PMA Prozeß- und Maschinen-Automation GmbH

Industrial Controller KS 40-1, KS 41-1 and KS 42-1

KS 40-1
KS 41-1
KS 42-1
MODBUS

Interfacedescription

MODBUS-protocol

9499 040 63511

Valid from: 02/2005

Explanation of symbols:

General information

General warning

Caution: ESD-sensitive components

MODBUS®
is a registered trademark of the MODBUS-IDA Organization

BluePort® and BlueControl®
are registered trademarks of PMA Prozeß- und Maschinen-Automation GmbH

This is a publication of
PMA Prozeß- und Maschinen Automation
P.O. Box 310229
D-34058 Kassel
Germany

Content

1 General

We thank you for purchasing a device from the *BluePort*® product range. This document describes the implementation and operation of the MODBUS interface used with the industrial controller KS 40-1 which will be called 'device' in the rest of this document.
This document is also valid for KS 41-1 and KS 42-1.

Devices with a MODBUS interface permit the transmission of process data, parameters, and configuration data. Electrical connections are made at the base of the device in the channel of the top-hat DIN rail. The serial communication interface provides a simple link to superordinate PLCs, visualization tools, etc.

An additional interface that is always fitted in the device's front panel is the BluePort® (PC) interface. This interface is not bussable, and serves for a direct connection with the BlueControl® software package that runs on a PC or laptop. Communication is done according to the master/slave principle. The device is always operated as a slave.

The most important characteristics and physical/electrical properties of the bus connection are:

- **Network topology**
  linear bus, possible with bus termination at both ends (see below).
- **Transmission media**
  screened and twisted 2-wire copper leads
- **Lead lengths (without repeater)**
  A maximum lead length of 1000 m should not be exceeded.
- **Transmission speeds**
  The following transmission speeds are supported:
  2400 … 38400 bits/s
- **Physical interface**
  RS 485 with bus connections in the top-hat rail; connections made on site.
- **Address range**
  1 ... 247
  (32 devices in one segment. Expandable to 247 with repeaters.)

1.1 References

Further information on the MODBUS-Protokoll:

[1] MODBUS Specifications
- MODBUS application Protocol Specification V1,1
- MODBUS over serial line specification and implementation guide V1.1
- http://www.modbus.org

Further information on RS 485:

[2] ANSI/TIA/EIA-485-A

Additional documentation for KS 4x-1 devices:

[3] Industrial controller KS 40-1 / KS 41-1 / KS 42-1
- Data sheet KS 40-1/41-1/42-1 9498 737 39913
- Operating instructions KS 40-1/41-1/42-1 9499 040 62711

2 Commissioning the interface

Instrument field bus connection is via the pins of connector B on the rear, via flat-pin connectors or via screw terminals dependent on version.
Construction of suitable cables must be done by the user.

2.1 Mounting hints

If possible, the place of installation should be exempt of vibration, aggressive media (e.g. acid, lye), liquid, dust or aerosol.

The unit may be operated only in environments for which it is suitable due to its protection type.

The housing ventilation slots must not be covered.

In plants where transient voltage peaks are susceptible to occur, the instruments must be equipped with additional protective filters or voltage limiters!

Caution! The instrument contains electrostatically sensitive components.

Please, follow the instructions given in the safety hints.

2.2 Electrical connections

The electrical connection of the interface can be done as two-wire RS 485, as well as four-wire RS 485 (often called RS 422).

2.2.1 RS 485 version (two-wire)

The bus is build as RS 485 - two-wire cable with common ground main.
All the participants of an RS 485 bus are connected in parallel to the signals 'Data A' and 'Data B'.

The meaning of the data line terms are defined in the unit as follows:

- for signal 1 (off) Data A is positive to Data B
- for signal 0 (on) Data A is negative to Data B

The terms Data A and Data B are reverse to A und B defined in [2] .

For the purpose of limiting ground current loops, signal ground (GND) can be grounded at one point via a resistor 'RGND' (100 ohms, ¼ watt).

Association of terms for the two-wire-MODBUS definition according to [1]:

Definition MODBUS	according to unit
D1	Data A
D0	Data B
Common	RGND

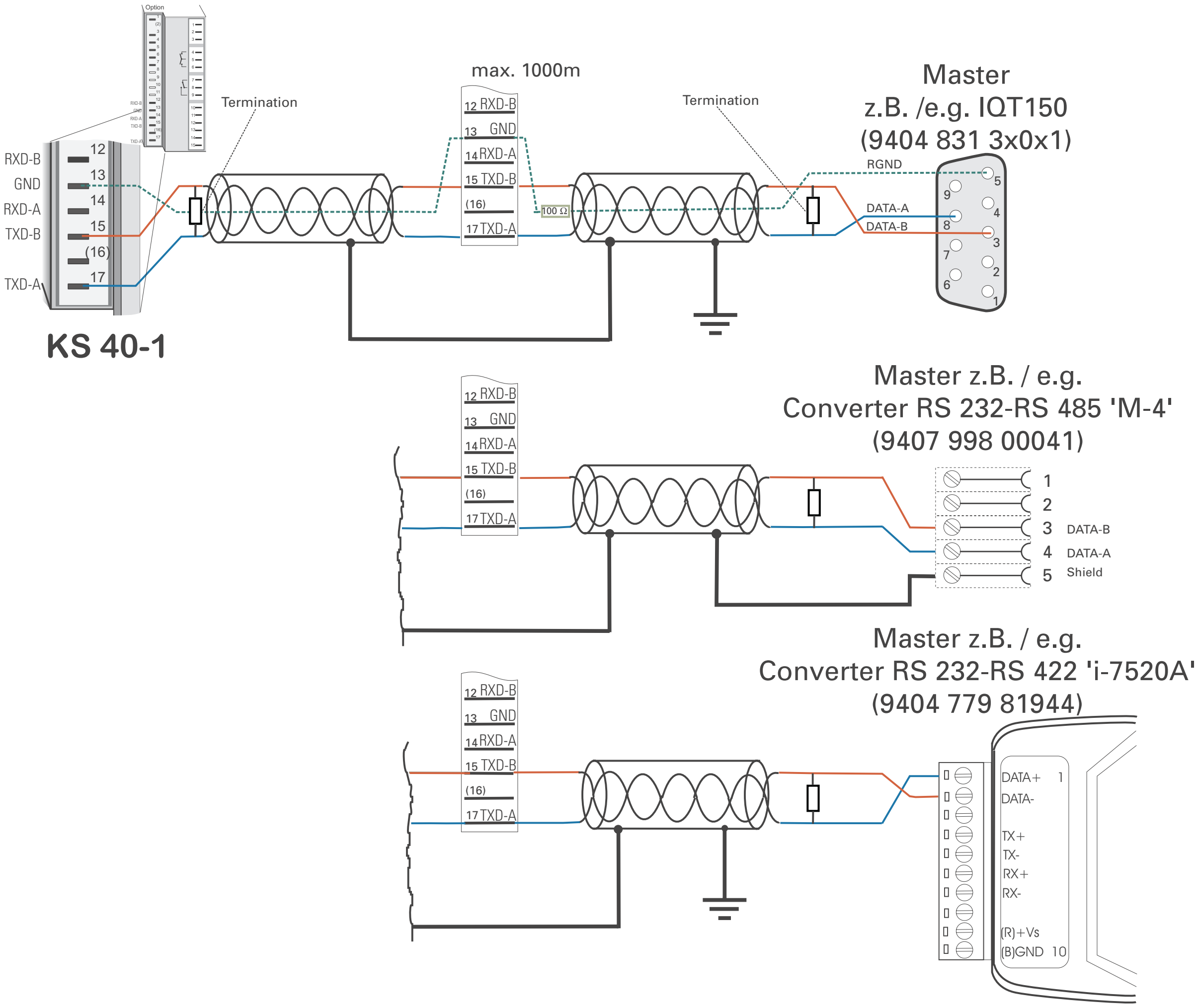
Notes:

❶ Terminating resistors between Data A and B at the cable ends (see 2.2.3 below)
❷ Screening (see 2.2.2 below)
❸ GND lead (see Fig. 6)

KS40-1		IQT 150		M-4		ADAM-4520-D	
Signal	Terminal	Signal	Terminal	Signal	Terminal	Signal	Terminal
TXD-B	15	DATA-B	3	TXD-A	3	DATA -	
TXD-A	17	DATA-A	8	TXD-B	4	DATA+	
GND	13	RGND	5	Shield	5		

There are various possibilities for cable entry of the RS 485.

Fig. 1 : connection example four-wire RS 485 (RS 422)



2.2.2 RS 422 version (four-wire - RS 485)

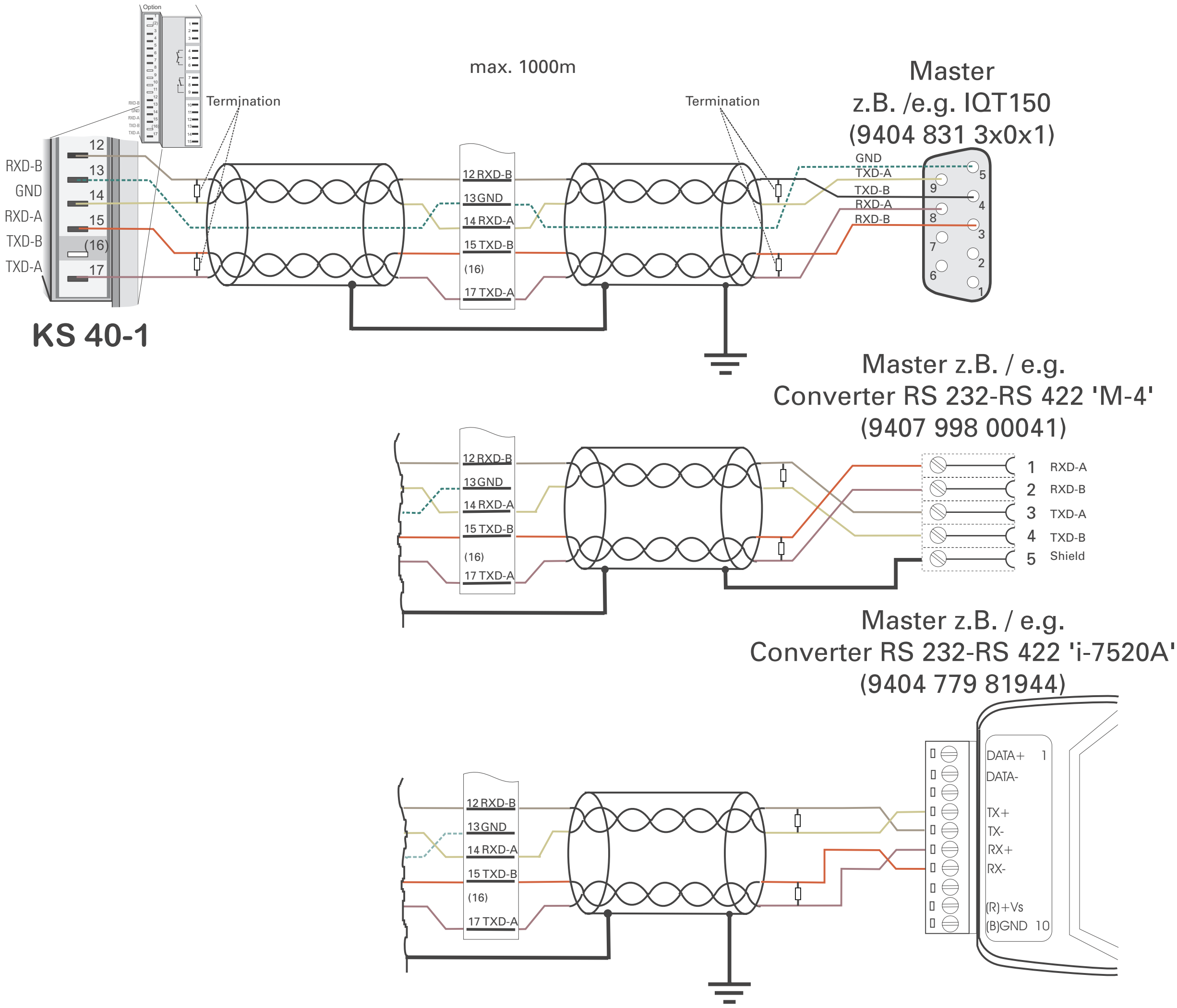
The RS 422 bus is of the RS 485 four-wire type with two pairs of conductors and a common ground.
The data on the master wire pair (RXD) are received only by the slaves. The data on the slave wire pair (TXD) are received only by the master.
Allocation of descriptions for the four-wire MODBUS definition according to [1]:

