still active. Consecutive numbering of the TANs is recommended, whereby the TAN should be greater than zero.

Command structure

Commands to the server can be freely defined.

Each command from the client has the following structure

Position	**Value**	**Type**	**Comment**
1	sendcmd	String	Identifier Command follows
2	Command ID	Integer	ID of the command > 0
3	Parameter	Variant	Depends on the type of command, see next table Parameters are separated from each other by semicolons (see examples a) and b))
4	TAN	Integer	TAN of the current command
5	msgend	String	End identifier of the message

The following commands are defined:

Command ID	**Description Parameter**
0	No command / placeholder (CMDID_NONE)

1	Tare (CMDID_TARE) Tare the specified channel Parameter: - ChannelID: Number of the channel to be tared 1 = Time 2 = Position / angle 3 = Force / torque 4 = Elongation - TareValue: Tare value - DoBasicTare: 0=no→ Tare value is not saved in the setup 1=yes→ Tare value is saved in the setup - DoDisplayValue: 0= no→ TareValue is sent to the EDC as a new tare value, display value results 1= yes→ TareValue becomes new display value, tare value of EDC is calculated accordingly - UseChannelPos: 0 = Channel ID is given directly in the ChannelID parameter 1 = Channel ID given indirectly, ChannelID contains the position in the data transmission string **Attention: the differentiation in the DoDisplayValue parameter only works as described from EDCi onwards, for older EDCs the variant DoDisplayValue=1 is always executed here**
2	Hold command (CMDID_HOLD) Machine holds for the time specified in HoldTime in the controller channel HoldCTRL. The status is retained after the hold time has elapsed until a new command is sent.

	Parameter: - HoldCTRL: Controller channel - HoldTime: Hold time (the command is signalled back after expiry) - Deceleration: If the machine is in motion with regard to the specified controller channel when the command is started, it is decelerated with the delay specified here.
3	Move command (CMDID_MOVE) Ramp move command Parameter: - MoveCTRL: Controller channel for the movement to the target (ramp) 0 = CTRL_POS = Position/angle 1 = CTRL_LOAD = Force/torque - DestCTRL: Destination controller channel see MoveCTRL - LimitMode: Limit mode for controller channel Move (MoveCTRL) 0 = LIMIT_ABSOLUTE→ Limit is only monitored in one direction, the value must be in the direction of travel 1 = LIMIT_RELATIVE→ Limit is monitored in both directions 2 = LIMIT_NOT_ACTIVE→ No limit monitoring - DestMode : 0 = DEST_APPROACH→ Destination is monitored like limit, if exceeded, MoveCTRL is stopped in the controller channel (no controller switching) 1 = DEST_POSITION→ Controller switchover to DestCTRL when approaching the target 2 = DEST_MAINTAIN → - Speed: Speed in controller channel MoveCTRL - Destination: Destination in the controller channel DestCTRL

	- Limit: Limit for controller channel MoveCTRL - Acceleration: Acceleration in the controller channel MoveCTRL - DecelerationLimit: Deceleration when approaching the limit in the controller channel MoveCTRL - DecelerationDest: Deceleration when approaching the target in the MoveCTRL controller channel if no controller switchover takes place In controller channel DestCTRL, if controller switching takes place If zero is transferred, the nominal acceleration from the setup is used for Acceleration, DecelerationLimit and DecelerationDest.
4	Stop (CMDID_STOP) Immediate stop in the controller channel Position/Angle No parameters
5	Set softends (CMDID_SETSFT) Parameters (in this order): - SensorID: Channel for which the soft ends are to be set 0 = CTRL_POS = Position/Angle 1 = CTRL_LOAD = force/torque - UpperSoftend: upper limit - LowerSoftend: lower limit - Reaction: 0 = REACT_STATUS→ only status bit is set by EDC 1 = REACT_ACTION→ Machine is stopped
6	Simple movement command (CMDID_MOVEMANUAL) Machine is moved in the specified direction Paramater: - MoveCTRL: Controller channel for move command

