37164B

LeoPC1
Engineering Manual

Manual
Software version 3.1.2

Manual 37164B

WARNING

Read this entire manual and all other publications pertaining to the work to be performed before installing, operating, or servicing this equipment. Practice all plant and safety instructions and precautions. Failure to follow instructions can cause personal injury and/or property damage.

The engine, turbine, or other type of prime mover should be equipped with an overspeed (overtemperature, or overpressure, where applicable) shutdown device(s), that operates totally independently of the prime mover control device(s) to protect against runaway or damage to the engine, turbine, or other type of prime mover with possible personal injury or loss of life should the mechanical-hydraulic governor(s) or electric control(s), the actuator(s), fuel control(s), the driving mechanism(s), the linkage(s), or the controlled device(s) fail.

Any unauthorized modifications to or use of this equipment outside its specified mechanical, electrical, or other operating limits may cause personal injury and/or property damage, including damage to the equipment. Any such unauthorized modifications: (i) constitute "misuse" and/or "negligence" within the meaning of the product warranty thereby excluding warranty coverage for any resulting damage, and (ii) invalidate product certifications or listings.

CAUTION

To prevent damage to a control system that uses an alternator or battery-charging device, make sure the charging device is turned off before disconnecting the battery from the system.

Electronic controls contain static-sensitive parts. Observe the following precautions to prevent damage to these parts.

- **Discharge body static before handling the control (with power to the control turned off, contact a grounded surface and maintain contact while handling the control).**
- **Avoid all plastic, vinyl, and Styrofoam (except antistatic versions) around printed circuit boards.**
- **Do not touch the components or conductors on a printed circuit board with your hands or with conductive devices.**

OUT-OF-DATE PUBLICATION

This publication may have been revised or updated since this copy was produced. To verify that you have the latest revision, be sure to check the Woodward website.

The revision level is shown at the bottom of the front cover after the publication number. The latest version of most publications is available at:

http://www.woodward.com/publications

If your publication is not there, please contact your customer service representative to get the latest copy.

Important definitions

WARNING

Indicates a potentially hazardous situation that, if not avoided, could result in death or serious injury.

CAUTION

Indicates a potentially hazardous situation that, if not avoided, could result in damage to equipment.

NOTE

Provides other helpful information that does not fall under the warning or caution categories.

Revision History

Rev.	Date	Editor	Changes
B	2013-12-16	GG	New software version LeoPC1 3.1.2 now is Windows 7 compatible. Manual was checked for operating system related changes, address, and typo corrected.
A	2008-02-29		Updated for new software version LeoPC1 3.1.1
NEW	2002-10-17		Release

Contents

Chapter 1.
General Information

Intended Use The configuration files were accordingly adjusted to the delivered units. The prerequisite for a proper and safe operation of the product is correct installation and configuration as well as a careful operation.

NOTE

This manual has been developed for a maximum usage of the software. Functions, parameters and other details described, which do not exist on your unit may be ignored.

The present manual has been prepared for processing the configuration files of the software.

It gives an overview over all parameters and function commands, which can be defined in the configuration files.

Chapter 2.
Software Configuration

NOTE

Please take heed that the following descriptions are given for those who have specific knowledge in editing such files. It is recommended to change parameters only via application dialogs.

Plant configuration (*.cfg)

The parameters for the application, plant and display are defined in the plant configuration (CFG file).

NOTE

Communication parameters and other individual parameters are written to the CFG while the software tool is being used. Use of the plant configuration is therefore possible only if the CFG is not write-protected.

Description of the application parameters

All values that the software tool requires to correctly display or execute global displays, file allocations, settings and connections are defined as application parameters. Most of these parameters are edited via dialogs.

Global specifications [Global]

Describes and defines global, overriding parameters.

Example
```
[GLOBAL]
caption=- Demo
prncap=Demo
ShowAppName=1
logo=LOGO.BMP
LogoX=0.2
LogoY=0.1
LogoCX=3.21
LogoCY=0.90
EventMemoryFile=Demo.dat
DL=DL\
ALARMS=ALARMS\
ErrorHelpfile=
ComDriver=Direkt
ComAutostart=0
ComIdleSleep=200
TestMode=0
CommRMC=0
CommVisual=2
UseEcho=1
```

Parameters for title bars and logos

Caption <your title bar>
Description Alternative or additional program headers for the title bar. If this entry is not specified, the path and name of the loaded configuration file is used.
Example `caption=- <your title bar>`
Editable via: Devices.. Service configuration.. Text within title row

ShowAppName **1** .....(Application name is displayed in title bar)
oder **0** .....(Application name is not displayed in title bar)
Example `ShowAppName=1`
Description Defines whether the application name is to be displayed on the title bar or not.
Editable via: Devices.. Service configuration.. Show name of program as title?

Prncap
Description is not used
Example `prncap=<Dateiname>`

Logo prnLogo.bmp
Description Bit map that is used for program printouts.
Example `logo=prnLOGO.BMP`

logoX 0.2
logoY 0.1
Description Bit map positions X and Y on the printout.
Example
```
LogoX=0.2
LogoY=0.1
```

logoCX 3.21
logoCY 0.90
Description Bit map lengths X and Y on the printout.
Example
```
LogoCX=3.21
LogoCY=0.90
```

Parameters for file and directory assignments

DL DL\
Description Directory in which the temporary files of the data logging are created. Please take heed that the path is always specified with a concluding backslash "\".
Example `DL=DL\`

Alarms ALARMS\
Description Directory in which the alarm protocol is stored. Please take heed that the path must always be specified with a concluding backslash "\".
Editable via: Devices.. General settings.. Path for alarm files
Example `ALARMS=Alarms\`

EventMemoryFile EventRecorder.dat
Description File in which the event memory data is temporarily backed up.
Example `EventMemoryFile=EventRecorder.dat`

ErrorHelpfile
Description Help file for plant-specific alarm messages, if available.
If no file is specified, the function is deactivated.
Editable via: Devices.. General settings.. Help file for alarms
Example `ErrorHelpfile=`

Parameters for communication

ComDriver
Direct ..... (Direct parameterization)
GW4 - RS232 (Gateway-RS232- connection)
Modem ..... (Gateway-Modem- connection)
IXXAT VCI2 - CAN (CAN bus connection via VCI2 driver software)
Description: Defines the driver that is to be used for communication.
Editable via: Devices.. General settings.. Driver
Example: `ComDriver=Direkt`

ComAutostart
0 ..... (no attempt to connect automatically)
1 ..... (data communication is started automatically)
Description: Defines whether a data communication is started automatically after loading the plant configuration or not.
Example: `ComAutostart=0`
Editable via: Devices.. General settings.. Start communication automatically while (...)
FAQ: It is recommended that this value be set to "0" upon delivery.

ComIdleSleep
200 ..... (most adequate value)
Description: Pause in ms between the continuous requesting/writing of values. The remote control data and remote monitoring data are included in this category.
Example: `ComIdleSleep=200`
Editable via: Devices.. Service configuration.. Break for data transfer

TestMode
0 ..... (customary standard)
1 ..... (no item date required; default data will be used in configuration window)
Example: `TestMode=0`
Description: This entry defines whether data should be ordered via communication driver or default values are used.
FAQ: This entry is used for the demo version.

CommRMC
0 ..... (not active, according to standard)
1 ..... (active)
2 ..... (while configuration/load language deactivated)
Description: This entry defines whether the remote control is active. The possibility exists of permanently activating/deactivating the remote control or only temporarily deactivating it for configuration and load language.
Example: `CommRMC=0`
Editable via: Devices.. General settings.. Remote control

CommVisual
0 ..... (not active, according to standard)
1 ..... (active)
2 ..... (while configuration/load language deactivated)
Description: This entry defines whether data updating is active for display. The possibility exists of permanently activating/deactivating the updating or only temporarily deactivating it for configuration and load language.
Example: `CommVisual=2`
Editable via: Devices.. General settings.. Displays

UseEcho
0 ..... (deactivates a detailed check of the command answer)
1 ..... (activates a detailed check of the command answer)
Description: This entry allows that the answers of the unit are checked more precisely on a read- or write command. The parameter report is depending on the used driver protocol.
Example: `UseEcho=1`

Definition of external tools [External Tools]

ToolCount **1 bis n** .....(complies with the defined number of tool lines)

Description This parameter defines how many entries are available in menu „Tools“. The number of **Tool<n>** entries must be defined accordingly.

Example `ToolCount=1`

Tool0
Tool<n> &<Menu entry>;["<Application>"]"<File path>" <Parameter list>

Description This parameter contains the definition of the respective tool. Starting with the entry in menu “Tools”, followed by a list of parameters including the file or program path which can vary according to the tool.

Example `Tool0=&GetConfig;"WScript.exe" "%APP_PATH%GetConfig.vbs" %LNG%;0`

Transfer parameter	Description	Example
%APP_PATH%	Path of the application list	C:\Program Files\LeoPC1
%CFG_PATH%	Path of the list for CFG files	C:\Program Files\LeoPC1
%ALM_PATH%	Path of the list for alarm files	C:\Program Files\LeoPC1\Alarms
%DL_PATH%	Path of the list for data recording	C:\Program Files\LeoPC1\DL
%LNG_PATH%	Path of the list for language files	C:\Program Files\LeoPC1\Lng
%STD_PATH%	Path of the list for standard value files	C:\Program Files\LeoPC1\Std
%LNG%	Current active language of the application	English
%COM%	Contents of the active COM-Port-settings (Port, Baud, Parity, Data Bits, Stop Bit)	COM1,9600,0,8,1

Specifications for configuration [Config]

DataBufferSize **65535** .....(maximum value)
<VisValuesStartID +VisValuesCount +1>
<highest para-ID +1>

Description Size of the buffer per device in which the data is temporarily stored.
In the lower range the visualization messages and in the upper range the parameter values were saved.