Description MODBUS	correspondence in the instrument
TXD1	RXD-A
TXD0	RXD-B
RXD1	TXD-A
RXD0	TXD-B
Common	GND

KS40-1		IQT 150		M-4		ADAM-4520-A	
Signal	Terminal	Signal	Terminal	Signal	Terminal	Signal	Terminal
TXD-B	15	RXD-B	3	RXD-A	1	RX-	
TXD-A	17	RXD-A	8	RXD-B	2	RX+	
RXD-B	12	TXD-B	4	TXD-A	3	TX-	
RXD-A	14	TXD-A	9	TXD-B	4	TX+	
GND	13	GND	5	Shield	5		

The following cable connection methods are possible.

Fig. 2 connection example RS 485



2.2.3 Cable installation

Depending on each application, suitable cables are to be used for the bus. When installing the cables, all relevant regulations and safety codes (e.g. VDE 0100) must be observed:

- Cable runs inside buildings (inside and outside of control cabinets)
- Cable runs outside buildings
- Potential balancing conductors
- Screening of cables
- Measures against electrical interference
- Length of spur lines

In particular, the following points must be considered:

- The RS 485 bus technology used here permits up to 32 devices in a segment to be connected to one bus cable. Several segments can be coupled by means of repeaters.
- The bus topology is to be designed as a line with up to 1000 m length per segment. Extensions by means of repeaters are permitted.
- The bus cable is to be taken from device to device (daisy chaining), i.e. not star connected.
- If possible, spur lines should be avoided, in order to prevent reflections and the associated disturbances in communication.
- The general notes on interference-free wiring of signal and bus leads are to be observed (see Operating notes "EMC – General information' (9407 047 09118)).
- To increase signal transmission reliability, we recommend using screened, twisted pairs for the bus leads.

2.2.4 Screening

The type of screening is determined primarily by the nature of the expected interference.

- For the suppression of electrical fields, one end of the screened cable must be grounded. This should always be done as the first measure.
- Interference due to alternating magnetic fields can only be suppressed, if the screened cable is grounded at both ends. However, this can lead to ground current earth loops: galvanic disturbance along the reference potential lead can interfere with the useful signal, and the screening effect is reduced.
- If several devices are linked to a single bus, the screen must be connected at each device, e.g. by means of screen clamps.
- The bus screen must be connected to a central PE point, using short, low-impedance connections with a large surface, e.g. by means of screen clamps.

2.2.5 Terminating resistors

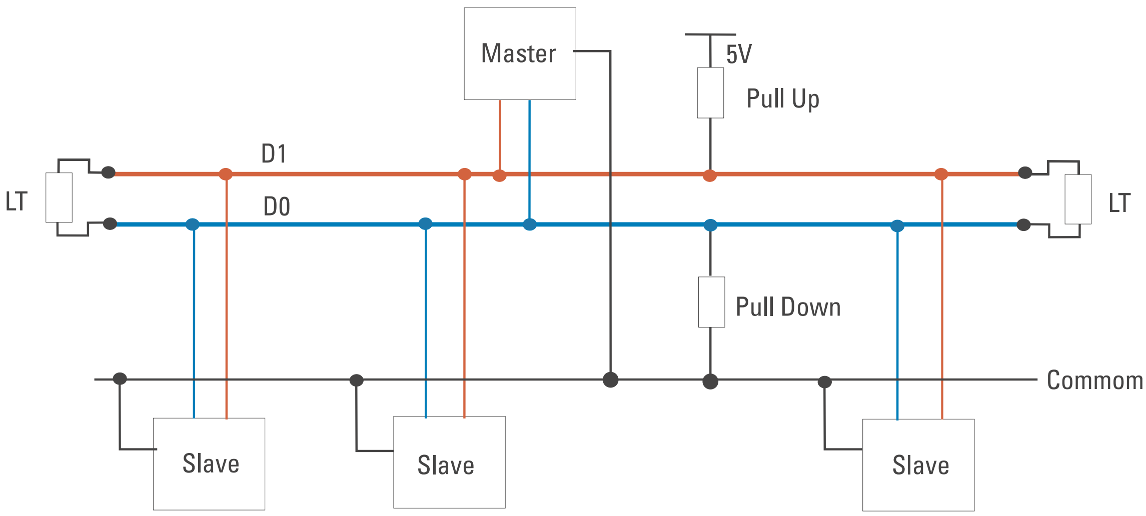
The widespread US Standard EIA RS 485 recommends fitting terminating resistors at each end of the bus cable. Terminating resistors usually have a value of approx. 120 ohms, and are connected in parallel between the data lines A and B (depending on the cable impedance; for details, see the cable manufacturer's data sheet). Their purpose is to eliminate reflections at the end of the leads, thus obtaining a good transmission quality. Termination becomes more important, the higher the transmission speed is, and the longer the bus leads are.

However, if no signals are applied to the bus, it must be ensured that the signal levels are clearly defined. This done by means of pull-up and pull-down resistors between +5V or GND, and the drivers. Together with the bus terminating resistor, this forms a voltage divider. Moreover, it must be ensured that there is a voltage difference of at least ±200mV between the data lines A and B, as seen by the receiver.

Normally, an external voltage source is provided.

Fig. 6 shows the device connections as recommended by the MODBUS User Organization [1].

Fig. 3 Recommended connections



With four-wire connection (RS 422), each wire pair corresponds to the drawing above.

If no external voltage source is available, and if there are only a few participants on the bus (e.g. only a master and a slave device), and the transmission speed is low (e.g. 9600 bits/s), the lead lengths are short, and terminating resistors have been fitted, it is possible that the minimum signal level cannot be reached. This will cause disturbances in signal transmission.

Therefore, if only a few PMA devices are connected, we recommend the following procedure before fitting terminating resistors:

Baudrate	Lead length	No. of PMA devices	Terminating resistor
≤ 9600 Bist/s	≤ 1000 m	< 8	no
19200 Bit/s	≤ 500 m	< 8	no
38400 Bit/s	≤ 250 m	< 8	no
beliebig		≥ 8	useful
			other cases: try out

If less than 8 PMA devices are connected to a bus with the above maximum lead lengths, no terminating resistors should be fitted.

Note: If additional devices from other manufacturers are connected to the bus, no general recommendations are possible – this means: trial and error!

2.2.6 Installation notes

- Measurement and data leads should be kept separate from control leads and power cables.
- Twisted and screened cables should be used to connect sensor. The screen must be grounded.
- Connected contactors, relays, motors, etc. should be fitted with RC snubber circuits in accordance with manufacturer specifications.
- The device must not be installed near powerful electrical or electromagnetic fields.

- The device is not certified for installation in explosion-hazarded areas.
- Incorrect electrical connections can result in severe damage to the device.
- Please observe all safety instructions.

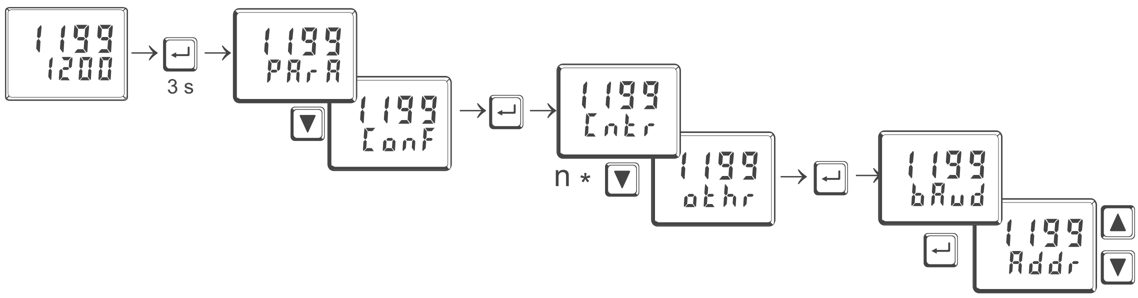
2.3 Bus settings

2.3.1 Bus address

The participant address of a device connected to a bus must be adjusted by one of the following means:

- the Engineering Tool BlueControl® using the menu item Othr/Addr
- or via the device's front panel (see below)

Fig. 4 Setting a bus address



Every device connected to a bus must have a different, unique address.

Please regard: When allocating the unit's addresses don't give the same address to two units. In this case a strange behaviour of the whole bus becomes possible and the busmaster will not be able to communicate with the connected slave-units.

2.3.2 Transmission parameters

The transmission parameters of all devices linked to a bus must have the same settings.

Baudrate (bAud)
The baudrate is the measure of data transmission speed. The devices support the following transmission speeds:

- 38000 bits/s
- 19200 bits/s
- 9600 bits/s
- 4800 bits/s
- **2400 bits/s**

Parity / Stop bit (PrtY)
The parity bit is used to check whether an individual fault has occurred within a byte during transmission.

The device supports:

- **even parity**
- **odd parity**
- **no parity**

With even parity, the parity bit is adjusted so that the sum of the set bits in the 8 data bits and the parity bit result in an even number. Conversely, the same applies for uneven parity.

If a parity error is detected upon receipt of a message, the receiving device will not generate an answer.

Other parameters are:

- 8 data bits
- 1 start bit
- 1 stop bit
  1 or 2 stop bits can be selected when adjusting 'no parity'.

The max. length of a message may not exceed 256 bytes.

2.4 System layout

Please observe the guidelines and notes provided by the manufacturer of the master device regarding the layout of a communication system.