	- Direction: Direction of the movement 0 = MOVE_HALT→ stop 1 = MOVE_UP→ upwards 2 = MOVE_DOWN→ downwards - Speed: Speed in the controller channel MoveCTRL - Acceleration: Acceleration in the controller channel MoveCTRL
7	Cyclical movement command (CMDID_CYCLE) Paramater: - MoveCtrl: Controller channel for move command - speed1: Speed to Dest1 - Dest1: First setpoint - Halt1: Hold time at Dest1 - Speed2: Speed to Dest2 - dest2: second setpoint - Halt2: Hold time at Dest2 - Cycles: Number of cycles - Speed: Speed to starting point (first setpoint) - Destination: Target position after all cycles have been run
8	Set output (CMDID_SETBITOUT) Sets a digital output of the EDC according to the Action parameter Parameter: - BOUTDev: Output device of the EDC (0..9) - BitNo: Number of the bit (0..15) - Action: 0 = DoSetBit

	1 = DoReSetBit 2 = DoFlashBit
9	Drive on/off (CMDID_DRIVEONOFF) Parameter: - OnOFF: 0 =Off 1 = On
12	Query sensor parameters (CMDID_GET_SENSORPARAM) Parameters (in this order): - Sensor number: integer 0..15 0 = Sensor_S (displacement) 1 = Sensor_F (force) 2 = Sensor_E (strain) 3 = DigiPoti 4..15 = optional sensors depending on application - Parameter number: integer 1..3 1 = Nominal value 2 = Upper range limit 3 = lower range limit 4 = upper soft end 5 = lower soft end Response telegram: acknowledged|sensor parameter|TAN|msgend Example regular request: `sendcmd|48|1;1;|1|msgend` Response from server: `acknowledged|3000|1|msgend` Example irregular request: `sendcmd|48|1;4;|1|msgend` Response from server: `notacknowledged|unknown parameter|1|msgend`
13	Switch configuration machine (CMDID_SELECT_MACHINE)

	The command can be used to switch between different configurations. The configuration data also contains the setup number of the EDC. When switching to a different setup, an existing connection is automatically disconnected as the EDC must be reinitialised. If the AutoConnectEDC option is activated for the newly loaded configuration, initialisation is started automatically. Parameter: - MachineID: Number of the machine from the list of available machines (1..8)
14	EDC connect / disconnect (CMDID_CONNECTEDC) Establish or disconnect the connection to the EDC Parameter: - Connect: 0 = disconnect 1 = connect
15	Request / release control point (CMDID_SETCTRLPOINT) Parameter: - NewCtrl: 0 = none 1 = RMC (only for EDCi) 2 = Internal software 3 = External software
16	Reset error (CMDID_RESETERROR) Resets the status of *stError* to *stReady* and the error to zero No parameters
17	Set the sign of the direction of travel (CMDID_SETDIRECTION) Parameter:

	- Direction: 1 = Pull or clockwise -1 = push or anti-clockwise
18	Query input signals (CMDID_GETBITIN) Parameter: - BINDev: integer = input device of the EDC (0..9) - BitNo: integer = number of the bit (0..15) Response telegram: acknowledged|Bit (0/1)|TAN|msgend
19	Drive command uncontrolled (CMDID_OPENLOOP) Parameter: - Direction: (1 = up; 2 = down) - Output: Output control value in % (0..100)

Examples:

Move up command (command ID=3)

```
sendcmd|3|0;1;1;1;0,1;100;0,5;0;0;0;|2|msgend
```

Control channel of the movement (move control): 0 = position-controlled

Control channel of the destination (destination control): 1 = force-controlled

LimitMode = LIMIT_RELATIVE

DestMode = DEST_POSITION

Speed = 0.1mm/sec

Destination = 100N

Limit = 0.5mm

Acceleration = 0→ Use nominal acceleration

DecelerationLimit = 0→ Use nominal acceleration

DecelerationDest = 0→ Use nominal acceleration

The EDC-UDC acknowledges the command with an Acknowledged response if the command can be executed, or with NotAcknowledged in the event of an error. The error is then indicated via the

error display in the EDC-UDC.