Example `DataBufferSize=65535`

Editable via: Devices.. Service configuration.. Size of data buffer

FAQ The entry must be larger than the highest parameterization ID used in the .asm file. And the entry must be larger than the sum of VisValuesStartID and VisValuesCount. Otherwise errors can occur.

VisValuesStartID **50000** .....(according to standard of the new product generation)
900 .....(according to standard of the former product generation)
<Maximum parameterization ID +1> (minimum value)

Description Defines the first ID within the buffer starting for the remote monitoring.

Example `VisValuesStartID=900`

Editable via: Devices.. Service configuration..ID for the first setting for remote monitoring

FAQ The entry must be larger than the highest parameterization ID used in the ASM file, otherwise the parameters that have larger IDs than the given value are not displayed in the dialogs.
If extended standard protocol is not used for Gateway-RS232 or Modem communications you can not use parameter-IDs larger than or equal to 900.

VisValuesCount **70** ..... (example)
<Maximum visualization ID +1> (minimum value)
Description Defines the number of values that are used for the remote monitoring. All IDs which are smaller than this value can be edited in the asm file.
Example `VisValuesCount=70`
Editable via: Devices.. Service configuration.. Size of buffer for remote monitoring
FAQ This value must be the largest display-ID used in the ASM file plus one. It is dependent on the type of device.

IDControl **503** ..... (according to predetermined standard)
Description ID for the control word (acknowledgement, remote start and remote stop) in the remote control.
Example `IDControl=503`
Editable via: Devices.. Service configuration.. ID for remote control word

Control command	Remote start	Remote stop	Others	Acknowledgement	Others
Bit:	0	1	2 + 3	4	5 - 15
Value	1	1	always 0	1	any

Modules <specific code>
Description Coded control word, which defines the Enable program modules.
Example `Modules=3504.95.2707`
Editable via: Devices.. Enable program modules

Specifications for the communication drivers [Modem, GW4, DiAc, RS232, CAN]

[RS232]	Identification for COM-Interfaces settings
[Modem]	Identification for Modem settings
[GW4]	Identification for Gateway-RS232 settings
[DiAc]	Identification for Direct settings
[CAN]	Identification for CAN bus settings
[ID-Mappings]	Identification for the allocation table for CAN-ID to device number

```
[Modem] ;* Modem driver
Init=ATH
Connect=ATDT
Disconnect=+++~~~~ATH^M
PhoneNumber=1
EditPhoneNumber=1
Connected=CONNECT
TimeoutInit=40000
TimeoutDial=40000
WriteReply=10
TimeoutWrite=500
TimeoutReadError=1000
ReadReply=10
TimeoutRead=700
TimeoutReadNoAnswer=200
RcNoAnswer=0
RcNoAnswerWait=100
ExtStandardProtocol=0

[GW4] ;* Gateway-RS232 driver
WriteReply=3
TimeoutWrite=100
TimeoutReadError=0
ReadReply=5
TimeoutRead=500
TimeoutReadNoAnswer=200
RcNoAnswer=0
RcNoAnswerWait=100
ExtStandardProtocol=0

[DiAc] ;* Direct driver
WriteReply=5
TimeoutWrite=50
TimeoutReadError=0
ReadReply=5
TimeoutRead=500
TimeoutReadNoAnswer=100
TimeoutStatusNoResponse=0

[RS232] ;* serial COM-Interface
Baud=9600
Parity=0
DataBits=8
StopBits=0
WriteDelay=150

[CAN] ;* CAN bus driver
TimeoutRead=500
WriteReply=5
Baud=7171
BufferOffset=1
WriteCanID=831

[ID-Mappings] ;* CAN bus driver
801=1
```


NOTE

Most driver settings are dependent on the connecting type and on the unit type as well as from the used PC system configuration and from the operating system.

General parameters that occur in all drivers

The following are editable via: Devices.. Driver/Settings... Options...

TimeoutRead **50 to 1000** ... (according to standard range)
Description Maximum time in ms for which an answer is waited.

WriteReply **0 to 8** ..... (according to standard of the new product generation)
3 to 10 ..... (according to standard of the former product generation)
Description Number of repetitions that pass for send command.

The following are editable via: Devices.. Driver/Settings...

Baud **9600** ..... (standard for serial COM-Interfaces)
7171 ..... (customary 125kB for CAN bus interfaces, monitoring bus)
Description Transfer speed setting (baud rate).

General parameters that occur in all serial drivers

The following are editable via: Devices.. Driver/Settings... Options...

TimeoutWrite **50 to 1000** ... (according to standard range)
Description Maximum waiting time in ms after writing a command.

WriteDelay **0 to 150** ..... (according to standard range)
Description Time in milliseconds of maximum delay between writing a command and the further processing.

ReadReply **1 to 10** ..... (according to standard range)
Description Maximum number of timeouts that will pass while waiting for an answer.

TimeoutReadError **0 to 1000** ..... (according to standard range)
Description Maximum waiting time in ms, if a CAN error is pending. This parameter is used in gateway connections.

TimeoutReadNoAnswer **0 to 500** ..... (according to standard range)
Description Maximum time in ms that is waited, if no answer comes.

The following are editable via: Devices.. Driver/Settings...

Parity **0** ..... (according to standard)
Description Definition of the parity used for Direct/Gateway-RS232/Modem driver.

DataBits **8** ..... (according to standard)
Description Quantity of data bits sent for Direct/Gateway-RS232/Modem driver.

StopBits **0** ..... (according to standard)
1 ..... (according 1.5)
2 ..... (according 2)
Description Definition of the stop bit used for Direct/Gateway-RS232/Modem driver.

Special parameters that occur only in Gateway-RS232, Modem driver

The following are editable via: Devices.. Driver/Settings... Options...

RcNoAnswer
0.....(waiting for an answer is enabled)
1.....(no answer is waited for)
Description: Definition of whether an answer is to be awaited from the remote control.

RcNoAnswerWait
0 to 1000 .....(according to standard)
Description: Maximum time in ms that will be waited for a remote control answer.

ExtStandardProtocol
0.....(Standard protocol is used, L, W commands with decimal IDs)
1.....(extended standard protocol is used, l, w commands with hexadecimal IDs)
Description: Defines the protocol used for Gateway connections.

Special parameters that occur only in Direct and CAN bus drivers

TimeoutStatusNoResponse
0 to 500 .....(according to standard)
Description: Maximum time in ms, that will be waited for, until connection will be regarded as not active or disturbed, if any (display) data is not received.

Special parameters that occur only in modem drivers

The following are editable via: Devices.. Driver/Settings... Options...

Init
ATH .....(standard initialization for local loop)
ATH&FO ....(initialization within key telephone systems) according to the modem manual.
Description: Command for initialization of the modem.

Connect
ATDT.....(MFV, tone dialing)
ATDP.....(IWV, pulse dialing)
ATX1DT/P ..(MFV/IWV within key telephone systems)
Description: Command for the modem dialing process.

Disconnect
+++~~~**ATH^M**
Description: Command for termination of the connection by hanging up.

PhoneNumber
0W<valid telephone number, and country- and city code> (with exchange line)
<valid telephone number, with country- and city code> (without exchange line)
Description: Please enter here the telephone number which shall be used.

EditPhoneNumber
1.....(allows you to edit the telephone no. before dialing)
0.....(dials directly the recorded connection)
Description: Option for requesting the input of another telephone number as the default.

Connected
CONNECT
Description: Check value for the answer for confirmation of connection setup.

TimeoutInit **40000 to 100000** (standard range)
Description Time available for initialization of the modem in milliseconds (displayed in seconds).

TimeoutDial **40000 to 100000** (standard range)
Description Time available for the modem dialing process in milliseconds (displayed in seconds).
FAQ For foreign connections Timeoutinit and TimeoutDial increase to 60 or more.

Special parameters that occur only in CAN drivers

The following are editable via: Devices.. Driver/Settings... Options...
BufferOffset **1** ..... (according to default standard)
Description Definition of the offset digit for the display buffer with the CAN driver.

WriteCanID **831** ..... (according to default standard)
Description Definition of the CAN ID for parameterization via the CAN driver.

The following are editable via: Devices.. Driver/Settings... CAN-ID...
[ID-Mappings] **801=1** ..... (example of a CAN-ID allocation that is used often)
830=x ..... (highest valid CAN ID)
Description Definition of the CAN ID for remote monitoring by means of CAN driver. Allocation of CAN IDs to a device no., that will be used in the configuration for remote monitoring via CAN bus driver.
FAQ For each device, that is sending display data via CAN bus, a unit ID must be assigned. If nothing is specified here and it is attempted to display data via CAN bus, erroneous display of the values can come about.

Specifications for initialization [Data]

1501 **x** ..... (x complies with the requested real power set value)
1502 **y** ..... (y complies with the requested Cosinus-Phi)
<Serial number>501
<Serial number>502
Description Values that are used for the remote control.

Example
```
1501=E0050
1502= 1.00
```
Editable via: Devices.. Remote control..