2.4.1 Minimum configuration of a MODBUS installation

A MODBUS installation consists of not less than the following components:

- a bus master, which controls the data traffic
- one or more slave participants, which provide data upon demand by the master
- the transmission media, consisting of the bus cable and bus connectors to link the individual participants, plus a bus segment (or several, which are connected by means of repeaters).

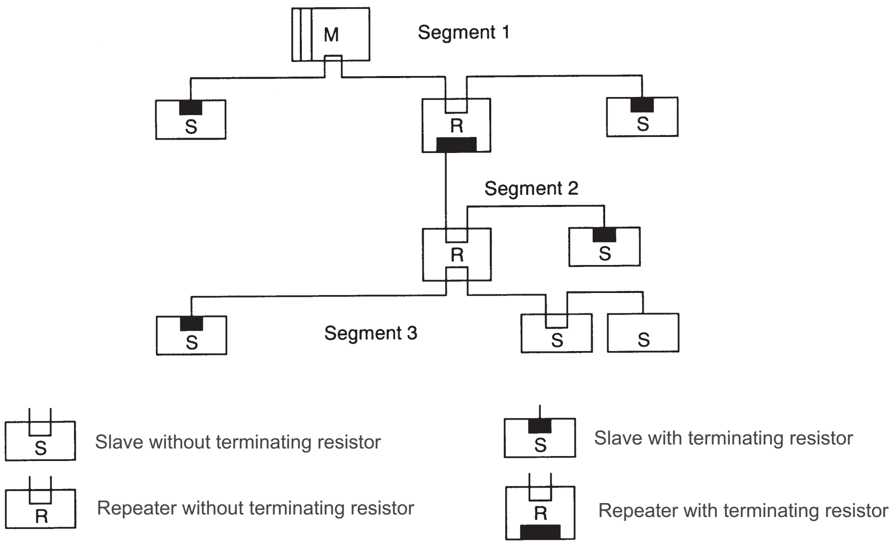
2.4.2 Maximum configuration of a MODBUS installation

A bus segment consists of max. 32 field units (active and passive). The greatest number of slave participants that can be operated by one MODBUS master via several segments, is determined by the internal memory structure of the master. Therefore, you should know the specifications of the master when planning a MODBUS installation.
The bus cable can be opened at any point in order to add another participant by means of a bus connector. At the end of a segment, the bus cable can be extended up to the total permissible length for a segment. The permissible length of a bus segment depends on the selected transmission speed, which in turn is determined mainly by plant layout (length of each segment, distributed inputs/outputs) and the required scan cycles for individual participants. All participants connected to the bus must be configured for the same transmission speed (bit rate).

MODBUS devices must be connected in a line structure.

If more than 32 participants are required, or larger distances than the permissible length of one segment are needed, the MODBUS installation can be extended by means of repeaters.

Fig. 5 structure



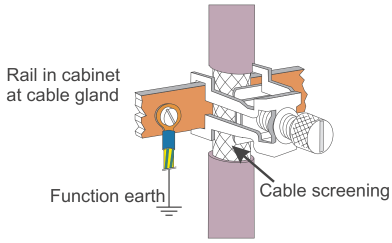
A fully configured MODBUS installation may contain max. 247 participants with the address range 1...247. Every installed repeater reduces the max. number of participants with a segment. Repeaters are passive participants and do not require a MODBUS address. However, its input circuit represents an additional load in the segment due to the current consumption of the bus driver. Nonetheless, a repeater has no influence on the total number of participants connected to the bus. The maximum number of series-connected repeaters can differ, depending on the manufacturer. Therefore, you should ask the manufacturer about possible limitations when planning a MODBUS installation.

2.4.3 Wiring inside buildings

The following wiring hints apply for twisted-pair cables with screen. The cable screen serves to improve overall electromagnetic compatibility.

Depending on requirements, the one or both ends of the cable screen must be connected to a central earth point (PE) by means of low-impedance connections with a large surface, e.g. screen clamps. When installing a repeater or field unit in a control cabinet, the cable screen should be connected to an earth rail mounted as close as possible to the cable entry into the cabinet.

Fig. 6 Screen connection



The screen must be taken right up to the field unit, where it is to be connected to the conductive housing and/or the metal connector. Hereby, it must be ensured that the device housing (and possibly the control cabinet in which the device is installed), are held at equal ground potential by means of low-impedance connections with a large surface. Connecting a screen to a lacquered or painted surface is useless. By observing these measures, high-frequency interference will be grounded reliably via the cable screens. Should external interference voltages still reach the data lines, the voltage potential will be raised symmetrically on both lines, so that in general, no destructive voltage differences can arise. Normally, a shift of the ground potential by several volts will not have an effect on reliable data transmission. If higher voltages are to be expected, a potential balancing conductor with a minimum cross-section of 10 mm^2 should be installed parallel to the bus cable, with connections to the reference ground of every field unit. In case of extreme interference, the bus cable can be installed in a metal conduit or channel. The conduit tube or the channel must be earthed at regular distances.

The bus cable must always be installed with a minimum separation of 20 cm from other cables carrying voltages above 60 V. Similarly, the bus cable must be run separately from telephone lines, as well as from cables leading into explosion-hazarded areas. In these cases, we recommend installing the bus cable in a separate cable tray or channel.

Cable trays or channels should always be made of conductive materials, and must be earthed at regular distances. Bus cables should not be subjected to any mechanical strains or obvious risks of damage. If this cannot be ensured, suitable measures must be undertaken, such as installation in conduit.

Floating installation:
If the installation must be floating (no earth connection) for certain reasons, the device reference ground must only have a high-impedance connection to earth (e.g. an RC combination). The system will then find its own earth potential. When connecting repeaters for the purpose of linking two bus segments, a floating installation is recommended, to prevent possible potential differences being transferred from one segment to the next.

3 Bus protocol

3.1 Composition of a transmission byte

Originally, the MODBUS protocol was defined for the communication between a supervisory system and the Modicon® PLC. It used a master/slave structure, in which only one device (master) is able to initiate data transactions (queries). The query message from the master is answered (response) by other devices (slaves), which supply the requested data. Moreover, the master can address a specific slave via its MODBUS address, or address all connected slaves by means of a general message (broadcast).
The MODBUS protocol determines the transmission formats for the query and the response. Function codes define the actions to be executed by the slaves.
Within the device, the MODBUS protocol uses the RTU (remote terminal unit) mode, i.e. every transmitted byte of a message contains two hexadecimal characters (0...9, A...F).

The composition of a byte in the RTU-protocol is as follows:

Start bit	8 data bits	Parity/Stop bit	Stop bit

3.2 General message frame

The message is read into a data buffer with a defined maximum length. Longer messages are not accepted, i.e. the device does not answer.

The message consist of the following elements:

Device address	Function code	Data field	CRC	End of frame detection
1 byte	1 byte	N * 1 bytes	2 bytes	

- Device address (Addr)
  The device address is used for identification. Device addresses can be assigned in the range of 1...127.
  The device address '0' is reserved for 'Broadcast' messages to all slaves. A broadcast message can be transmitted e.g. with a write instruction that is then executed by all the slaves on the bus. Because all the slaves execute the instruction, no response messages are generated.

- Function code
  The function code defines the transaction type in a message. The MODBUS specification defines more than 17 different function codes. Supported codes are described in Section 3.6. „Function codes".

- Data field
  The data field contains the detailed specifications of the transaction defined by the function code. The length of the data field depends on the function code.

- CRC
  As a further means of fault detection (in addition to parity bit detection) a 16-bit cyclical redundancy check (CRC) is performed. The CRC code ensures that communication errors are detected. For additional information, see Section 3.2.1. "CRC".

- End of frame detection
  The end of a message is defined by a period of 3,5 characters, during which no data transfer occurs. For additional information, see Section 3.2.2. „End of frame detection"

Further information is given in the documents named in **[1]** or under http://www.modbus.org.

3.2.1 CRC

The CRC is a 16-bit value that is attached to the message. It serves to determine whether a transmitted message has been received without errors. Together with the parity check, this should detect all possible communication errors.

ⓘ If a parity fault is detected during reading, no response message will be generated.

The algorithm for generating a CRC is as follows:

① Load CRC register with FFFFhex.

② Exclusive OR the first transmit/receive byte with the low-order byte of the CRC register, putting the result into the CRC register, zero-filling the MSB.

③ Shift the CRC register one bit to the right.

④ If the expelled bit is a '0' repeat step 3.
If the expelled bit is a '1', exclusive OR the CRC register with value A001hex.

⑤ Repeat steps 3 and 4 for the other 7 data bits.

⑥ Repeat steps 2 to 5 for all further transmit/receive bytes.

⑦ Attach the result of the CRC register to the message (low-order byte first, then the high-order byte).
When checking a received message, the CRC register will return '0', when the message including the CRC is processed.

3.2.2 End of frame detection

The end of a message (frame) is defined as a silence period of 3.5 characters on the MODBUS.
A slave may not start its response, and a master may not start a new transmission before this time has elapsed.

However, the evaluation of a message may begin, if a silence period of more than 1.5 characters occurs on the MODBUS. But the response may not start before 3,5 characters of silence.

3.3 Transmission principles

Two transmission modes are used with MODBUS:

- **Unicast mode**
- **Broadcast mode**

In the Unicast mode, the master addresses an individual device, which processes the received message and generates a response. The device address can be 1...247. Messages always consist of a query (request) and an answer (response). If no response is read within a defined time, a timeout error is generated.

In the Broadcast mode, the master sends a write instruction (request) to all participants on the bus, but no responses are generated. The address '0' is reserved for broadcast messages.

3.4 Response delay (dELY)

Some devices require a certain period to switch from transmit to receive. The adjusted delay is added to the silent period of 3,5 characters at the end of a message, before a response is generated. The delay is set in ms.

3.5 Modem operation (C.dEL)

The end of frame detection of a received MODBUS message can be increased by the period 'C.del'. This time is needed e.g. for transmission via a modem, if messages cannot be transmitted continuously (synchronous operation). The delay is set in ms.

3.6 Function codes

Function codes serve to execute instructions. The device supports the following function codes:

Function code hex	dez	Description	Explanation
0x03	3	Read Holding (Output) Register	Reading of process data, parameters, and configuration data
0x04	4	Read Input Register	Reading of process data, parameters, and configuration data
0x06	6	Preset Single Register (Output)	Wordwise writing of a value (process value, parameter, or configuration data)
0x08	8	Diagnostics	Reading the MODBUS diagnostic register
0x10	16	Preset Multiple Register (Output)	Wordwise writing of several values (process data, parameter or configuration data)

The behaviour of function codes 3 and 4 is identical.
The following sections show various examples of message composition.

3.6.1 Reading several values

Messages with function codes 3 or 4 are used for (wordwise) reading of process data, parameters or configuration data. For reading 'Float' type data, 2 values must be requested for each datum.

The composition of a read message is as follows:
Request:

Field name	Value (hex)	Explanation
Address	11	Address 17
Function	03 or 04	Reading process data, parameters or configuration data
Start address High Start address Low	02 8A	Starting address 650
No. of values	00 02	2 datums (2 words)
CRC	CRC-Byte1 CRC-Byte2	

Response:

Field name	Value (hex)	Explanation
Address	11	Address 17
Function	03 oder 04	Reading process data, parameters or configuration data
No. of bytes	04	4 data bytes are transmitted
Word 1	00 DE	Process data, parameters or configuration data. Address 650= 222
Word 2	01 4D	Process data, parameters or configuration data. Address 651= 333
CRC	CRC-byte1 CRC-byte2	

A broadcast message is not possible for function codes 3 and 4.