Cancelling the connection

If the connection is interrupted by the server, the following message is sent to the client:

```
server closing|msgend
```

The client is then informed, responds with an Acknowledged message and stops communication.

The StopAction command

If it is necessary to bring the server or the hardware connected to it into a stopped state (position control/S-stop) in the event of system faults (test stop, etc.), the client can use the command

in such cases, the client can use the command:

```
Stopaction|msgend
```

in such cases. This command is not acknowledged.

Integration in LabMaster

Procedure:

1. Create data column(s): Menu *Experiment module / Data columns / New*

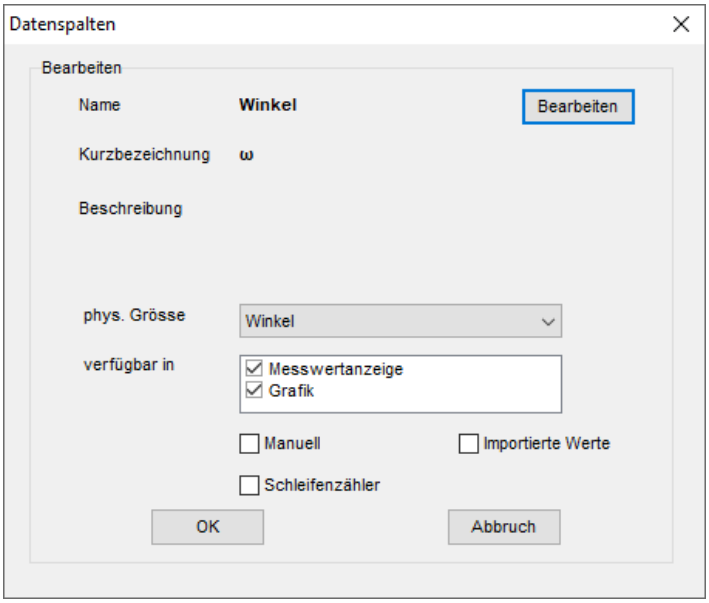


Illustration 26

2. Create user-defined channel: *Devices* / *Custom channel* / *New* menu
3. Make settings: Type: TCPIP(Cmd), set channels (see figure "User-defined channel settings for EDCPanel")

User-defined channel settings for EDCPanel

TCP/IP communication settings

- Address: IP address of the server service in the EDCPanel
- Number: Port number
- Data acquisition: Time interval at which data is to be requested from the EDCPanel
- External TAN:
- Stop action: LabMaster automatically sends Stop to the EDCPanel for events such as end of test
- Nos top action on test start: no stop is sent at the test start event
- Tare cmd ID: Command ID for taring
- No fatal communication error: EDCPanel has no error messages where communication is disconnected