Real power set value (x)	Outgoing power (Export)	Fixed value power (Constant)	Reference power (Import)
Input	E<value>	C<value>	I<value>
internal value	<value>	<value> + (32768 / 2)	<value> + 32768
setting Bit 14-15	0x00	0x01	0x10
<Value> (Standard)	0 to 6900	0 to 6900	0 to 6900

Cosinus-Phi (y)	capacitive		inductive
Input	k0.01	1.00	i0.01
internal value	-1	100	1
maximum valid range	-99 to -1	100	1 to 99

Specifications for Short-term storage [Trigger]

These parameters are written in the CFG file automatically, when settings are done in the module Short-term storage..

ToolDescription <ENGINE.Name – TOOL.Name>
Description Contains the currently selected device with its plant allocation.
Example `ToolDescription=Generator 1`
Editable via: Devices.. Short-term storage.. General.. Settings... Device

The following are editable via: Devices.. Short-term storage.. General.. Settings...

ActivationDescription <Activation condition>
Description Displays the current triggering event syntax. Conditions can be an alarm message, a measured value and a limit value or a display-ID xx, an operator and a unformatted value.
Example `ActivationDescription=Measuring value 'Battery voltage' > 26`
Editable via: Error / alarm, exceeding a limit value, manually value

AtIssueTime <At issue time>
Description Minimum time in seconds the event must be pending for the purpose of activation.
Example `AtIssueTime=3`

SamplingRate <Sampling rate>
Description Minimum time in seconds that should exist between two pollings.
Example `SamplingRate=2`

HoldBackTime <Hold-back time>
Description Time in seconds which is co-logged prior to occurrence of the event.
Example `HoldBackTime=100`

FollowUpTime <Follow up time>
Description Time that is still logged after cessation of the event.
Example `FollowUpTime=200`
FAQ HoldBack and FollowUpTime added together make up the logging time period.

Filename <File name>
Description Designation of the file in which an event time period is to be saved.
Example `Filename=VBatt_%y%m%d-%H%M%S.llo`

Permitted Formattings	Year	Month	Day	Weekday / Week	Hours / Minutes / Seconds
Input	%Y / %y	%m / %b / %B	%d / %a / %A	%w / %W	%H / %M / %S
Format (example)	2004 / 04	06 / Jun / June	30 / mo / Monday	3 / 27	10 / 55 / 45

FAQ Please take heed that formatting with special characters are not suitable as output for file names. Saving will fail.

The following are controlled via internal application processes

DeviceNo <Device no.>
Description Number of the selected device for the purpose of internal double checking.
Example `DeviceNo=1`

Mode **0** ..... (activated by an error)
1 ..... (activated by exceeding a limit value)
2 ..... (activated by edited value)
Description Represents the selected mode for activating data to be logged.
Example `Mode=1`

ID <display-ID +VisValuesStartID> or one of them
Description Contains the internal value for activating as a check value.
Example `ID=19`

Bitmask <hexadecimal number>
Description Contains the hex value of the alarm option that corresponds with the error masking.
Example `Bitmask=0`

DLValue <natural number>
Description Contains the decimal value of the options with limit values.
Example `DLValue=26`

Operator **0 bis 5** ..... (range of values)
Description Contains the value of the selected operand.
Example `Operator=0`

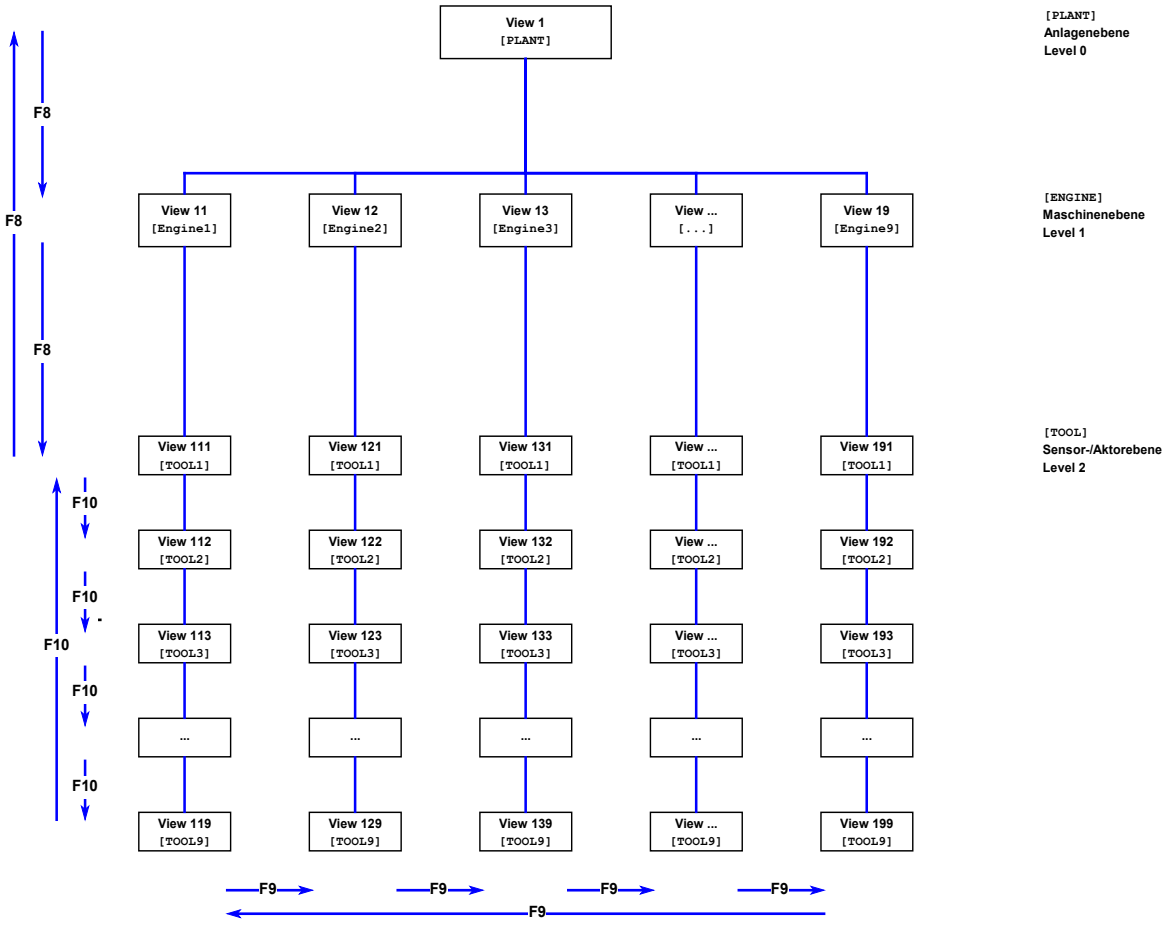
Operator	**AND**	**less than**	**less than or equal to**	**larger than**	**larger than or equal to**	**equal to**
Description	Logical AND, for analyzing binary values	Less than specified value	Less than or equal to specified value	Larger than specified value	Larger than or equal to specified value	Equal to specified value
Input	AND	<	<=	>	>=	=
Memory value	0	1	2	3	4	5

Description of a plant

Configuration of a plant takes place in two steps.

The plant must be described as step one. This is effected on three levels in the **CFG** file. The names of the machines and devices, the parameters for display and the assignments are defined.

The description of the contents of the views that have to be selected for the user is then effected as the second step. This is effected in the assigned **ASM** files and, if necessary, in an **OPT** file for dynamic configurations.



Example

```
[PLANT]
NAME=Demo
DEF_VIEW=1
ENGINE1="Engine 1"

[ENGINE1]
TOOL1="Device 1"

[ENGINE1TOOL1]
DEVICE_NR=1
CanRead=1
ASM-FILE=%ASM_PATH%\Devive1.asm
```

Definition of the plant level [PLANT]

The level **[PLANT]** is allowed to be present once only in each configuration. The standard view **DEF_VIEW** and the machines **ENGINEx** that are present in the plant are specified here.

NAME "<unique character chain>"
Description Name of the plant (is not used in the application so far).
Example `NAME=Demo`

DEF_VIEW **1** ..... (standard for plant level)
<VIEW-Nummer>
Description View that is displayed after loading the plant configuration.
Example `DEF_VIEW=1`

ENGINEx "<unique character chain>"
Description All entries within the segment **[PLANT]** that start with **ENGINE** are interpreted as being machines. The value specifies the name of the machine. Virtually a random number of machines can be defined. All ENGINEx entries must be **unique.** A numeric sequence can be used for x. The machine is described in more detail in another segment via this designation of the entry.
Example `ENGINE1="Engine 1"`
FAQ Up to now ascending values for x of between 1 and 9 have proven to be good.

Definition of the machine level [ENGINEx].

A more detailed specification of machines is effected on this level **[ENGINEx].** One or several devices can be defined for each machine **TOOLx.** The device is described more precisely later via the designation **x** (a numeric sequence). The value **"abc 123"** for each device specifies the name of the corresponding device. The segments for the individual machines are defined by the designations in the segment [PLANT].

TOOL1
TOOL<n> "<unique character chain>"
Description Definition of the displayed name of the corresponding device.
Example `TOOL1="Device 1"`
FAQ This value may also appear on the printouts and is therefore limited in length. However, it does on the other hand allow a differentiation of the devices, e.g. by including version numbers.