If the first addressed value is not defined, an error message "ILLEGAL DATA ADDRESS" is generated.
If no further data are defined in the areas to be read following the first value, these areas will be entered with the value "NOT DEFINED VALUE". This enables areas with gaps to be to be read in a message.

3.6.2 Writing a single value

Messages with function code 6 are used for (wordwise) writing of process data, parameters or configuration data as integers. This function is not suitable for writing 'Float' type data.

The composition of a write message is as follows:
Request:

Field name	Value (hex)	Explanation
Address	11	Address 17
Function	06	Writing a single value (process data, parameter or configuration)
Write address High Write address Low	02 8A	Write address 650
Value	00 7B	Preset value = 123
CRC	CRC-byte1 CRC-byte2	

Response:

Field name	Value (hex)	Explanation
Address	11	Address 17
Function	06	Writing a single datum (process data, parameter or configuration)
Write address High Write address Low	02 8A	Write address 650
Value	00 7B	Preset value = 123
CRC	CRC-Byte1 CRC-Byte2	

If everything is correct, the response message corresponds exactly to the default.

The devices can also receive this message as a broadcast with the address '0'.

A default value in the 'Real' data format is not possible, as only 2 bytes can be transmitted as value.

If a value is outside the adjustable range, the error message "ILLEGAL DATA VALUE" is generated. The datum remains unchanged. Also if the datum cannot be written (e.g. configuration data, and the device is online), an error message "ILLEGAL DATA VALUE" is generated.

3.7 Writing several values

Messages with function code 16 are used for (wordwise) writing of process data, parameters or configuration data. For writing 'Float' type data, 2 values must be transmitted for each datum.

The composition of a write message is as follows:
Request:

Field name	Value (hex)	Explanation
Address	11	Address 17
Function	10	Writing several process values, parameters or configuration data
Start address High Start address Low	02 8A	Write address 650
No. of values	00 02	2 values
No. of bytes	04	4 data bytes are transmitted
Word 1	00 DE	Process value, parameters or configuration data. Address 650 = 222
Word 2	01 4D	Process value, parameters or configuration data. Address 651 = 333
CRC	CRC byte1 CRC byte2	

Response:

Field name	Value (hex)	Explanation
Address	11	Address 17
Function	10	Writing several process values, parameters or configuration data
Start address High Start address Low	02 8A	Write address 650
No. of values	00 02	2 process values, parameters or configuration data
CRC	CRC byte1 CRC byte2	

The devices can also receive this message as a broadcast with the address '0'.

If the first value is not defined, an error message "ILLEGAL DATA ADDRESS" is generated.
If the first value cannot be written (e.g. configuration data, and the device is online), an error message "ILLEGAL DATA VALUE" is generated.

If no further data are defined or cannot be written in the specified areas following the first value, these areas will be skipped. The data in these locations remains unchanged. This enables areas with gaps, or that are currently not writable, to be changed with a message. No error message is generated.

If a value is outside the adjustable range, the error message "ILLEGAL DATA VALUE" is generated. Subsequent data are not evaluated. Previously accepted correct data are active.

3.8 Error record

An error record is generated, if a message is received correctly, but message interpretation or the modification of a datum is not possible.

If a transmission error is detected, no response is generated. The master must retransmit the message.

Detected transmission errors are:

- Parity fault
- Framing error (no stop bit received)
- Overrun error (receiving buffer has overflowed or data could not be retrieved quickly enough from the UART)
- CRC error

The composition of the error record is as follows:

Field name	Value	Explanation
Address	11	Address 17
Function	90	Error record for the message 'Writing several parameters or configuration data'. Composition: 80_{hex} + function code
Error code	02	ILLEGAL DATA ADDRESS
CRC	CRC byte1 CRC byte2	

In the 'Function' field, the most significant bit is set.
The error code is transmitted in the subsequent byte.

3.8.1 Error codes

The following error codes are defined:

Code	Name	Explanation
01	ILLEGAL FUNCTION	The received function code is not defined in the device.
02	ILLEGAL DATA ADDRESS	The received address is not defined in the device, or the value may not be written (read only). If several data are read simultaneously (function codes 01, 03, 04) or written simultaneously (function codes 0F, 10), this error is only generated if the first datum is not defined.
03	ILLEGAL DATA VALUE	The received value is outside the adjusted limits or it cannot be written at present (device is not in the configuration mode). If several data are written simultaneously (function codes 0F, 10), this error is only generated if the first datum cannot be written.
04	SLAVE DEVICE FAILURE	More values are requested than permitted by the transmission buffer.

Other error codes specified in the MODBUS protocol are not supported.

3.9 Diagnosis

By means of the diagnosis message, the device can be prompted to send check messages, go into operational states, output counter values or to reset the counters.
This message can never be sent as a broadcast message.

The following functions have been defined:

Code	Explanation
0x00	Return transmission of the received message
0x01	Restart of communication (terminates the Listen Only mode)
0x02	Return transmission of the diagnosis register
0x04	Change to the Listen Only mode
0x0A	Delete the counter and reset the diagnosis register
0x0B	Return transmission of the message counter (all messages on the bus)
0x0C	Reset of the counter for faulty message transmissions to this slave (parity or CRC error)
0x0D	Return transmission of the counter for messages answered with error code
0x0E	Return transmission of the message counter for this slave
0x0F	Return transmission of the counter for unanswered messages
0x10	Return transmission of the counter for messages answered with NAK
0x11	Return transmission of the counter for messages answered with Busy
0x12	Return transmission of the counter for too long messages
0x40	Return transmission of the parity error counter
0x41	Return transmission of the framing error counter (stop bit not detected)
0x42	Return transmission of the counter for full buffer (message longer than receiving buffer)

- ❑ Request in the Integer format:
  If the setting for Integer with decimals (most significant 3 bits) is used for the address, the counter contents will be transmitted in accordance with the necessary conversion factor.

- ❑ Request in the Float format:
  If the setting for Float (most significant 3 bits are 010) is used for the address, the counter contents will be transmitted in the IEEE format. The largest value is 65535, because the counters in the device are designed as word counters.
  In the Float format, a 4-byte data field is returned with a request for counter contents. In all other cases, a 2-byte data field is returned.

When switching into the Listen mode (0x04) and at restart after the device has changed into the Listen mode, no response is generated.
If a restart diagnosis message is received while the device is not in the Listen mode, the device generates a response.

A diagnosis message is composed as follows:

Request:

Field name	Value	Explanation
Address	11	Address 17
Function	08	Diagnosis message
Sub-function High Sub-function Low	00 YY	Sub-function code
Data field	Byte 1 Byte 2	Further data definitions
CRC	CRC byte1 CRC byte2	

3.9.1 Return transmission of the received message (0x00)

The message serves as a check whether communication is operational.
Definition of the received and returned data:

Sub-function	Received data field	Transmitted data field
00 00	2 bytes of any content	Return transmission of the received datum

3.9.2 Restart of communication (terminates the Listen Only mode) (0x01)

The slave is instructed to initialize its interface, and to delete the event counters. In addition, the device is instructed to exit the Listen Only mode. If the device already is in the Listen Only mode, no response is generated.
Definition of the received and returned data:

Sub-function	Received data field	Transmitted data field
00 01	00 00	00 00

3.9.3 Return transmission of the diagnosis register (0x02)

The slave sends its 16-bit diagnosis register to the master. The data contained in this register are freely definable. For example, the information could be: EEPROM faulty, LED defective, etc.
Definition of the received and returned data:

Sub-function	Received data field	Transmitted data field
00 02	00 00	Contents of the diagnosis register

3.9.4 Change to the Listen Only mode (0x04)

The slave is instructed not to execute or answer any messages addressed to it. The device can only return to normal operation by means of the diagnosis message 'Sub-function 00 01' or by means of a new power up.

The function serves to disable a module that is behaving erratically on the MODBUS, so that the bus can continue operations. The device does not generate a response after receiving this message.
Definition of the received and returned data:

Sub-function	Received data field	Transmitted data field
00 04	00 00	No response

3.9.5 Delete the counter and reset the diagnosis register (0x0A)

The slave is instructed to delete the contents of its event counter and to reset the diagnosis register.
Definition of the received and returned data:

Sub-function	Received data field	Transmitted data field
00 0A	00 00	00 00

3.9.6 Return transmission of the message counter (0x0B)

The slave is instructed to return the value of its message counter.
The counter contains the sum of all messages, which the slave has recorded on the bus. This count includes all the messages transmitted by the master and the other slaves. The count does not include the response messages of this slave.
Definition of the received and returned data:

Sub-function	Received data field	Transmitted data field
00 0B	00 00	Message counter

3.9.7 Return transmission of the counter for faulty message transmissions (0x0C)

The slave is instructed to return the value of its counter for faulty message transmissions.
The counter contains the sum of all messages addressed to the slave, in which an error was detected. Hereby, the faults can be CRC or parity errors.
Definition of the received and returned data:

Sub-function	Received data field	Transmitted data field
00 0C	00 00	Contents of counter for faulty message transmissions

3.9.8 Return transmission of the counter for messages answered with error code (0x0D)

The slave is instructed to return the value of its counter for the messages answered with error code. The counter contains the sum of all messages addressed to the slave, and which were answered with an error code.
Definition of the received and returned data:

Sub-function	Received data field	Transmitted data field
00 0D	00 00	Contents of counter for messages answered with an error code

3.9.9 Return transmission of the message counter for this slave (0x0E)

The slave is instructed to return the value of its counter for messages to this slave.
The counter contains the sum of all messages addressed to the slave.
Definition of the received and returned data:

Sub-function	Received data field	Transmitted data field
00 0E	00 00	Contents of counter for messages addressed to this slave

3.9.10 Return transmission of the counter for unanswered messages (0x0F)

The slave is instructed to return the value of its counter for unanswered messages.
The counter contains the sum of all messages addressed to the slave, which were not answered because of internal events or detected errors.
Definition of the received and returned data:

Sub-function	Received data field	Transmitted data field
00 0F	00 00	Contents of counter for unanswered messages

3.9.11 Return transmission of the counter for messages answered with NAK (0x10)

The slave is instructed to return the value of its counter for messages answered with NAK.
The counter contains the sum of all messages addressed to the slave, which were answered with NAK.
Definition of the received and returned data:

Sub-function	Received data field	Transmitted data field
00 10	00 00	Contents of counter for messages answered with NAK

3.9.12 Return transmission of the counter for messages answered with Busy (0x11)

The slave is instructed to return the value of its counter for messages answered with Busy.
The counter contains the sum of all messages addressed to the slave, which were answered with Busy.
Definition of the received and returned data:

Sub-function	Received data field	Transmitted data field
00 12	00 00	Contents of counter for messages answered with Busy