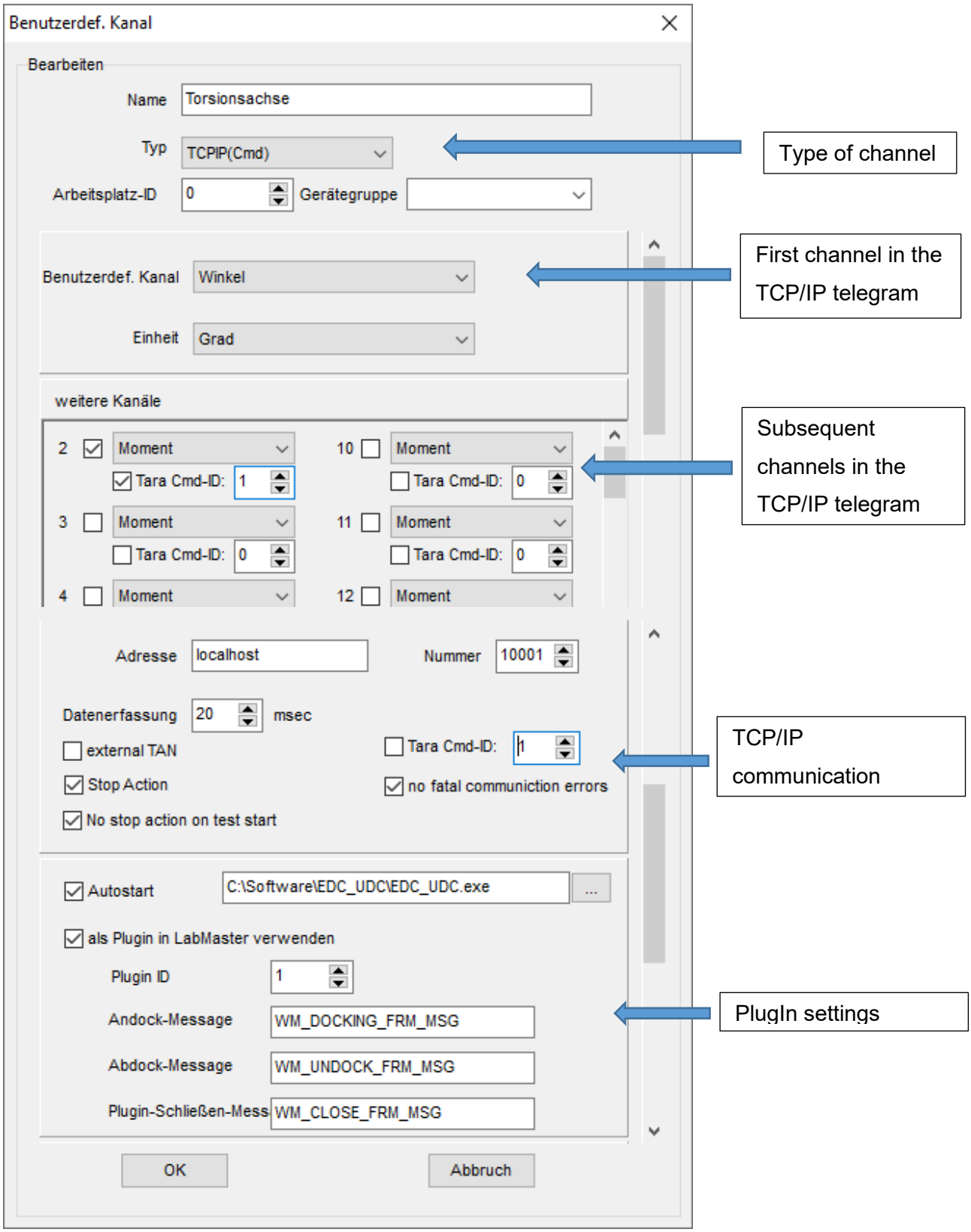


Illustration 27

PlugIn settings

- Autostart: EDCPanel is started by LabMaster when a measurement series in which it is activated is opened

- Use as plug-in in LabMaster: Dock EDCPanel in LabMaster
- Docking message: internal message for docking
- Undock message: internal message for undocking
- PlugIn close message: internal message for closing

Creating the command set

There is a predefined SQL script in LabMaster for creating the list of UDC commands, status and error messages. The complete command set can be inserted with one click via the *Devices* / *Settings* menu.