Definition of the tool level [ENGINExTOOLx]

The segments for the individual machines are defined by the designation from the segment **[ENGINEx]** and the relevant designation of the machine **TOOLx.** The two designations are put together, thus providing the segment name for the device **[ENGINExTOOLx].**

DEVICE_NR <natural number>
Description Device no. of the device that is for the purpose of assigning the commands
Example `DEVICE_NR=1`

CanRead **0** ..... (read not permissible)
1 ..... (read permissible)
Description This entry defines whether values are to be read by the control for monitoring.
Example `CanRead=1`

ASM-FILE %ASM_PATH%\<file name>.asm
Description Defines the file that contains the more exact parameterization and display descriptions of the device. Specification as an absolute path or by means of a path variable that is defined in the software system settings and is assignable.
Example `ASM-FILE=%ASM_PATH%\Device1.asm`

Description of a view

For the views **bit maps** can be defined for the background display, **objects** and **buttons.** Which objects or buttons are displayed on which view, is defined in the *.ASM files. The position of the corresponding values is then defined on the view in the *.CFG file. Connection between the ASM file and CFG file takes place via the ITEM-ID. The numeric range for the ITEM-ID can range between 0 and 65,535.

Default values [Defaults]

DefVisualWidth <natural number>
Description Standard width for the display of a value (total of the widths for designation, value, unit). This value is used, if –1 is specified on the ITEM line in position 3.
Example `defVisualWidth=120`

DefValueWidth <natural number>
Description Standard width for the display of values. This value is used, if –1 is specified on the ITEM line in position 4.
Example `defValueWidth=40`

defUnitWidth <natural number>
Description Standard width for the display of units. This value is used, if –1 is specified on the ITEM line in position 5.
Example `defUnitWidth=20`

defFormat "<font size=Pixel><font face=font type><color=color code> <bgcolor=color code>"
Description Standard values for font size, font type, font and background color. These values are valid, if no other values have been defined in the ASM files for definition of the parameterization or display lines in " ".
<font size> indicates the number of pixel, <font face> the font type, <color> the font color and <bgcolor> background color of an object. It is also possible to specify only some of these formattings as default. Font formattings use current specifics, no special formattings.
Color details occur in a 6-digit hexadecimal number with the following color mix: 2xblue, 2xgreen and 2xred. Values from 0 to F (e.g. 0c0c0c for LIGHT GREY). If the bgcolor is indicated with–1, the objects will be formatted according to a standard key: Description- and object fields with GREY and value fields white background.
Example
```
defFormat=
"<font size=10><font face=Arial><color=0c0c0c><bgcolor=-1>"
```
FAQ With the color formatting as default, the texts of marked lines are no longer displayed inverse so that color overlaps can occur. Therefore it is advisable to define color formattings inline and do not use global details.

Definition of a bit map view [VIEWx]

Example
```
[VIEW1]
NAME="Power plant"
LEVEL=0
TYPE=0
BITMAPFILE=%BITMAP_PATH%\Powerplant.bmp
BITMAP X-POS=0
BITMAP Y-POS=0
BITMAP WIDTH=-1
BITMAP HEIGHT=-1
```

Name "<unique character chain>"
Description Name of the view that is displayed in the selection list of the level bar.

Level **0** ..... (Power plant, left-hand selection list)
1 ..... (Machine level, middle selection list)
Description This value defines in which level the view can be selected.

Type **0** ..... (Bit map)
Description Definition of the display method used.

BitmapFile %BITMAP_PATH%\<file name>.bmp
Description Background mask (bit map) to be used. Specification as an absolute path or with path variables that can be defined in the system settings.

BITMAP X-Pos **0** ..... (top)
BITMAP Y-Pos **0** ..... (left)
Description Position X or Y in which the bit map will be output.

BITMAP Width **-1** ..... (original width)
BITMAP Height **-1** ..... (original height)
Description Width or height with which the bit map is output.

Definition of a table view [VIEWx]

Example
```
[VIEW111]
NAME="Device 1"
LEVEL=2
TYPE=1
PARENT=ENGINE1
```

Name "<unique character chain>"
Description Name of the view that is displayed in the selection list of the level bar.

Level **2** ..... (Sensors-/actuator level, right selective list)
Description This value defines in which level the view can be selected.

Type **1** ..... (table)
Description Definition of the display method used.

PARENT ENGINEx
Description Definition of, to which machine a table view belongs.

Definition of objects of visualization

ITEM<Item-ID> 1;2;3[;4;5;6;n]

Description Position specifications for display of the data objects and buttons. xxxxx is the ITEM-ID with which the object has been created in the ASM file. Depending on the object type various parameters must be specified, separated by ‚**;**'. If the value –1 is specified for one of the values, the standard value as defined in the segment [DEFAULT] is used.

<Item-ID> 0 to 65534 (standard range)

The following are available in: Bitmap view

Data object
- **1**.....Position X of the object
- **2**.....Position Y of the object
- **3**.....Width of the whole object; the remaining range can be used for object description.
- **4**.....Width of range to display the object value.
- **5**.....Width of range to display the object unit.
- **6**.....Height of the object (needn't be specified)

Example
```
ITEM20201=235;90;-1;-1;-1
```

Definite texts
- **1**.....Position X of the object
- **2**.....Position Y of the object
- **3**.....Width of the object
- **4**.....Set object value to zero
- **5**.....Set object unit to zero
- **6**.....Height of the object

Example
```
ITEM20200=235;46;58;0;0;18
```

The following are available in: Bitmap- and table views

Button:
- **1**.....Position X of the object
- **2**.....Position Y of the object
- **3**.....Width of the object
- **4**.....Height of the object

Example
```
ITEM301=60;20;146;25
```

FAQ In the case of buttons no default specifications by means of -1 are permitted.

Bit map alternative (@BILD):
- **1**.....Position X of the object
- **2**.....Position Y of the object
- **3**.....Default, if condition are false
- **4**.....Displayed if condition are true

Example
```
ITEM2006=27;21;%BITMAP_PATH%\Image1.bmp;%BITMAP_PATH%\Image2.bmp
```

Bitmap alternative (@BMPS):
- **1**.....Position X of the object
- **2**.....Position Y of the object
- **3**.....Bmp<Number>, if condition are false
- **4**.....Bmp<Number>, if condition 1 is true
- **n**.....Bmp<Number>, if condition n is true

Example
```
ITEM2006=27;21;0;1
```

Definition of available figures [BitmapList]

Bmp0 "" .....(no figure is displayed)

Bmp<n> %BITMAP_PATH%\<Bildname>.bmp (more defined objects)

Description Generally available objects, which can be linked by the bitmap number of a BMPS syntax.

Example
```
Bmp0=""
Bmp1=%BITMAP_PATH%\Image1.bmp
```


Configuration file (*.asm)

The configuration file is subdivided according to the device functions into the following segments (if available):

- Definition of the values for the views
- Definition of the values for the data logging
- Definition of the configuration data
- Definition of the values for the remote control
- Definition of the values for the event memory

Definition of a view

Definition of the objects for a view is initiated by first of all defining the view for which the ensuing objects are valid. This is effected by the code word **;@VIEW** followed by the view number **x** (plant level), **xx** (machine level) or **xxx** (sensor/actor level). This number provides the reference to the CFG file. Definition of the individual data objects with the introductory syntax then follows: **;@ ...** and of the **asm-ID** for referencing for bit map views and **;& ...** without asm-ID for table views.

Definition of a bitmap view (;@VIEWx)

Example

```
;* Power plant (VIEW 1)
;@VIEW1
;@TEXT 10100, 0, "Generator"
;@DYNU 10101, 1,"U Gen  00000 V" ,4,H'00FF,1.0
;@DYNI 10103, 3,"P Gen  00000 kW",4,H'FF00,0.001
(...further definitions)
;*.......................................... operating mode
;@FKLA 10120,49, "BA:",H'4000,"Stop",H'2000,"Probe",H'1000,"
Manual ",H'0800,"Auto"
;@BILD 10130,10,H'00C0, H'0000 ;*.............. Generator breaker

;@FELI 10140,  0," No alarms ", "--- ALARM ---", H'00C0
;@MORE 10150, 11," Details "

;* Machine level (VIEW 11)
;@VIEW11
;@UNSI 10105,21,"Rounds  0000U/min",1.0
;@INTE 10106,25,"React.power    00000kvar",1.0
;@DOPP 10107,50,"Act. energy", "kWh"
(...further definitions)
;@FELI 10140,  0," No alarms ", "--- ALARM ---", H'00C0
;@MORE 10150,111," display more "
;@MORE 12152,  1," back""
(...further definitions)
```

Definition of a table view (;@VIEWx)

Example

```
;* Sensor/actors (VIEW 111)
;@VIEW111
;&TEXT  0, "------Generator-------"
;&DYNU  1,"U Generator  00000V" ,4,H'00FF,1.0
;&DYNI  3,"P Generator  00000kW",4,H'FF00,0.001
(...further definitions)
;&UNSI 21, "Gn:Rounds  0000U/min",1.0
;&INTE 25, "Gn: Reactive power.    00000kvar",1.0
;&COSP 26, "Gn:CosPhi"
(...further definitions)
```

Definition of display objects

Remark **;***
Description Defines all ensuing line characters not to be heeded by the program.
Example `;* Definition of display values`

Bit map display **;@<Data type> <Item-ID>, <display-ID1>, <parameter list>**
Table display **;&<Data type> <display-ID1>, <parameter list>**

<Item-ID> 0 to 65534 (standard range)
Description The reference number is the number that interconnects the ASM and CFG files. This concerns the visualizing objects.