3.9.13 Return transmission of the parity error counter (0x40)

The slave is instructed to return the value of its counter for parity errors.
The counter contains the sum of all messages addressed to the slave, in which a parity error was detected.
Definition of the received and returned data:

Sub-function	Received data field	Transmitted data field
00 40	00 00	Contents of counter for the number of parity errors

3.9.14 Return transmission of the framing error counter (0x41)

The slave is instructed to return the value of its counter for the number of framing errors.
The counter contains the sum of all messages addressed to the slave, in which a framing error was detected. A framing error occurs, if the stop bit at the end of a byte is not detected.
Definition of the received and returned data:

Sub-function	Received data field	Transmitted data field
00 41	00 00	Contents of counter for the number of framing errors

3.9.15 Return transmission of the counter for too long messages (0x12)

The slave is instructed to return the value of its counter for too long messages.
The counter contains the sum of all messages addressed to the slave, which caused an overflow of the receiving buffer, or if the data were not retrieved from the UART quickly enough.
Definition of the received and returned data:

Sub-function	Received data field	Transmitted data field
00 42	00 00	Counter for too long messages

4 MODBUS addresses, address areas, and address formats

4.1 Area definitions

The address is coded in 2 bytes. The most significant 3 bits determine the data transmission format.
The following formats are available for *rail line* devices:

- **Integer**
- **Integer with 1 decimal**
- **(Float acc. to IEEE)**

Address area hex	dez.	Data transfer format	Smallest transferable value	Largest transferable value	Resolution
0x0000 ... 0x1FFF	0 ... 8191	Integer without decimals	-30000	+32000	+/- 1
0x2000 ... 0x3FFF	8192 ... 16383	Integer with 1 decimal	-3000.0	+3200.0	+/- 0.1
0x4000 ... 0x7FFF	16384...32767	Float (IEEE format)	-1.0 E+037	+1.0 E+037	+/-1.4E-045

For integer numbers with and without decimals, the value range -30000 to +32000 is transmitted via the interface. Scaling with the factor 1 or 10 must be carried out by the transmitting device as well as by the receiving device.

- Values are transmitted in the Motorola format (big endian).
- The relevant areas are grouped for process data, parameter and configuration data reading and writing.
- Multiple definition of process data in different groups is possible.

4.2 Special values

The following special values are defined for transmission in the integer format:

- -31000 Sensor fault
  This value is returned for data that do not represent a meaningful value due to a sensor fault.

- -32000 Switch-off value
  The function is disabled.

- -32500 Undefined value
  The device returns this value, if a datum is not defined within the requested range („NOT DEFINED VALUE“).

- -32768 Corresponds to 0x8000 hex.
  The value to be transmitted lies outside the transferable integer value range.

The following special values are defined for transmission in the Float format:

- -1.5E37 This datum is not defined.
  The device returns this value, if a datum is not defined within the requested range.

4.3 Composition of the address tables

In the address tables shown in Section 5, the addresses for every parameter of the corresponding data format are specified in decimal values.
The tables are structured as follows:

Name	R/W	Address	Integer	Real	Type	Value/off	Description
		base 1dP					

- Name — Description of the datum
- R/W — permitted type of access: R = read, W = write
- Address integer — Address for integer values
- base — Integer without decimals
- 1 dP — Integer with 1 decimal
- Real — Floating point number / Float (IEEE format)
- Type — internal data type
- Value/off — permissible value range, switch-off value available
- Description — Explanations

4.4 Internal data types

The following data types are assigned to data used in the device:

- Float
  Floating point number
  Value range: -1999 ... -0.001, 0, 0.001 ... 9999
- INT
  Positive whole integer number
  Value range: 0 ... 65535
  Exception: Switch-off value '-32000'
- Text
  Text string consisting of n characters, currently defined n = 5
  Permissible characters: 20H...7FH
- Long
  Positive whole Long number
  Value range: 0 … 99999
- Enum
  Selection value

5 Index

Index

6 Address tables

The following sections describe the address tables for:

- **Industrial controller KS 40-1, valid also for KS 41-1 and KS 42-1**

Table Of Contents

1 Cntr

ConF

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
SP.Fn	r/w	base 1dP 2dP 3dP	3150 11342 19534 27726	39068	Enum	*Enum_SPFN*	Basic configuration for setpoint processing, e.g. 'setpoint controller switchable to external setpoint'. Configuration of special, controller-dependent setpoint functions.

0	set-point controller can be switched over to external set-point (->LOGI/SP.E)
1	Program controller for setpoint profile. The program profile is definable by the user.
2	Timer, operating mode 1 (bandwidth monitoring, switch-off at the end). After timer start, the controller lines out at the defined setpoint. The timer time (t.SP) runs when the process value enters the adjusted band around the setpoint (x = SP ± b.ti). When the timer has elapsed, the controller is switched to Y2 (= fixed positioning value) and the lower display alternates between 'End' and the setpoint.
3	Timer, operating mode 2 (bandwidth monitoring, pause at the end). After timer start, the controller lines out at the defined setpoint. The timer time (t.SP) runs when the process value enters the adjusted band around the setpoint (x = SP ± b.ti). When the timer has elapsed, the controller continues with setpoint SP, and the lower display alternates between 'End' and the setpoint.
4	Timer, operating mode 3 (switch-off at the end). After timer start, the controller lines out at the defined setpoint. The timer time (t.SP) runs immediately after switchover. When the timer has elapsed, the controller is switched to Y2 (= fixed positioning value) and the lower display alternates between 'End' and the setpoint.
5	Timer, operating mode 4 (pause at the end). After timer start, the controller lines out at the defined setpoint. The timer time (t.SP) runs immediately after switchover. When the timer has elapsed, the controller continues with setpoint SP, and the lower display alternates between 'End' and the setpoint.
6	Timer, operating mode 5 (delayed start). The timer starts immediately. The controller continues with Y2 (= fixes positioning value). When the timer (t.SP) has elapsed, the controller switches over to the adjusted setpoint.
7	Timer, operating mode 6 (setpoint switchover). After switching over from SP to SP.2, the controller lines out at SP.2. The time (t.SP) runs when the process value enters the adjusted band around the setpoint (x = SP ± b.ti). When the timer has elapsed, the controller switches back to setpoint SP, and the lower display alternates between 'End' and the setpoint.

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
b.ti	r/w	base 1dP 2dP 3dP	3152 11344 19536 27728	39072	Float	0...9999 ☐	Timer tolerance band for operating mode:1 (bandwidth monitoring with switch-off at the end)2 (bandwidth monitoring with pause at the end), and6 (setpoint switchover). The timer runs as long as the process value is within the bandwidth limits (setpoint ± b.ti).
C.Fnc	r/w	base 1dP 2dP 3dP	5050 13242 21434 29626	42868	Enum	*Enum_CFnc*	Control behaviour (algorithm) referred to output value: e.g. 2- or 3-point controller, signaller, 3-point stepping control.

0	on/off controller or signaller with one output. The on/off controller or signaller switches if the process value drifts from the setpoint more than the hysteresis.
1	PID control, e.g. heating, with one output: Switched as a digital output (2-point) or used as an analog output (continuous). PID controllers respond quickly to changes of the control deviation, and typically do not exhibit any permanent control offset.
2	D / Y / Off, or 2-point controller with partial/full load switch-over. 2 digital outputs: Y1 is the switching output and Y2 is the changeover contact for D/Y.
3	2 x PID control, e.g. heating/cooling. Two outputs: Switched as a digital output (3-point) or used as an analog output (continuous). PID controllers respond quickly to changes of the control deviation, and typically do not exhibit any permanent control offset.
4	3-point stepping controller, e.g. for motor actuators. Two digital outputs. No actuating pulses are generated when the process is lined out.

1 Cntr

ConF

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
mAn	r/w	base 1dP 2dP 3dP	5051 13243 21435 29627	42870	Enum	*Enum_mAn*	Enables the output value to be adjusted in manual operation. If adjustment is not enabled, the output value cannot be changed in manual operation, neither with the front keys nor via the interface.Note: This setting does not affect the auto/manual switchover function.

0	The output value cannot be changed in manual operation, neither with the front keys nor via the interface.
1	The output value is to be adjusted in manual operation (see also LOGI/mAn).

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
C.Act	r/w	base 1dP 2dP 3dP	5052 13244 21436 29628	42872	Enum	*Enum_CAct*	Operating sense of the controller. Inverse operation (e.g. heating) means increased heat input when the process value falls. Direct operation (e.g. cooling) means increased heat input when the process value increases.

0	Inverse or opposed-sense response, e.g. heating. The controller output is increased with a falling process value, and decreased with a rising process value.
1	Direct or same-sense response, e.g. cooling. The controller output is increased with a rising process value, and decreased with a falling process value.

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
FAIL	r/w	base 1dP 2dP 3dP	5053 13245 21437 29629	42874	Enum	*Enum_FAIL*	With the sensor break response, the operator determines the instrument's reaction to a sensor break, thus ensuring a safe process condition.

0	controller outputs switched off
1	y = parameter Y2 (Caution: fixed parameter Y2, not controller output Y2!). Note for three-point stepping controller: With Y2 < 0.01 CLOSED is set (DY= -100%), with 0.01 =< Y2 =< 99.9 no output is set (DY=0%), with Y2 > 99.9 OPEN is set (DY= +100%). Note for signallers: With Y2 < 0.01 OFF is set, with 0.01 =< Y2 =< 99.9 status keeps unchanged, with Y2 > 99.9 ON is set.
2	y = mean output. The maximum permissible output can be adjusted with parameter Ym.H. To prevent determination of inadmissible values, mean value formation is only if the control deviation is lower than parameter L.Ym.

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
rnG.L	r/w	base 1dP 2dP 3dP	5059 13251 21443 29635	42886	Float	-1999...9999 ☐	Lower limit for the controller's operating range. The control range is independent of the measurement range. Reducing the control range will increase the sensitivity of the self-tuning process.
rnG.H	r/w	base 1dP 2dP 3dP	5060 13252 21444 29636	42888	Float	-1999...9999 ☐	Upper limit for the controller's operating range. The control range is independent of the measurement range. Reducing the control range will increase the sensitivity of the self-tuning process.
Adt0	r/w	base 1dP 2dP 3dP	5061 13253 21445 29637	42890	Enum	*Enum_Adt0*	Optimization of the switching cycles t1 and t2 for the DED conversion can be disabled here. In order to fine-tune the positioning action, the switching periods are changed by the self-tuning function, if automatic tuning is configured.

0	The cycle duration is determinated by auto-tuning. Thereby the best controlling results are obtained.
1	The cycle duration is not determinated by auto-tuning. An oversized cycle duration causes bad control behavior. An undersized cycle duration causes a more frequent switching, which can raise the wearout of mechanical actuators (relay, contactor).