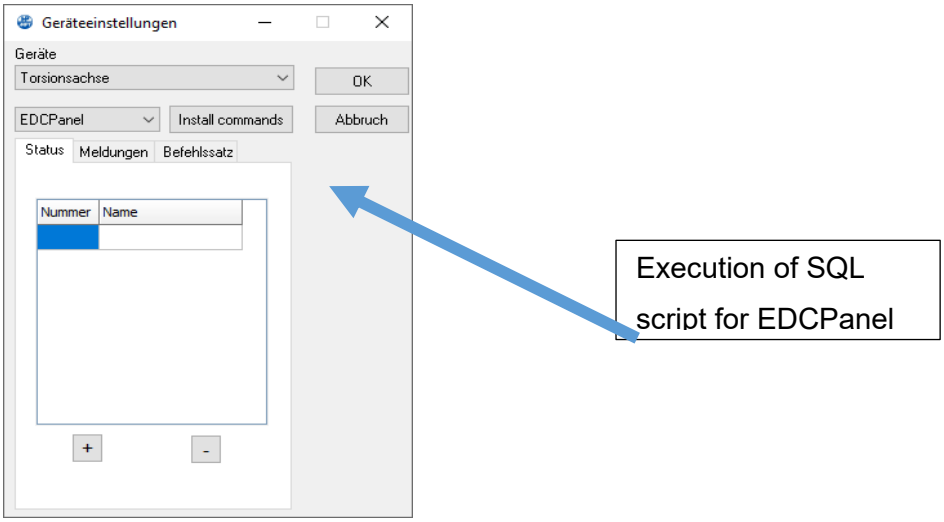


Illustration 28

Once the list has been generated, the command set page can be used to configure the command parameters.

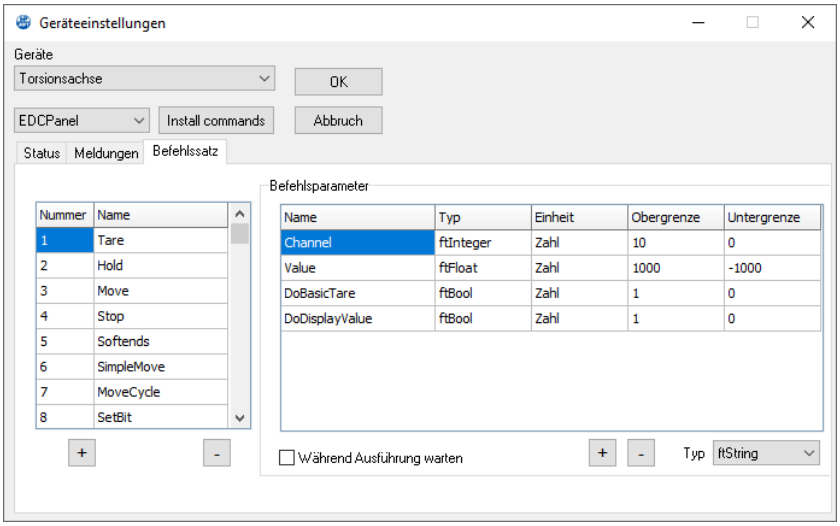


Illustration 29

Use in LabMaster

1. To use the device in a measurement series, it must be activated for this measurement series via *Measurement series* / *Parameters* / *Additional channel*. The data columns belonging to the device are then automatically added to the measured value recording.