<display-ID1> 0 to VisValuesCount.value –1
Description Specifies which word of the received protocol is to be processed.

<display-ID2> 0 to VisValuesCount.value –1
but not equal to display-ID1
Description Specifies which word of the received protocol is to be referenced.

"<Text Mask>" "f Generator 00.00Hz" (example describes the generator frequency)
Description The text contains the designation, the mask defines the number of places and the unit of the parameter value. This is placed right-justified on the mask and displayed.
FAQ Correct display of the values is dependent on correct selection of the decimal point, if required.

"<Text1 Value1>" "V Generator: Ok" (example reproduces an error message option)
Description Defines the displayed value, if necessary also the designation, in compliance with a suitable masking.
FAQ Please take heed that correct display for specifications for designation and value between two character sequences requires at least 2 spaces.

"<Text>", "<Unit>" "Running hours", " h" (example: defines designation, unit for double word)
Description The value of a double word is displayed between Text1 and Text2.

<Bit mask> H'8000 (example: of a bit mask, here the highest bit is enabled)
Description Mask in hexadecimal for the bits that can be read out from the word.

<Multiplier> <decimal number>
Description The purpose of the multiplier is the correct display of values that must be adapted in terms of size after machining processes.

<Divisor> <decimal number>
Description The purpose of the divisor is the correct display of values. For more exact processing it divides arriving values that have been internally multiplied by the device..

In the following the data types are described:

Unsigned Integer ;&UNSI <display-ID1>, "<Text Mask>",<Divisor>

Description Display of an unsigned integer (0 to 65.536).

Example
```
;@UNSI 10105,21,"Rounds  0000 U/min",1.0
```

Signed Integer ;&INTE <display-ID1>, "<Text Mask>", <Divisor>

Description Display of a signed integer (-32.767 to 32.767).

Example
```
;@INTE 10106,25,"React. Power  00000 kvar",1.0
```

Unsigned Integer (expanded) ;&UNS_ <display-ID1>, "<Text Mask>", <Bit mask>, <Multiplier>

Description The value is pushed to the right due to the connection of the bit mask with an AND until the mask stops on the right. This value is then multiplied by the multiplier and shown on the mask. The purpose of the type is, e.g., to filter byte values.

Example
```
;@UNS_ 10105 84, "DM1: FMI 1  00",H'FF00,1.0
```

Signed Integer (expanded) ;&INT_ <display-ID1>, "<Text Mask>", <Bit mask>, <Multiplier>

Description The value is pushed to the right due to the connection of the bit mask with an AND until the mask stops on the right. This value is then multiplied by the multiplier and shown on the mask. The purpose of the type is, e.g., to filter sign bearing byte values.

Example
```
;@INT_ 10105 84, "DM1: FMI 1  00",H'FF00,1.0
```

Signed Long ;&DOPP <display-ID1>, <Text>, <Unit>

Description Represents a signed long (-2.147.483.647 to 2.147.483.647) between text and unit.

Example
```
;@DOPP 10107,50,"Act. Energy", "kWh"
```

Signed Long formatted ;&DOP_ <display-ID>, "<Text Mask>", <Multiplier>

Description Represents a signed long (-2147483647 bis 2147483647) in formatted mode.

Example
```
;@DOP_ 10107,50,"Act. Energy  00000 kWh",1.0
```

Scaled value without sign ;&DYNU <display-ID1>, "<Text Mask>", <display-ID2>, <Bit mask>, <Multiplier>

Description The value of display-ID2 is pushed to the right due to the connection of the bit mask with an AND until the mask stops on the right. The result is the exponent with base ten. The whole thing is multiplied by the content of the display-ID1 and a fixed multiplier, rounded and shown as an unsigned integer.

Example
```
;@DYNU 10101, 1,"V Gen  00000 V" ,4,H'00FF,1.0
```

Scaled value with sign ;&DYNI <display-ID1>, "<Text Mask>", <display-ID2>, <Bit mask>, <Multiplier>

Description The value of display-ID2 is pushed to the right due to the connection of the bit mask with an AND until the mask stops on the right. The result is the exponent with base ten. The whole thing is multiplied by the content of the display-ID1 and a fixed multiplier, rounded and shown as a signed integer.

Example
```
;@DYNI 10103, 3,"P Gen  00000 kW",4,H'FF00,0.001
```

Fester Text ;&TEXT <display-ID1>, "<Text>"

Description Display of a text independent off display-ID1.

Example
```
;@TEXT 10100, 0, "Generator"
```

Hexadecimal number ;&HEXA <display-ID1>, "<Text Mask>"

Description Represents a value, directly in the hexadecimal.

Example
```
;@HEXA 10100, 2, "Status  0000 hex"
```

Yes/No alternatives

;&JANE <display-ID1>,<Bit mask>,"<Text1 Value1>“,“<Text2 Value2>“,<Double bit>

Description

Data word is tested for the combination specified in the bit mask. If the condition is true, Text1 is displayed, otherwise Text2. Text1 or can also be defined as an empty string. If double bit=1, testing is on B’10 or B‘01, if double bit=0 testing is on B’11. See example:

Example

```
;&JANE 30, H'00FF,"V Gen.:  Ok", "V Gen.:  missing", 0
```

Bit mask (example)	**H'4**000 **(binary: 0100...)**		**H'C**000 **(binary: 1100...)**			
Value (binary, x = any)	x0xx	x1xx	00xx	01xx	10xx	11xx
Display at double bit=0:	Text2	Text2	Text2	Text2	Text2	***Text1***
Display at double bit=1:	Text2	***Text1***	Text2	Text2	***Text1***	Text2

Text display depending on the bit pattern

;&FKLA <display-ID1>,"<Text>",<H’Mask1>,"<Text1>",....,<H’Maskx>,"<Textx>"

Description

Allows a selection from four texts in compliance with the H’mask. If a valid mask is not available, the data field will be empty.

Example

```
;&FKLA  9, "DI:", H'00C0,"4", H'0030,"3", H'000C,"2", H'0003,"1"
```

Text display depending on a number

;&MULT <display-ID1>, "<Text>", <H’Mask1>, "<Text1>",\
..., <H‘Maskx>, "<Textx>"

Description

Displays a text depending on the number arriving via the bus on display-ID1. The mask can be a hexadecimal number or a number from 2^0 to 2^{16}. The syntax can be typed on more than one row by with ",\" at the end. If a valid mask is not available, the data field will be empty.

Example

```
;@MULT 10121, 69, "",H'00FF,"Synchr. GCB",H'01FF,"Synchr. MCB",\
H'02FF, "Deadbus GCB",H'03FF,"Deadbus MCB",H'04FF,"Start",\
H'1CFF,"Sprinkler",H'1BFF,"Emerg./Sprinkl."
```

Power rate

;&LEIS <display-ID1>, "<Text Mask>", <Multiplier>, <display-ID2>

Description

Represents the value on display-ID1 after multiplication with multiplier and display-ID2.

Example

```
;@LEIS 11109,  5, "Gen. P Set  00000kW", 0.000357142, 6
```

Cosinus Phi

;&COSP <display-ID1>, "<Text>“

Description

Display of the CosPhi in accordance with the following standard:

Example

```
;@COSP 11108, 26, " Gen. Cosinus Phi"
```

Cosinus-Phi (y)	Capacitive		Inductive
Input	k0.01	1.00	i0.01
internal value	-1	100	1
maximum permitted range	-99 to –1	100	1 to 99

Alarm messages

;&FEHL <display-ID1>, <Bit mask>, "<Text1>“, "", <Error no.>, <Double bit>

Description

&FEHL is defined on view level 3. If the result through the bit mask is true (1 with a 1-bit mask and 11 with a 2 bit mask), Text1 is displayed in dialog Current Alarms via @FELI-button with the corresponding double bit. Display within the levels is not possible. If available a information can be linked in an error help file via the error no. (0 and 1 opens the index of an corresponding *.hlp). &FEHL is linked with the button object of @FELI by the same double bit.

Example

```
;&FEHL 58,  H'C000,"Terminal 34  ", "", 5808, H'0003
```

Recording in alarms ;@FELI <item-ID>, <display-ID1>, <Text1>, <Text2>, <Double bit>

Description Generates an alarm button. If a &FEHL bit mask becomes true, Text2 appears on the button, otherwise Text1. Upon pressing the button a window appears with the messages that are current at present. All messages from &FEHL are recorded in the alarm list, the double bit of which fits into the @FELI mask.

Superimposing masks ;@BILD <item-ID>, <display-ID1>, <Bit mask1>, <Bit mask2>

Description Sets a mask in the display depending on the display-ID1 masked by the bit mask. If Bit mask1 is true, bit map2 (in the CFG file on position 2 defined) will be displayed, otherwise bit map1.

Example
```
;@BILD 11122, 10, H'00C0, H'0000 ;* GCB
```
GCB and MCB breaker Bit mask1 activates bit map ON and bit mask2 OFF. If this value is 0 and bit mask1 is false, bit map OFF linked with bit mask2 will be displayed.

Masking bitmaps (extended) ;@BMPS <item-ID>, <display-ID1>, <Bit mask1>, <Bit mask n>

Description Sets a bitmap (picture) in visualization depending on the defined conditions. If bit mask 1 is fulfilled, the bitmap, which is defined in cfg-file on second position is set. If no condition is fulfilled the bitmap of the first position is displayed.

Example
```
;@BMPS 20101 , 77, H'0300, H'0100, H'0200 ;* breaker logic
```

Navigation buttons ;@MORE <item-ID>, <view-no.>, "<Text>"

Description Definition and inscription of a button. The view-no. defines which view is to be displayed when the button is clicked.