1 Cntr

PArA

Name	r/w	Adr. Integer		real	Typ	Value/off		Description
Pb1	r/w	base 1dP 2dP 3dP	5000 13192 21384 29576	42768	Float	1...9999	☐	Proportional band 1 (heating) in engineering unit, e.g. °C. Pb defines the relationship between controller output and control deviation. The smaller Pb is, the stronger is the control action for a given control deviation. If Pb is too large or too small, the control loop will oscillate (hunting).
Pb2	r/w	base 1dP 2dP 3dP	5001 13193 21385 29577	42770	Float	1...9999	☐	Proportional band 2 (cooling) in engineering units, e.g. °C. Pb defines the relationship between controller output and control deviation. The smaller Pb is, the stronger is the control action for a given control deviation. If Pb is too large or too small, the control loop will oscillate (hunting).
ti1	r/w	base 1dP 2dP 3dP	5002 13194 21386 29578	42772	Float	1...9999	☑	Integral action time 1 (heating) [s]. Ti is the time constant of the integral portion. The smaller Ti is, the faster is the response of the integral action. Ti too small: Control tends to oscillate. Ti too large: Control is sluggish and needs a long time to line out.
ti2	r/w	base 1dP 2dP 3dP	5003 13195 21387 29579	42774	Float	1...9999	☑	Integral action time 2 (cooling) [s]. Ti is the time constant of the integral portion. The smaller Ti is, the faster is the response of the integral action. Ti too small: Control tends to oscillate. Ti too large: Control is sluggish and needs a long time to line out.
td1	r/w	base 1dP 2dP 3dP	5004 13196 21388 29580	42776	Float	1...9999	☑	Derivative action time 1 (heating) [s], second parameter set. Td is the time constant of the derivative portion. The faster the process value changes, and the larger the value of Td is, the stronger will be the derivative action. Td too small: Very little derivative action. Td too large: Control tends to oscillate.
td2	r/w	base 1dP 2dP 3dP	5005 13197 21389 29581	42778	Float	1...9999	☑	Derivative action time 2 (cooling) [s], second parameter set. Td is the time constant of the derivative portion. The faster the process value changes, and the larger the value of Td is, the stronger will be the derivative action. Td too small: Very little derivative action. Td too large: Control tends to oscillate.
t1	r/w	base 1dP 2dP 3dP	5006 13198 21390 29582	42780	Float	0,4...9999	☐	Minimum duty cycle 1 (heating) [s]. With the standard duty cycle converter, the shortest pulse duration is 1/4 x t1. If the duty cycle is not to be optimized, this must be entered in the configuration. (Default: Optimization of the duty cycle during self-tuning, but also if the output value is less than 5%).
t2	r/w	base 1dP 2dP 3dP	5007 13199 21391 29583	42782	Float	0,4...9999	☐	Minimum duty cycle 2 (cooling) [s]. With the standard duty cycle converter, the shortest pulse duration is 1/4 x t1. If the duty cycle is not to be optimized, this must be entered in the configuration. (Default: Optimization of the duty cycle during self-tuning, but also if the output value is less than 5%).
SH	r/w	base 1dP 2dP 3dP	5014 13206 21398 29590	42796	Float	0...9999	☐	Neutral zone, or switching difference of the signaller [engineering unit].Too small: unnecessarily high switching frequency.Too large: reduced controller sensitivity.With 3-point controllers this slows down the direct transition from heating to cooling. With 3-point stepping controllers, it reduces the switching operations of the actuator around setpoint.

1 Cntr

PArA

Name	r/w	Adr. Integer		real	Typ	Value/off		Description
d.SP	r/w	base 1dP 2dP 3dP	5016 13208 21400 29592	42800	Float	-1999...9999	☐	Separation of the D / Y switch-over point from the setpoint [engineering unit]. With a significant control deviation heating start is in delta connection. When the control deviation increases, the instrument switches over to reduced power (Y connection) for line-out to the set-point.
tP	r/w	base 1dP 2dP 3dP	5009 13201 21393 29585	42786	Float	0,1...9999	☑	Minimum pulse duration [s]. Used for switching with constant periods. For positioning values that require a shorter pulse than adjusted for 'tp', the output is suppressed, but 'remembered'. The controller continues adding the internal short pulses until a value equal to 'tp' can be output.
tt	r/w	base 1dP 2dP 3dP	5015 13207 21399 29591	42798	Float	3...9999	☐	Travel time of the actuator motor [s]. If no feedback signal is available, the controller calculates the actuator position by means of an integrator and the adjusted motor travel time. For this reason, a precise definition of the motor travel time between min and max (0% and 100%) is important.
Y.Lo	r/w	base 1dP 2dP 3dP	5018 13210 21402 29594	42804	Float	-105...105	☐	Lower output limit [%] The range is depedant of the type of controller: 2 point controller: 0...ymax+1 3 point controller: -105 ymax-1
Y.Hi	r/w	base 1dP 2dP 3dP	5019 13211 21403 29595	42806	Float	-105...105	☐	Upper output limit [%] The range is ymin+1 105
Y2	r/w	base 1dP 2dP 3dP	5017 13209 21401 29593	42802	Float	-100...100	☐	Second positioning value [%]. Activated Y2 = positioner control. Caution: The parameter 'positioning output Y2' must not be confused with the controller output Y2!
Y.0	r/w	base 1dP 2dP 3dP	5020 13212 21404 29596	42808	Float	-105...105	☐	Offset for die positioning value [%]. This is added to the controller output, and has the most effect with P and PD controllers. (With PID controllers, the effect is compensated by the integral action.) With a control deviation = 0, the P controller generates a control output Y0.
Ym.H	r/w	base 1dP 2dP 3dP	5021 13213 21405 29597	42810	Float	-105...105	☐	Limit for the mean control output value Ym in case of sensor break [%]. The mean control output value is configurable as the response to sensor break. The maximum mean output value = YmH.
L.Ym	r/w	base 1dP 2dP 3dP	5022 13214 21406 29598	42812	Float	1...9999	☐	Max. control deviation (xw), at the start of mean value calculation [engineering unit]. When calculating the mean value, data are only taken into account if the control deviation is small enough. 'Lym' is a preset value that determines how precisely the calculated output value is matched to the setpoint.

1 Cntr

Signal

Name	r/w	Adr. Integer		real	Typ	Value/off		Description
Tu2	r	base 1dP 2dP 3dP	5145 13337 21529 29721	43058	Float	0...9999	☐	'Cooling' delay time of the loop. Tu is calculated by the self-tuning function: It is the time delay before the process reacts significantly. In effect, Tu is a dead time that is determined by the reaction of the process to a change of the control output. It is used for defining controller action.
Vmax2	r	base 1dP 2dP 3dP	5146 13338 21530 29722	43060	Float	0...9999	☐	Max. rate of change for 'cooling', i.e. the fastest process value increase during self-tuning. Vmax is calculated by the self-tuning function, and is determined by the reaction of the process to a change of the control output. It is used for defining controller action.
Kp2	r	base 1dP 2dP 3dP	5147 13339 21531 29723	43062	Float	0...9999	☐	Process gain for 'cooling'. For control loops with self-regulation, process gain is the ratio determined by the change of the control output and the resulting permanent change of the process value. Kp is calculated by the self-tuning function, and is used for defining controller action.
St.Cntr	r	base 1dP 2dP 3dP	5100 13292 21484 29676	42968	Int	0...65535	☐	Status informations of the controller.f.e. switching signals, controller off or informations about selftuning. The controller sratus shows the actual adjustments of the controller.

Bit 0: Switching signal heating: 0: off 1: on
Bit 1: Switching signal cooling: 0: off 1: on
Bit 2: Sensor error 0: ok 1: error
Bit 3: Controlsignal: Manual/automatic
0: automatic 1: manual
Bit 4: Controlsignal: Y2
0: Y2 not activ 1: Y2 activ
Bit 5: Controlsignal: Ext. setting of outputsignal
0: not activ 1: activ
Bit 6: Controlsignal: Controller off
0: contr. on 1: contr. off
Bit 7: Controlsignal:The activ parameter set
0: parameterset 1
1: parameterset 2
Bit 8: Loopalarm
0: no alarm
1: alarm
Bit 9: Soft start function
0: not activ
1: activ
Bit 10: Rate to setpoint
0: not activ
1: activ
Bit 11: Not used
Bit 12-15: Internal functional statuses (operating state)
0 0 0 0 Automatic
0 0 0 1 Selftuning is running
0 0 1 0 Selftuning faulty
(Waiting for operator signal)
0 0 1 1 Sensor error
0 1 0 0 Not used
0 1 0 1 Manual
0 1 1 1 Not used
1 0 0 0 Manual, with external presetting of the outputsignal
1 0 0 1 Outputs switched off (neutral)
1 0 1 0 Abortion of the selftuning (by control- or error-signal)

1 Cntr

Signal

Name	r/w	Adr. Integer		real	Typ	Value/off		Description
diFF	r	base 1dP 2dP 3dP	5104 13296 21488 29680	42976	Float	-1999...9999	☐	Control deviation, is defined as process value minus setpoint. Positive Xw means that the process value is above the setpoint. A small control deviation indicates precise control.
POS	r	base 1dP 2dP 3dP	5105 13297 21489 29681	42978	Float	0...100	☐	The position feedback Yp shows the actuator position with 3-point stepping controllers. If Yp is outside the limits Ymin and Ymax, the output of positioning pulses is suppressed.
Tu1	r	base 1dP 2dP 3dP	5141 13333 21525 29717	43050	Float	0...9999	☐	'Heating' delay time of the loop. Tu is calculated by the self-tuning function: It is the time delay before the process reacts significantly. In effect, Tu is a dead time that is determined by the reaction of the process to a change of the control output. It is used for defining controller action.
Ypid	r	base 1dP 2dP 3dP	5103 13295 21487 29679	42974	Float	-120...120	☐	Output value Ypid is the output signal determined by the controller, and from which the switching pulses for the digital and analog control outputs are calculated. Ypid is also available as an analog signal. e.g. for visualization.
Ada.St	r/w	base 1dP 2dP 3dP	5150 13342 21534 29726	43068	Enum	*Enum_AdaStart*		Starting / stopping the self-tuning function.After the start signal, the controller waits until the process reaches a stable condition (PIR) before it starts the self-tuning process. Self-tuning can be aborted manually at any time. After a successful self-tuning attempt, the controller automatically resumes normal operation.

0	'Stop' will abort the self-tuning process, and the controller returns to normal operation with the previous parameter settings.
1	Start of the self-tuning process is possible during manual or automatic controller operation.

Name	r/w	Adr. Integer		real	Typ	Value/off		Description
Yman	r/w	base 1dP 2dP 3dP	5151 13343 21535 29727	43070	Float	-110...110	☐	Absolute preset output value, which is used as output value during manual operation. Caution: With 3-point stepping controllers, Yman (evaluated the same as Dyman) is added to the actual output value as a relative shift.
dYman	r/w	base 1dP 2dP 3dP	5152 13344 21536 29728	43072	Float	-220...220	☐	Differential preset output value, which is added to the actual output value during manual operation. Negative values reduce the output.
Yinc	r/w	base 1dP 2dP 3dP	5153 13345 21537 29729	43074	Enum	*Enum_YInc*		Increasing the output value. There are two speeds: 40 s or 10 s for the change from 0 % to 100 %. Note: The 3-point stepping controller translates the increments as UP.