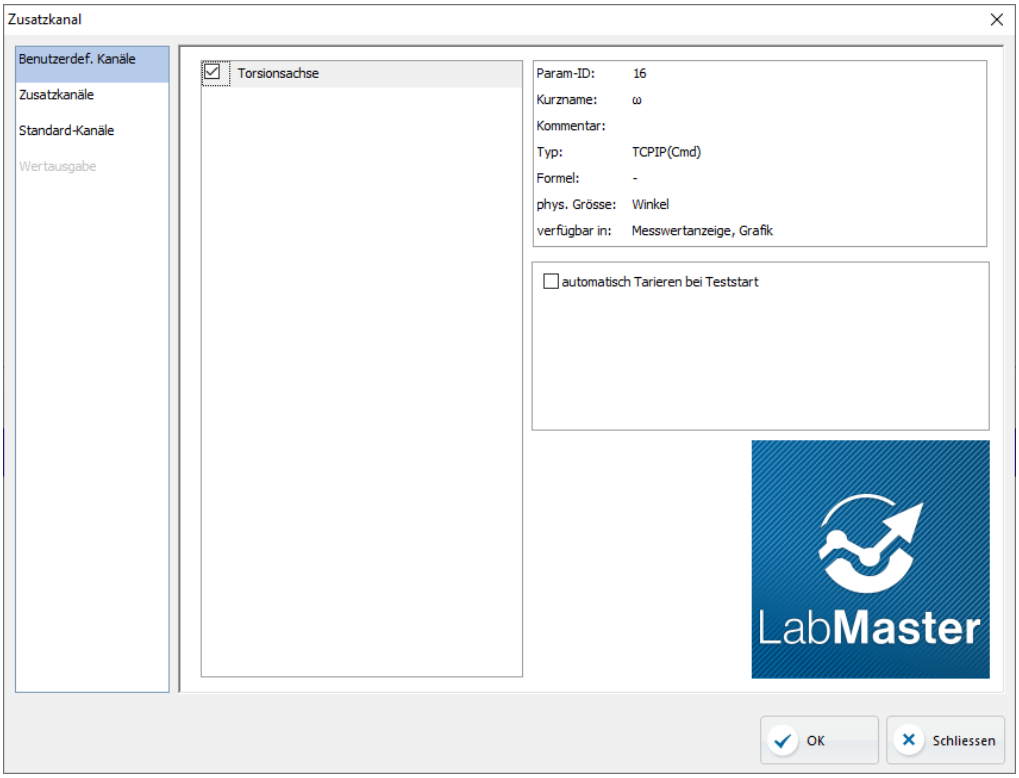


Illustration 30

2. In the test sequence (block programme), actions of the EDC panel are started from LabMaster via the so-called UDC command