Example
```
;@MORE 20100 , 201, "Generator 1"
```

Definition of data logging objects

The values to be displayed in the data logging are initiated by the segment **;@WRIT.** The code word is followed by the individual values that can be selected in data logging. The definition corresponds with that for the displays, but each value has been expanded by a further 2 parameters. These two parameters define the display area on axis Y in the data logging dialog. **0** (minimum) and **800** (maximum) are standard.

General Syntax **;@<Data type>**, <item-ID>, <visu-ID>, <parameter list>, Y-Min, Y-Max

Description
```
;@WRIT
;@UNSI 10007, 2, "Gen. Frequency  00.00Hz",1.0, 0, 80
```

Data types, that are limited in use by data logging

Special data types ;@COSP ...

Data types, that cannot be recorded:

Constant data types ;@TEXT ...

Scaled data types ;@DYNI ...
;@DYNU ...

Bit or Byte oriented data types ;@JANE ...
;@FKLA ...
;@MULT ...
;@BILD ...
;@BMPS ...
;@FEHL ...
;@HEXA ...

Definition of configuration objects

Remark **;***
Description Defines all ensuing line characters not to be heeded by the software tool.
Example `;* Definition of configurable values`

Comment **;!K**
Description Defines all ensuing line characters as visible text in the configuration dialog.
Example `;!K Enter Password before writing`

General configuration syntax **%TAB 0,0,0,<Access>;**!<data type> <para-ID1>,<parameter list>

<Access> **H'01** .....(Read, Bit 0 is set)
H'02 .....(Write, Bit 1 is set)
H'03 .....(Read/write, Bit 0 and 1 is set)
H'10 .....(Exception: write without first reading, Bit 4 set)
Description Defines the write and read rights of a parameter in the configuration dialog.
FAQ Generally is valid: a parameter can only be written, if reading first was possible and if the current password level of the unit permits this.

<para-ID1> 1 to VisValuesStartID.value –1
DataBufferSize.value = VisValuesStartID.value +1
Description Defines the number of the parameter to be processed.

<para-ID2-4> 1 to VisValuesStartID.value –1
not equal with para-ID1
Description Defines the number of the reference parameter to be used.

<Text> "<unique character chain>"
Description Defines the parameter explanation shown in the configuration dialog.

<Text1-n> "<unique character chain>"
Description Defines an alternative text for display the value.

<Time1-2> "<unique character chain>"
Description Defines the displayed terms for delay times indicated in the configuration dialog.

<Mask> "00.00Hz" (example masks the generator target frequency)
Description Defines the number of places and the unit of the parameter value. This is placed right-justified on the mask and displayed.
FAQ Please take heed that a space has to be placed on a one-place mask: " 0".

<H'Mask> H'00F0 (example releases bits 4 to 7)
Description Allows only individual bits of a word to be changed.

<Unit> "kWh" (example defines kWh as a unit of a real power controller: target value fixed)
Description Specifies the unit for double words (long and power rate). These data types do not need any other masking to define the number of places.

<Bit pattern> H'0010 (example allows only bit 4 to be changed/processed)
Description Masking of the data word bit to be processed in hexadecimal.

<Multiplier> 25.6 (example sends value as the product <value>*25.6=<internal value)
Description The parameter values are usually not sent internally with the displayed value, but in a multiple thereof to the device to be parameterized.

<Divisor> 100 (example is suitable for percentages with relative values)
Description Either as a constant specified number or a value referenced via para-ID2.

<Sequence type> 2 (example reads/writes in ascending sequence further Para-Ids)
Description Defines the sequence in which further para IDs belonging to the data type are read or written.

<Alternative number> 128 (example specifies the maximum possible number)
Description Possible text alternatives, the quantity must be an integer (2 to 128).

<Number of characters> 32 (example specifies the maximum possible number)
Description Text length must have an even number of characters (between 2 and 32)

<Min>, <Max> -100, 100
Description Minimum and maximum input value, for hex numbers in hex, otherwise in decimal.

In the following the data types are described:

Unsigned Integer !z <para-ID1>, "<Text>", "<Mask>", <Multiplier>, <Min>, <Max>
Description Display/input of an unsigned integer. (0 to 65.536).
Example
```
%TAB 0,0,0,H'03;!z 66, "Start position", "000%", 655.35, 0, 100
```

Signed Integer !Z <para-ID1>, "<Text>", "<Mask>", <Multiplier>, <Min>, <Max>
Description Display/input of a signed integer (-32.767 to 32.767).
Example
```
%TAB 0,0,0,H'03;!Z 11, "Gen. f set", "00.0Hz", 25.6, 480, 620
```

Percentual integer !N <para-ID1>, "<Text>", "<Mask>", <Multiplier>
Description Display of an integer value, that serves as a reference para-ID. This is always read before all others and should not be writable.
Example
```
%TAB 0,0,0,H'01;!N151, "Apparent power formated","00000", 1
```
FAQ If a parameter uses a reference ID that is not available all parameters could not be read out No read-out is possible at all.

Signed Long !y <para-ID1>, "<Text>", "<Unit>", <Multiplier>, <Min>, <Max>
Description A signed Long number (-2.147.483.647 to 2.147.483.647) is read/written as two consecutive words, first the High- and then the Low-Word.
Example
```
%TAB 0,0,0,H'03;!y3550,"Y-Value 1","00000",1.0,-99999,99999
```

Unsigned Long !I <para-ID1>, "<Text>", "<Unit>", <Multiplier>, <Min>, <Max>
Description An unsigned Long number (0 to 4.294.967.293) is read/written as two consecutive words, first the High- and then the Low-Word.
Example
```
%TAB 0,0,0,H'03;!l1752,"Rated power","00000.0KW",1.0,5,999999
```

Relative value with preceding sign, divisor 100 (percentages)

!P <para-ID1>, "<Text>", "<Mask>",<para-ID2>, <Min>, <Max>

Description: Input of a percentage/readout of a relative value without a preceding sign with a fixed divisor of 100 for percentages of a reference value on para-ID2.

Example:
```
%TAB 0,0,0,H'03;!p 140,"max. act. load hop","00%",240,10,80
%TAB 0,0,0,H'01;!N 240,"Apparent power formated","00000",1
```

Relative value with preceding sign, with selectable divisor

!q <para-ID1>, "<Text>", "<Mask>", <para-ID2>, <Divisor>, <Min>, <Max>

Description: Input of a percentage/readout of a relative value with a preceding sign with a selectable divisor for percentages of a reference value on para-ID2. The parameters are interpreted as having a preceding sign.

Example:
```
See Type !p
```

Relative value without preceding sign, with selectable divisor

!p <para-ID1>, "<Text>", "<Mask>", <para-ID2>, <Divisor>, <Min>, <Max>

Description: Input of a percentage/readout of a relative value with a preceding sign with a selectable divisor for percentages of a reference value on para-ID2. Processing and display is without preceding sign.

Example:
```
%TAB 0,0,0,H'03;!p 10,"Gen. V set","000V",507,400,50,500
%TAB 0,0,0,H'81;!N507,"Gen.Voltage, formated","00000",1
```

Text with fixed length (16 characters)

%TAB 16,0,0,<Access>;!T < para-ID1>, "<Text>"

Description: Input/reading of a text with a maximum of 16 characters.

Example:
```
%TAB16 0,0,H'03;!T836, "Input Terminal 96, Text"
```

Text with variable length

%TAB 16,0,0,<Access>;!t <para-ID1>, "<Text>",<Number of characters>

Description: Input/reading of a text with a maximum of 32 characters, the number of characters is selectable, but may only be an even number (2, 4,..., 30, 32).

Example:
```
%TAB16 0,0,H'03;!t836, "Input Terminal 96, Text", 16
```

Text alternatives

!M <para-ID1>,"<Text>",<Bit pattern>,<Alternative number>,"<Text1>",\
"<Text2>","<Textx>"

Description: Input/reading alternatives possible. Line break with backslash "\" is possible within definition. In accordance with the bit pattern the text1 to Textx are counted up.

Example:
```
%TAB 0,0,0,H'03, "Function terminal 6:",H'0300,3,"Sprinkler",\
"Release Engine","External acknoledgement"
```

Selection Yes/No or Text1/Text2

!B <para-ID1>, "<Text>", <Bit pattern>, "<Text1>", "<Text2>"

Description: Input/reading of a Yes/No decision (default texts) as per bit pattern. No=0 and Yes=1. When Text1 and Text2 are specified, are Text1=0 and Text2=1.

Example:
```
%TAB 0,0,0,H'03;!B248,"Post run by start without CB", H'0020
```

4 bit array

!F <para-ID1>, "<Text>", "<Mask>"

Description: Input/reading of an array of 4 bits. This represent a 4-fold Yes/No decision. There are 4 buttons that can be activated. The button on the right sets bit 12, the button on the left sets bit 15. Bits 0 to 11 always remains unaffected. The inscription of the buttons is controlled via the mask. For example, if the masking sets "JN", an "J" activates and an "N" deactivates in the buttons.

Example:
```
%TAB 0,0,0,H'03;!F127,"DI.1-4 Motor delay (Y/N)","YN"
```

Group of flags !f <ID1>, "<Text>", "<Mask>", <Bit mask>, <Direction>, "<Text1>", ... "<Textx>"

Description All 16 bits of a word can be set individually. Masking is possible for individual bits that are to be processed. The display inscription is controlled via the mask. For example, if the masking sets "JN", an "J" activates and an "N" deactivates in the buttons. Texts 1 to x define the texts that appear next to the control buttons.