0	Not active
1	increment output

1 Cntr

Signal

Name	r/w	Adr. Integer		real	Typ	Value/off		Description
Ydec	r/w	base 1dP 2dP 3dP	5154 13346 21538 29730	43076	Enum	*Enum_YDec*		Decreasing the output value. There are two speeds: 40 s or 10 s for the change from 0 % to 100 %. Note: The 3-point stepping controller translates the increments as DOWN.

0	Not active
1	decrement output

Name	r/w	Adr. Integer		real	Typ	Value/off		Description
SP.EF	r	base 1dP 2dP 3dP	5101 13293 21485 29677	42970	Float	-1999...9999	☐	Effective setpoint. The value reached at the end of setpoint processing, after taking W2, external setpoint, gradient, boost function, programmer settings, start-up function, and limit functions into account. Comparison with the effective process value leads to the control deviation, from which the necessary controller response is derived.
In.1	r	base 1dP 2dP 3dP	5102 13294 21486 29678	42972	Float	-1999...9999	☐	Measurement value after the measurement value correction (e.g. with offset or 2-point correction, and scaling).
St.Tune	r	base 1dP 2dP 3dP	5140 13332 21524 29716	43048	Int	0...65535	☐	Status information during self-tuning, e.g. the actual condition, and possible results, warnings, and error messages.

Bit 0 Process lined out; 0 = No; 1 = Yes
Bit 1 Operating mode 'Self-tuning controller; 0 = Off; 1 = On
Bit 2 Result of controller self-tuning; 0 = OK; 1 = Fault
Bit 3 - 7 Not used
Bit 8 - 11 Result of the 'heating' attempt
0 0 0 0 No message / Attempt still running
0 0 0 1 Successful
0 0 1 0 Successful, with risk of exceeded setpoint
0 0 1 1 Error: Wrong operating sense
0 1 0 0 Error: No response from process
0 1 0 1 Error: Turning point too low
0 1 1 0 Error: Risk of exceeded setpoint
0 1 1 1 Error: Step output too small
1 0 0 0 Error: Setpoint reserve too small
Bit 12 - 15 Result of 'cooling' attempt (same as heating attempt)

Name	r/w	Adr. Integer		real	Typ	Value/off		Description
Vmax1	r	base 1dP 2dP 3dP	5142 13334 21526 29718	43052	Float	0...9999	☐	Max. rate of change for 'heating', i.e. the fastest process value increase during self-tuning. Vmax is calculated by the self-tuning function, and is determined by the reaction of the process to a change of the control output. It is used for defining controller action.
Kp1	r	base 1dP 2dP 3dP	5143 13335 21527 29719	43054	Float	0...9999	☐	Process gain for 'heating'. For control loops with self-regulation, process gain is the ratio determined by the change of the control output and the resulting permanent change of the process value. Kp is calculated by the self-tuning function, and is used for defining controller action.

1 Cntr

Signal

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
Msg2	r	base 1dP 2dP 3dP	5148 13340 21532 29724	43064	Enum	*Enum_Msg*	The result of self-tuning for 'cooling' indicates whether self-tuning was successful, and with what result.

0	No message / Tuning attempt still running
1	Self-tuning has been completed successfully. The new parameters are valid.
2	Self-tuning was successful, but with a warning. The new parameters are valid. Note: Self-tuning was aborted due to the risk of an exceeded setpoint, but useful parameters were determined. Possibly repeat the attempt with an increased setpoint reserve.
3	The process reacts in the wrong direction. Possible remedy: Reconfigure the controller (inverse <-> direct). Check the controller output sense (inverse <-> direct).
4	No response from the process. Perhaps the control loop is open. Possible remedy: Check sensor, connections, and process.
5	The process value turning point of the step response is too low. Possible remedy: Increase the permitted step output range, i.e. increase the parameter Y.Hi ('heating') or reduce the parameter Y.Lo ('cooling').
6	Self-tuning was aborted due to the risk of an exceeded setpoint. No useful parameters were determined. Possible remedy: Repeat the attempt with an increased setpoint reserve.
7	The step output change is not large enough (minimum change > 5 %). Possible remedy: Increase the permitted step output range, i.e. increase the parameter Y.Hi ('heating') or reduce the parameter Y.Lo ('cooling').
8	The controller is waiting. Setpoint reserve must be given before generating the step output change. Acknowledgment of this error message leads to switch-over to automatic mode. If self-tuning shall be continued, change set-point, change process value, or decrease set-point range.

1 Cntr

Signal

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
Msg1	r	base 1dP 2dP 3dP	5144 13336 21528 29720	43056	Enum	*Enum_Msg*	The result of self-tuning for 'heating' indicates whether self-tuning was successful, and with what result.

0	No message / Tuning attempt still running
1	Self-tuning has been completed successfully. The new parameters are valid.
2	Self-tuning was successful, but with a warning. The new parameters are valid. Note: Self-tuning was aborted due to the risk of an exceeded setpoint, but useful parameters were determined. Possibly repeat the attempt with an increased setpoint reserve.
3	The process reacts in the wrong direction. Possible remedy: Reconfigure the controller (inverse <-> direct). Check the controller output sense (inverse <-> direct).
4	No response from the process. Perhaps the control loop is open. Possible remedy: Check sensor, connections, and process.
5	The process value turning point of the step response is too low. Possible remedy: Increase the permitted step output range, i.e. increase the parameter Y.Hi ('heating') or reduce the parameter Y.Lo ('cooling').
6	Self-tuning was aborted due to the risk of an exceeded setpoint. No useful parameters were determined. Possible remedy: Repeat the attempt with an increased setpoint reserve.
7	The step output change is not large enough (minimum change > 5 %). Possible remedy: Increase the permitted step output range, i.e. increase the parameter Y.Hi ('heating') or reduce the parameter Y.Lo ('cooling').
8	The controller is waiting. Setpoint reserve must be given before generating the step output change. Acknowledgment of this error message leads to switch-over to automatic mode. If self-tuning shall be continued, change set-point, change process value, or decrease set-point range.

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
YGrw	r/w	base 1dP 2dP 3dP	5155 13347 21539 29731	43078	Enum	*Enum_YGrwLs*	Gradient of Y-variation 'slow' or 'fast'. Changes the positioning output speed. There are two speeds for output variation: from 0% to 100% in 40s or in 10s.

0	Slow change of Y, from 0% to 100% in 40 seconds.
1	Fast change of Y, from 0% to 100% in 10 seconds.

2 InP.1

ConF

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
S.tYP	r/w	base 1dP 2dP 3dP	1150 9342 17534 25726	35068	Enum	*Enum_StYP*	Sensor type selection. For sensors with signals of resistance transducer, current or voltage measuring, scaling can be adjusted

0	thermocouple type L (-100...900°C), Fe-CuNi DIN Fahrenheit: -148...1652°F
1	thermocouple type J (-100...1200°C), Fe-CuNi Fahrenheit: -148...2192°F
2	thermocouple type K (-100...1350°C), NiCr-Ni Fahrenheit: -148...2462°F
3	thermocouple type N (-100...1300°C), Nicrosil-Nisil Fahrenheit: -148...2372°F
4	thermocouple type S (0...1760°C), PtRh-Pt10% Fahrenheit: 32...3200°F
5	thermocouple type R (0...1760°C), PtRh-Pt13% Fahrenheit: 32...3200°F
18	Special thermocouple with a linearization characteristic selectable by the user. This enables non-linear signals to be simulated or linearized.
20	Pt100 (-200.0 ... 100.0(150.0)°C) Measuring range up to 150°C at reduced lead resistance. Fahrenheit: -328...212(302) °F
21	Pt100 (-200.0 ... 850,0 °C) Fahrenheit: -328...1562 °F
22	Pt 1000 (-200.0...850.0 °C) Fahrenheit: -328...1562 °F
23	Special : 0...4500 Ohms. For KTY 11-6 with preset special linearization (-50...150 °C or -58...302 °F).
30	Current : 0...20 mA / 4...20 mA
40	0...10V / 2...10V

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
S.Lin	r/w	base 1dP 2dP 3dP	1151 9343 17535 25727	35070	Enum	*Enum_SLin*	Linearization (not adjustable for all sensor types S.tYP). Special linearization. The linearization table can be created with the Engineering Tool. The default characteristic is for KTY 11-6 temperature sensors.

0	No special linearization.
1	Special linearization. Definition of the linearization table is possible with the Engineering Tool. The default setting is the characteristic of the KTY 11-6 temperature sensor.

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
Corr	r/w	base 1dP 2dP 3dP	160 8352 16544 24736	33088	Enum	*Enum_Corr3*	Measured value correction / scaling

0	Without scaling
1	The offset correction (in the CAL Level) can be done on-line in the process. If InL shows the lower input value of the scaling point, then OuL must be adjusted to the corresponding display value. Adjustments are made via the front panel keys of the device only.
2	Two-point correction (in CAL-Level) ist possible offline via process value transmitter or on-line in the process. Set process value for the upper and lower scaling point and confirm as input value InL or InH, then set the belonging displayed value OuL and OuH. The settings are done via the front of the device.
3	Scaling (at PArA-level). The input values for the upper (InL, OuL) and lower scaling point (InH. OuH) are visible at the parameter level. Adjustment is made via front operation or the engineering tool.

2 InP.1

PArA

Name	r/w	Adr. Integer		real	Typ	Value/off		Description
InL.1	r/w	base 1dP 2dP 3dP	1100 9292 17484 25676	34968	Float	-1999...9999	☐	Input value of the lower scaling point. Depending on sensor type, the input value can be scaled to the required display value in the Parameter Level. The display of the input value of the lower scaling point (e.g. 4 mA) is done using the corresponding electrical value.
OuL.1	r/w	base 1dP 2dP 3dP	1101 9293 17485 25677	34970	Float	-1999...9999	☐	Display value of the lower scaling point. Depending on sensor type, the input value can be scaled to the required display value in the Parameter Level. The operator can change the display value of the lower scaling point, e.g. 4 mA is displayed as 2 [pH].
InH.1	r/w	base 1dP 2dP 3dP	1102 9294 17486 25678	34972	Float	-1999...9999	☐	Input value of the upper scaling point. Depending on sensor type, the input value can be scaled to the required display value in the Parameter Level. The display of the input value of the upper scaling point (e.g. 20 mA) is done using the corresponding electrical value.
OuH.1	r/w	base 1dP 2dP 3dP	1103 9295 17487 25679	34974	Float	-1999...9999	☐	Display value of the upper scaling point. Depending on sensor type, the input value can be scaled to the required display value in the Parameter Level. The operator can change the display value of the upper scaling point, e.g. 20 mA is displayed as 12 [pH].
t.F1	r/w	base 1dP 2dP 3dP	1104 9296 17488 25680	34976	Float	0...100	☐	Filter time constant [s]. Every input is fitted with a digital (software) low-pass filter for suppressing process-related disturbances on the input leads. Higher filter settings improve the suppression, but increase the delay of the input signals.