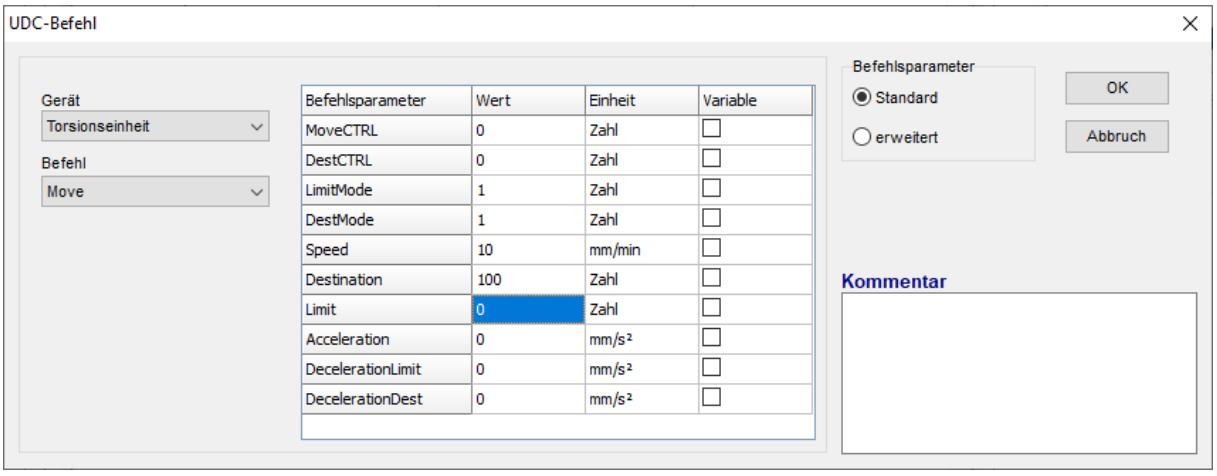


Illustration 31

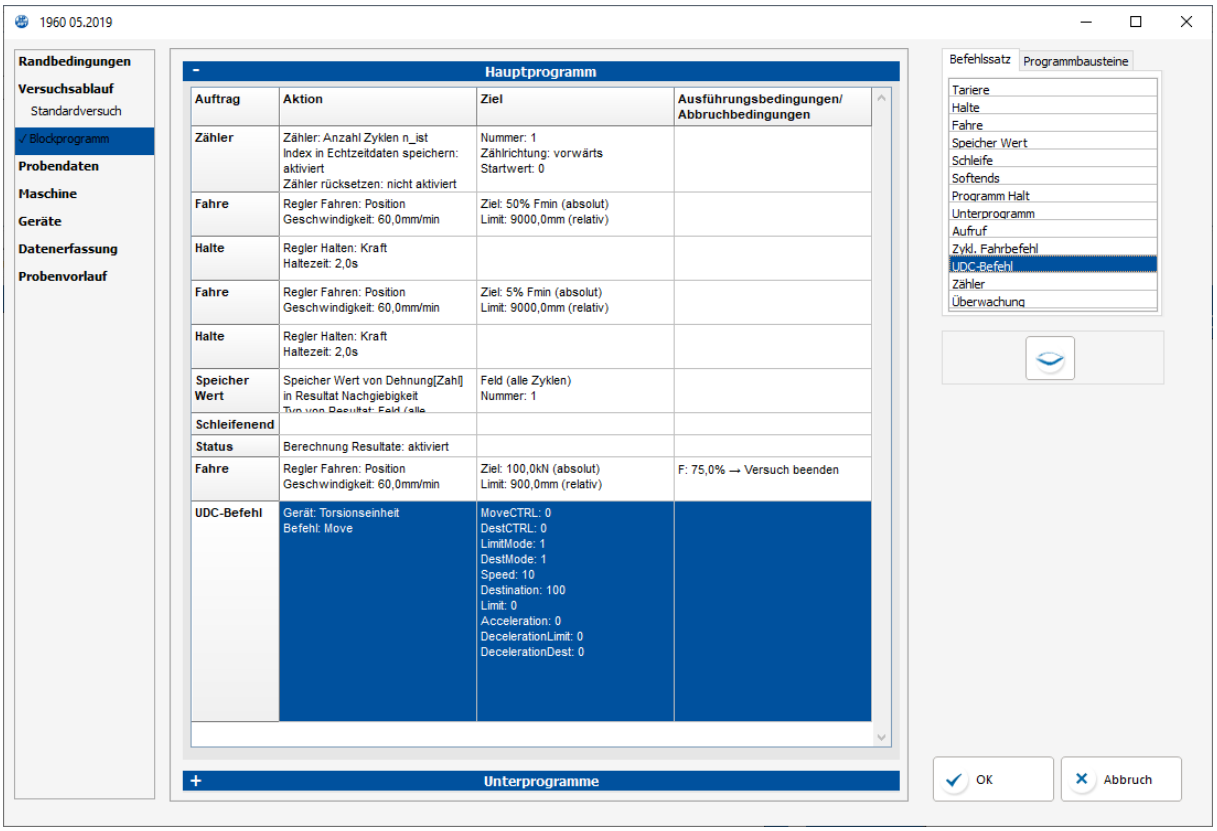


Illustration 32

3. Various statuses of the EDC panel can be reacted to in the block programme within event handling (*abort at event → External event → User-defined channel*).