Example
```
%TAB 0,0,0,H'03;!f125,"DI 1-4 Function (R/A)","AR",H'F000,1,"DI 1=A","DI 2=A","DI 3=A","DI 4=A"
```

Direction 0 (The LSB, least significant byte, is set with the top control box)
or 1 (The MSB, most significant byte, is set with the top control box)

Hexadecimal number !H <para-ID1>, "<Text>", "<Mask>", <Min>, <Max>

Description Input/Reading of a word as a number between 0000 and FFFF and storage in hex.

Example
```
%TAB 0,0,0,H'03;!H100,"Byte Status","0000",0000,FFFF
```

BCD number !D <para-ID1>, "<Text>", "<Mask>", <Min>, <Max>

Description Input/Reading of a word in the form of a number between 0000 and FFFF and storage in hex. The minimum and maximum input values are thereby checked place by place, each 4 bits with the corresponding input range. Via the input range (Min and Max) in hexadecimal can be defined in which ranges the individual word bits may be changed.

Example
```
%TAB 0,0,0,H'03;!D100,"Byte Status","0000",0000,0FF0
```

Power rate !L <para-ID1>, "<Text>", "0KW", 1, <Min>, <Max>

Description Input of an output value in accordance with the following standard. The bits 0 to 13 are available for the power rate in KW. Bit 14 and 15 define the power rate.

Example
```
%TAB 0,0,0,H'03;!L 69,"Power set point 1","0kW", 1, -6900, 6900
```

Real power set value	Outgoing power (Export)	Fixed value power (Constant)	Reference power (Import)
Input	E<value>	C<value>	I<value>
internal value	<value>	<value> + (32768 / 2)	<value> + 32768
setting Bit 14-15	0x00	0x01	0x10
<Value> (Standard)	0 to 6900	0 to 6900	0 to 6900

Cosinus phi !C <para-ID1>, "<Text>", <Mask>, 1, <Min>, <Max>

Description Input of a CosPhi (power factor) in accordance with the following standard. Min and Max are specified as a negative or positive integer xx, e.g. +/-70. In contrast the display is effected with k0.xx or i0.xx.

Example
```
%TAB 0,0,0,H'03;!C 13,"Power factor","0.00",1, -70,70
```

Cosinus-Phi	Capacitive		Inductive
Input	k0.01	1.00	i0.01
internal value	-1	100	1
maximum permitted range	-99 to -1	100	1 to 99

Relay manager linkage !R <para-ID1>, "<Text>"

Description Input of a relay manager linkage in accordance with the following standard. Always three sequential word are read/written. These include up to three links.

Example
```
%TAB 0,0,0,H'03;!R436,"Assignment 1. relay","000", 1
```

Type of links	OR	AND	NOT	End coding
Input	+	*	-	
internal value	0x00	0x01	0x02	0x04

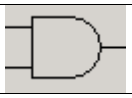
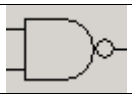
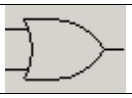
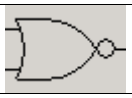
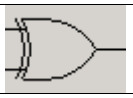
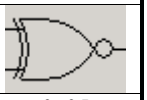
Logic manager !A <para-ID1>, "<Text>", "<Time1>", "<Time2>", "<Mask>", "<unit>", <Sequence type>, <para-ID2>, <para-ID3>, <para-ID4>
!#<para-ID2-4>, <Alternative number>,"<Text1>",\
"<Text2>","<Textx>"

Description Input of a logic manager configuration with two delay times, three values with selectable signs and two links. The value options were logded in tables respective for each selection. Each selection can also be entered in the same table. Idle value options ("") were removed from the selection menu. The configuration will be transmitted in seven sequential words.

Example
```
%TAB 0,0,0,H'03;!A12100, "Relais 1", "Delay ON", "Delay OFF",
"000.00","sec",2,12000,12000,12000
(...)
%TAB 0,0,0,H'01;!#12000,300,"00.01 Flag 1",\
"00.02 Flag 2",\
(...),\
""
```

Parameter (word)	time1	time2	operator1	operator2	value1	value2	value3
Description	Time for pickup delay	Time for drop-out delay	Content of the word is indicated below	Content of the word is indicated below	Value of the first selection	Value of the second selection	Value of the third selection
Sequence type=1	para-ID1+0	para-ID1+1	para-ID1+2	para-ID1+3	para-ID1+4	para-ID1+5	para-ID1+6
Sequenze type=2 (Standard)	para-ID1+6	para-ID1+5	para-ID1+4	para-ID1+3	para-ID1+2	para-ID1+1	para-ID1+0

Sign, unary	NOT value	Value	always „1“	always „0“
Description	The value is transferred negative	The value is looped 1:1	The value is independent of the actual state pass over as „TRUE“.	The value is independent of the actual state pass over as „WRONG”.
Input				
internal value	0x00	0x10	0x20	0x30
Sign to value1	The state will be transmitted in the bits 4 to 7 in operator1.			
Sign to value2	The state will be transmitted in the bits 12 to 15 in operator1			
Sign to value3	The state will be transmitted in the bits 4 to 7 in operator2			

Linkage, binary	AND	NAND	OR	NOR	XOR	NXOR
Description	Logical AND	Logical negatived AND	Logical OR	Logical negatived OR	Exclusive OR	Exclusive negatived OR
Input						
internal value	0x00	0x01	0x02	0x03	0x04	0x05
Linking 1 (value1 with value2)	This status is assigned in the bits 0 to 3 in the operator1.					
Linking2 (value1/2 with value3)	This status is assigned in the bits 8 to 11 in the operator1.					

Definition of remote control objects

The data for the remote control is defined by the code word **%RC**. The number of these elements must not be altered. If the ID for the control word has to be changed, the corresponding entry in the CFG file under **[Data]** will also have to be corrected. The data types correspond with those in configuration.

Example
```
%RC 0,0,0,H'03;!L501, "Active power setpoint","0kW",1,-6900,6900
%RC 0,0,0,H'03;!C502, "Cosinus Phi", "0.00" ,1 ,-71, 71
%RC 0,0,0,H'01;!F503, "Control word", "YN"
```

Definition of event recorder objects

The texts of the event recorder data were firmly defined and linked over an event number with the read out values. Preceded to the definitions of each event type, the start ID has to be indicated as the first address in the unit and additionally the number of events which can be recorded in all.

Start-ID ;=FSP0 <Start-ID>

Description Definition of the start ID for requesting the event memory. Once this value is written, the memory content is output in the device. The value is self resetting after reading.

Example
```
;=FSP0 505
```

Event number ;=FSPL <Number of possible events>

Description Definition of the number of possible or expectable events.

Example
```
;=FSPL 75
```

FAQ If this entry is not or unsuitably specified, reading is not possible.

Normal result ;=FSPA <No.>,“<Name>“

Description Definition of category A. These are stored as a number [1 to about 100] with a time stamp. Output is an error number, the event time and a corresponding designation of the event.

Example
```
;=FSPA 1,"Lube oil level"
```

Continuous event ;=FSPB <No.>,“<Name>“, “<Mask>“

Description Definition of category B. It is stored here how long an event was pending. There is a fixed number of "measuring points" which have one time counter each. The time always counts up when the corresponding event is pending and is reset only by acknowledgement. Specified in the mask are zeros as place markers and, if necessary, a unit separated by a space.

Example
```
;=FSPB 31,"ERROR Lube oilpress.:","0000000 sec"
```

Maximum value ;=FSPC <Nr>,"<Name>", <Multiplier>

Description Definition of category C. A maximum value is read out here, i.e. the maximum value that a measured value has reached after the last acknowledgement and the time point at which this occurred. The read-out value is taken as an integer and multiplied by the multiplier prior to displaying.

Example
```
;=FSPC 31,"Maximal value",0.01
```

Stop event ;=FSPD <No.>,"<Name>“,“<Text1:%4d> <Unit1>...<Text7:%4d> <Unit7>", \
<Multiplicer1>,...,<Multiplicer7>

Description Definition of category D. These events are similar to type A. The number of the error and a time stamp are saved. In addition seven measured values are saved and output at the time point of the event. The values are multiplied by the corresponding multiplier prior to output.

Example
```
;=FSPD 100, "Stopping (100):","Lube oilpress.:%4d bar
Cool.Wa.HT:%4d°C\nCharge B:%4d°C Exhaust A:%4d°C\nExhaust
B:%4d°C Gear oil:%4d bar\nGear oil:%4d°C",\
       1.0,1.0,1.0,1.0,1.0,1.0,1.0
```


Text formatting

Selected text formatting can be specified within the configuration and standard value dialogs and in the bit map views.

Global formatting in the CFG file

This possibility of formatting is described in parameter **defFormat.**

Inline formatting in the ASM file

The following can be used in Bitmap and table views and in configuration dialog

Font color <color=color code> (...) </color>

Description color defines the font color between the relevant start and end tags with the specified color code (2xBLUE, 2xGREEN and 2xRED, 6-place hexadecimal number).

Example
```
%TAB 0,0,0,H'03;!Z1,"<color="ff0000">Hz","<color=00ff00>00.0",..
;@UNSI 130,2,"<color="ff0000">f Gen</color>  00.00Hz",...
```

Bold print (...)

Description b defines the text between the relevant start and end tags as being bold.

Example
```
%TAB 0,0,0,H'03;!Z1,"<b>f Gen","<b>00.0Hz",...
;@UNSI 130,2,"f Gen  <b>00.00Hz</b>",...
```

The following can be used in Bitmap views

Font size <font size= pixel number> (...)

Description font size defines the font size between the relevant start and end tags with the specified size. Only for objects in a bitmap view.

Example
```
;@UNSI 10110,2,"<font size=20>f Gen  00.00Hz</font size>",...
```

Font type <font face=font type> (...)

Description font face defines the font faces between the relevant start and end tags with the specified standard types (no special formats). Only for objects in a bitmap view.

Example
```
;@UNSI 10110,2,"<font face=Arial>f Gen  00.00Hz</font face>",...
```

Background color <bgcolor=color code> (...) </bgcolor>

Description bgcolor defines the background color between the relevant start and end tags with the specified color code (2xBLUE, 2xGREEN and 2xRED, 6-place hexadecimal number). If the bgcolor is specified by –1, the objects are formatted in accordance with a standard code: designation and unit boxes with a gray background and value boxes with a white background.

Example
```
;@UNSI 10110,2,"<bgcolor="ff0000">f Gen  00.00Hz</bgcolor>",...
```

Dynamic configuration of a plant

If necessary, read-in of the CFG and ASM files can be dynamically controlled with the aid of the code words **%IFWERT** or **%IFBIT** and **%ENDIF**, optionally also with an **%ELSE**. As a condition for this, up to 32 option numbers are available, the contents of which can be compared either with a numeric value (IFWERT) or bitwise (IFBIT).
The current values of the options are managed in the CFG file under the entry **[Options]**. They are defined in an additional **OPT** file.
In order that old projects are also still supported when CFG files are read in, the function of the dynamic configuration is available here only if the CFG file is marked accordingly. Marking is effected by the entry **###CFG_VERSION2###,** which must be positioned before the segment [PLANT].

NOTE

Multiple nesting is not allowed and is therefore not processed.

%IfWERT, %IFBIT, %ELSE and %ENDIF have a higher priority as a remark designation for lines with ;{! and ;!} within ASM files.

No application parameters should be defined in accordance with ###CFG_VERSION2###, because this can cause error messages. However, as a rule the functionality is fully retained..

Allowing options in the CFG file

FILE .\Tools\<File name>.opt
Description The path of the option file is specified relatively to the location of the CFG file. This file contains all the necessary specifications for definition of the options.
Example
```
FILE=.\Tools\DynConfig.opt
```

OPT1
0.....(German)
1.....(English)
2.....(Portuguese)
Description Option 1 comprises the value of the language selection of the last dynamic configuration so that when next called the configuration is displayed in this language. This entry is controlled via the register data base.
FAQ This option should be controlled only via %IFWERT statements, because 0 cannot be processed via %IFBIT.

OPT<2 bis 32> <value>
Description Options 2 to 32 comprise the value of the corresponding options in the last dynamic configuration. In this way, when next called, the configuration is created with these option values. These entries are controlled via the OPT file.

Legitimation ###CFG_VERSION2###
Description A dynamic configuration can be created and executed only if this entry is set above the entry [Plant]. Furthermore, this entry enables the use of selected text formatting.
Example
```
###CFG_VERSION2###
[PLANT]
```

Defining options in the OPT file

Remark ;*
Description Defines all ensuing line characters not to be heeded by the software tool.

Constant value (K) %OPT <opt-No.>, K, <Bit mask>
Description Assigned to a specific option number is a defined value which is converted for each dynamic configuration, if the option is used.
Example
```
%OPT  6, K, H'0003
```

Configuration value (P) %OPT < opt-No.>, P, <DeviceNr>, <para-ID>, <Bit mask>
Description A defined configuration ID of a device and a specific value is assigned to a certain option number. This is polled for each dynamic configuration and, if it agrees, implemented.
Example
```
%OPT 12, P, 1, 94, H'000F
```

Visualisierwert (V) %OPT < opt-Nr>, V, <DeviceNr>, <visu-ID>, <Bitmaske>
Description A defined display parameter of a device and a specific value is assigned to a certain option number. This is polled for each dynamic configuration and, if it agrees, converted.
Example
```
%OPT 13, V, 1, 54, H'000F
```

Creating options in the ASM and CFG files

Bit value %IFBIT <opt-No.>, <Bit mask>
Description Via this statement the value is checked bitwise in conformity with the definition in the OPT file. If the result is true, all ensuing lines up to the next %ELSE or %ENDIF are initialized. 0-values cannot be analyzed.
Example
```
%IFBIT 22, H‘8000
;* Asynchrony mode
%ELSE
;* Synchrony mode
%ENDIF
```

Decimal value %IFWERT <opt-No.>, <number>
Description Via this statement the number is compared with the value in accordance with the definition in the OPT file. If the result is true, all ensuing lines up to the next %ELSE or %ENDIF are initialized.
Example
```
%IFWERT 1, 0
;* Deutsch
%ENDIF
%IFWERT 1, 1
;* English
%ENDIF
```

Alternative %ELSE
Description Via this instruction all ensuing lines are initialized up to the next %ENDIF, if the relevant %IFBIT or %IFWERT statement provides the result false.

End %ENDIF
Description Via this statement an %IFBIT, %IFWERT or %ELSE statement is concluded. All ensuing lines up to the next %IFBIT or %IFWERT are always co-initialized.

Annex A.
Service Options

Product Service Options

The following factory options are available for servicing Woodward equipment, based on the standard Woodward Product and Service Warranty (5-01-1205) that is in effect at the time the product is purchased from Woodward or the service is performed. If you are experiencing problems with installation or unsatisfactory performance of an installed system, the following options are available:

- Consult the troubleshooting guide in the manual.
- Contact Woodward technical assistance (see "How to Contact Woodward" later in this chapter) and discuss your problem. In most cases, your problem can be resolved over the phone. If not, you can select which course of action you wish to pursue based on the available services listed in this section.

Returning Equipment For Repair

If a control (or any part of an electronic control) is to be returned to Woodward for repair, please contact Woodward in advance to obtain a Return Authorization Number. When shipping the unit(s), attach a tag with the following information:

- name and location where the control is installed;
- name and phone number of contact person;
- complete Woodward part numbers (P/N) and serial number (S/N);
- description of the problem;
- instructions describing the desired type of repair.

CAUTION

To prevent damage to electronic components caused by improper handling, read and observe the precautions in Woodward manual 82715, *Guide for Handling and Protection of Electronic Controls, Printed Circuit Boards, and Modules.*

How To Contact Woodward

Please contact following address if you have questions or if you want to send a product for repair:

Woodward GmbH
Handwerkstrasse 29
70565 Stuttgart - Germany

Phone: +49 (0) 711 789 54-510 (8:00 – 16:30 German time)
Fax: +49 (0) 711 789 54-101
e-mail: stgt-info@woodward.com

For assistance outside Germany, call one of the international Woodward facilities to obtain the address and phone number of the facility nearest your location where you will be able to get information and service.

You can contact the Woodward Customer Service Department or consult our worldwide directory on Woodward's website (**www.woodward.com**) for the name of your nearest Woodward distributor or service facility. [For worldwide directory information, go to **www.woodward.com/ic/locations**.]

Engineering Services

Woodward Industrial Controls Engineering Services offers the following after-sales support for Woodward products. For these services, you can contact us by telephone, by e-mail, or through the Woodward website.

- Technical support
- Product training
- Field service during commissioning

Technical Support is available through our many worldwide locations, through our authorized distributors, depending on the product. This service can assist you with technical questions or problem solving during normal business hours. Emergency assistance is also available during non-business hours by phoning our toll-free number and stating the urgency of your problem. For technical engineering support, please contact us via our toll-free or local phone numbers, e-mail us, or use our website and reference technical support.

Product Training is available on-site from several of our worldwide facilities, or at your location, depending on the product. This training, conducted by experienced personnel, will assure that you will be able to maintain system reliability and availability. For information concerning training, please contact us via our toll-free or local phone numbers, e-mail us, or use our website and reference ***customer training***.

Field Service engineering on-site support is available, depending on the product and location, from one of many worldwide Woodward offices or authorized distributors. Field engineers are experienced on both Woodward products as well as on much of the non-Woodward equipment with which our products interface. For field service engineering assistance, please contact us via our toll-free or local phone numbers, e-mail us, or use our website and reference ***field service***.

Technical Assistance

If you need to telephone for technical assistance, you will need to provide the following information. Please write it down here before phoning:

Contact

Your company _____

Your name _____

Phone number _____

Fax number _____

Software (see information dialog)

Version **LeoPC1** _____

Description of your problem

_____

_____

_____

_____

_____

_____

_____

_____

_____

Please be sure you have a list of all parameters available.

We appreciate your comments about the content of our publications.

Please send comments to: stgt-documentation@woodward.com

Please include the manual number from the front cover of this publication.

Woodward GmbH
Handwerkstrasse 29 - 70565 Stuttgart - Germany
Telefon +49 (711) 789 54-510 • Fax +49 (711) 789 54-101
stgt-info@woodward.com

Homepage

http://www.woodward.com

Woodward has company-owned plants, subsidiaries, and branches, as well as authorized distributors and other authorized service and sales facilities throughout the world.

Complete address/phone/fax/e-mail information for all locations is available on our website (www.woodward.com).

2013/12/Stuttgart