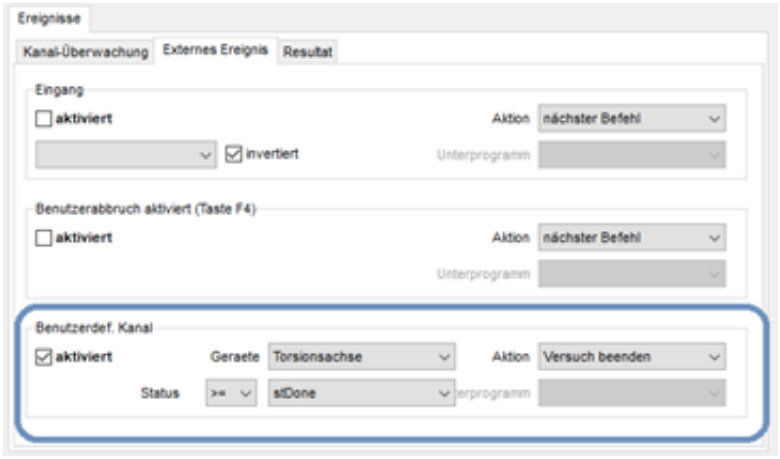
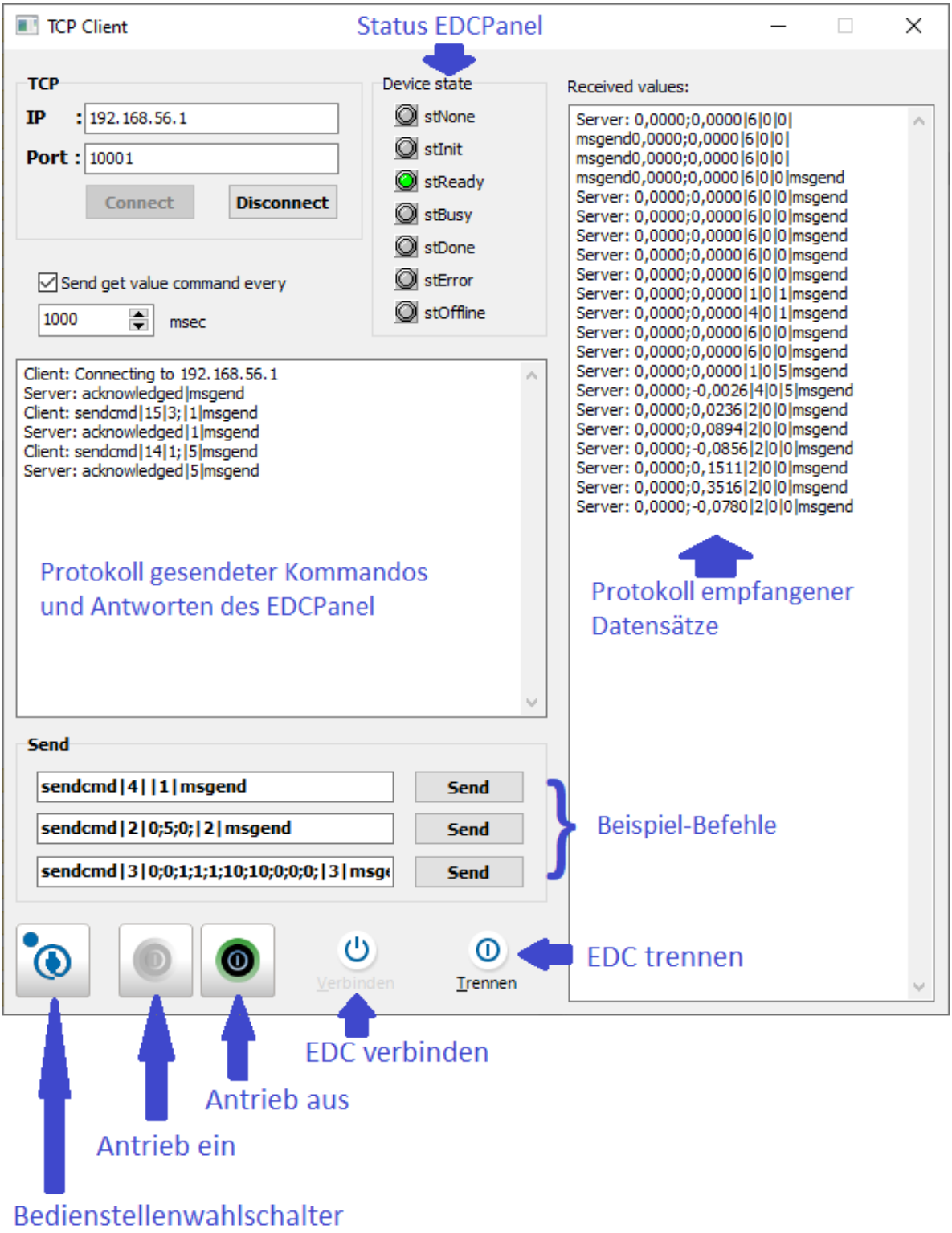


Illustration 33

Test programme TCPClient

The "TCP Client" test programme is installed together with the EDCPanel. This can be used to test the functionality and effectiveness of the commands.



The corresponding commands are already stored for the buttons in the lower area.

The buttons are activated depending on the current status of the EDC panel. Commands can be entered and sent in the three lines above.

The connection data for the TCP/IP connection to the EDCPanel can be set in the upper area. The "Send get value command" option can be used to activate an automatic query of the measured values and status information at the specified intervals. These measured value data records are entered in the list on the right-hand side.

The list on the left logs all commands sent and the corresponding responses from the EDC panel.