Signal

Name	r/w	Adr. Integer		real	Typ	Value/off		Description
In.1r	r	base 1dP 2dP 3dP	1170 9362 17554 25746	35108	Float	-1999...9999	☐	Measurement value before the measurement value correction (unprocessed).
Fail	r	base 1dP 2dP 3dP	1171 9363 17555 25747	35110	Enum	*Enum_InpFail*		Input circuit fault: faulty or incorrectly connected sensor.

0	no error
1	sensor break
2	Incorrect polarity at input.
4	Short circuit at input.

Name	r/w	Adr. Integer		real	Typ	Value/off		Description
In.1	r	base 1dP 2dP 3dP	1172 9364 17556 25748	35112	Float	-1999...9999	☐	Measurement value after the measurement value correction (e.g. with offset or 2-point correction, and scaling).
F.Inp	r/w	base 1dP 2dP 3dP	1180 9372 17564 25756	35128	Float	-1999...9999	☐	Forcing the value for an analog input INP. Forcing involves the external operation of an input. The instrument takes over the value at this input like a measurement value (preset value for inputs from a superordinate system, e.g. for a function test.)

3 InP.2

ConF

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
I.Fnc	r/w	base 1dP 2dP 3dP	161 8353 16545 24737	33090	Enum	*Enum_IFnc*	Selection of the function assigned to the value at INP2, e.g. value at INP2 is the external setpoint.

0	no function (subsequent input data are skipped)
1	Heating current input.
2	External setpoint SP.E or (depending on version) external setpoint shift SP.E. (Switchover is done via -> LOGI/SP.E).

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
S.tYP	r/w	base 1dP 2dP 3dP	1250 9442 17634 25826	35268	Enum	*Enum_StYP2*	Sensor type selection. For sensors with signals of resistance transducer, current or voltage measuring, scaling can be adjusted.

30	Current : 0...20 mA / 4...20 mA
31	0...50 mA current (AC)

PArA

Name	r/w	Adr. Integer		real	Typ	Value/off		Description
InL.2	r/w	base 1dP 2dP 3dP	1200 9392 17584 25776	35168	Float	-1999...9999	☐	Input value of the lower scaling point. Depending on sensor type, the input value can be scaled to the required display value in the Parameter Level. The display of the input value of the lower scaling point (e.g. 4 mA) is done using the corresponding electrical value.
OuL.2	r/w	base 1dP 2dP 3dP	1201 9393 17585 25777	35170	Float	-1999...9999	☐	Display value of the lower scaling point. Depending on sensor type, the input value can be scaled to the required display value in the Parameter Level. The operator can change the display value of the lower scaling point, e.g. 4 mA is displayed as 2 [pH].
InH.2	r/w	base 1dP 2dP 3dP	1202 9394 17586 25778	35172	Float	-1999...9999	☐	Input value of the upper scaling point. Depending on sensor type, the input value can be scaled to the required display value in the Parameter Level. The display of the input value of the upper scaling point (e.g. 20 mA) is done using the corresponding electrical value.
OuH.2	r/w	base 1dP 2dP 3dP	1203 9395 17587 25779	35174	Float	-1999...9999	☐	Display value of the upper scaling point. Depending on sensor type, the input value can be scaled to the required display value in the Parameter Level. The operator can change the display value of the upper scaling point, e.g. 20 mA is displayed as 12 [pH].

Signal

Name	r/w	Adr. Integer		real	Typ	Value/off		Description
In.2	r	base 1dP 2dP 3dP	1270 9462 17654 25846	35308	Float	-1999...9999	☐	Measurement value after the measurement value correction (e.g. with offset or 2-point correction, and scaling).

3 InP.2

Signal

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
Fail	r	base	1271	35310	Enum	*Enum_InpFail*	Input circuit fault: faulty or incorrectly connected sensor.
		1dP	9463				
		2dP	17655				
		3dP	25847				

0	no error
1	sensor break
2	Incorrect polarity at input.
4	Short circuit at input.

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
In.2r	r	base	1272	35312	Float	-1999...9999 ☐	Measurement value before the measurement value correction (unprocessed).
		1dP	9464				
		2dP	17656				
		3dP	25848				
F.Inp	r/w	base	1280	35328	Float	-1999...9999 ☐	Forcing the value for an analog input INP. Forcing involves the external operation of an input. The instrument takes over the value at this input like a measurement value (preset value for inputs from a superordinate system, e.g. for a function test.)
		1dP	9472				
		2dP	17664				
		3dP	25856				

4 Lim

ConF

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
Fnc.1	r/w	base	2150	37068	Enum	*Enum_Fcn*	Activation and adjustment of the limit value alarm (e.g. for input circuit monitoring), e.g. with/without storage.
		1dP	10342				
		2dP	18534				
		3dP	26726				

0	No limit value monitoring.
1	measured value monitoring. The alarm signal is generated, if the limit is exceeded. If the measured value is within the limits (including hysteresis) again, this alarm signal is resetted.
2	Measured value monitoring + alarm status latch. An alarm signal is generated, if the limit is exceeded. A latched alarm signal remains latched until it is manually resetted.

4 Lim

ConF

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
Src.1	r/w	base	2151	37070	Enum	*Enum_Src*	Source for limit value. Selection of which value is to be monitored.
		1dP	10343				
		2dP	18535				
		3dP	26727				

0	Process value = absolute alarm
1	control deviation xw (process value - set-point) = relative alarm Note: Monitoring with the effective set-point Weff. For example using a ramp it is the changing set-point, not the target set-point of the ramp.
2	Control deviation Xw (= relative alarm) with suppression during start-up and setpoint changes. Limit value monitoring is continued as soon as the control deviation comes within the alarm limits again, at the latest after 10 * Tn.
6	effective set-point Weff. For example the ramp-function changes the effective set-point untill it matches the internal (target) set-point.
7	correcting variable y (controller output)
11	Control deviation Xw (= relative alarm) with suppression during start-up and setpoint change. Limit value monitoring is continued as soon as the control deviation comes within the alarm limits again.

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
HC.AL	r/w	base	2050	36868	Enum	*Enum_HCAL*	Activation of alarm heat current function. Either overload or break can be monitored, overload = current I > heat current limit, or break = current I < heat current limit. Short circuit is monitored in both cases.
		1dP	10242				
		2dP	18434				
		3dP	26626				

0	No heating current alarm.
1	Overload and short circuit monitoring. Overload = current I > heat current limit.
2	Break and short circuit monitoring. Break = current I < heat current limit.

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
LP.AL	r/w	base	5058	42884	Enum	*Enum_LPAL*	Monitoring of control loop interruption (not possible with 3-point stepping controller, not possible with signaller)
		1dP	13250				
		2dP	21442				
		3dP	29634				

0	switched off / inactive
1	LOOP alarm is generated, if with Y=100% there is no corresponding reaction of the process variable within the time of 2 x ti. Possible remedial action: Check heating or cooling circuit, check sensor and replace it, if necessary, check controller and switching device.

PArA

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
L.1	r/w	base	2100	36968	Float	-1999...9999 ☑	Lower limit value. The alarm is triggered if the value falls below the limit, and is reset with lower limit value plus hysteresis.
		1dP	10292				
		2dP	18484				
		3dP	26676				
H.1	r/w	base	2101	36970	Float	-1999...9999 ☑	Upper limit value. The alarm is triggered if the value rises above the limit, and is reset with upper lower limit value plus hysteresis.
		1dP	10293				
		2dP	18485				
		3dP	26677				

4 Lim

PArA

Name	r/w	Adr. Integer		real	Typ	Value/off		Description
HYS.1	r/w	base 1dP 2dP 3dP	2102 10294 18486 26678	36972	Float	0...9999	☐	Hysteresis of the limit value. Switching difference for upper and lower limit value. The limit value must change by this amount (rise above upper limit or fall below lower limit) before the limit value alarm is reset.
HC.A	r/w	base 1dP 2dP 3dP	2000 10192 18384 26576	36768	Float	-1999...9999	☐	Heating current monitoring limit [A]. Depending on configuration, and apart from short-circuit monitoring, an overload test checks whether the heating current is above the adjusted current limit, or below the limit when the heating is switched off. The heating current is measured by means of a current transformer (accessory), and the current range can be adapted.

Signal

Name	r/w	Adr. Integer		real	Typ	Value/off		Description
St.HC	r	base 1dP 2dP 3dP	2070 10262 18454 26646	36908	Int	0...3	☐	Status of the heating current alarm. Displayable are heating current short-circuit and/or heating current alarm. Depending on configuration, the heating current alarm is either an interruption of heating current (I < limit value) or heating current overload (I > limit value).
HC	r	base 1dP 2dP 3dP	2071 10263 18455 26647	36910	Float	-1999...9999	☐	Measured heating current [A]. Apart from the short circuit test, and depending on configuration, an overcurrent test (current I > heating current limit) and an open circuit test (current I < heating current limit) is executed. The heating current is measured by means of a (separate) current transformer, whereby the input range can be scaled.
SSr	r	base 1dP 2dP 3dP	2072 10264 18456 26648	36912	Float	-1999...9999	☐	Measured current with SSr [A]. The heating current (SSR) is short circuited, if there is a current flow even though the controller output is switched off.Suggested remedy: check heating current circuit, replace solid-state relay if necessary.
St.Lim	r	base 1dP 2dP 3dP	2170 10362 18554 26746	37108	Enum	*Enum_LimStatus*		Limit value status: No alarm present or stored.

0	no alarm
1	latched alarm
2	A limit value has been exceeded.

5 Lim2

ConF

Name	r/w	Adr.	Integer	real	Typ	Value/off	Description
Fnc.2	r/w	base	2250	37268	Enum	*Enum_Fcn*	Activation and adjustment of the limit value alarm (e.g. for input circuit monitoring), e.g. with/without storage.
		1dP	10442				
		2dP	18634				
		3dP	26826				

Value	Description
0	No limit value monitoring.
1	measured value monitoring. The alarm signal is generated, if the limit is exceeded. If the measured value is within the limits (including hysteresis) again, this alarm signal is resetted.
2	Measured value monitoring + alarm status latch. An alarm signal is generated, if the limit is exceeded. A latched alarm signal remains latched until it is manually resetted.

Name	r/w	Adr.	Integer	real	Typ	Value/off	Description
Src.2	r/w	base	2251	37270	Enum	*Enum_Src*	Source for limit value. Selection of which value is to be monitored.
		1dP	10443				
		2dP	18635				
		3dP	26827				

Value	Description
0	Process value = absolute alarm
1	control deviation xw (process value - set-point) = relative alarm Note: Monitoring with the effective set-point Weff. For example using a ramp it is the changing set-point, not the target set-point of the ramp.
2	Control deviation Xw (= relative alarm) with suppression during start-up and setpoint changes. Limit value monitoring is continued as soon as the control deviation comes within the alarm limits again, at the latest after 10 * Tn.
6	effective set-point Weff. For example the ramp-function changes the effective set-point untill it matches the internal (target) set-point.
7	correcting variable y (controller output)
11	Control deviation Xw (= relative alarm) with suppression during start-up and setpoint change. Limit value monitoring is continued as soon as the control deviation comes within the alarm limits again.

PArA

Name	r/w	Adr.	Integer	real	Typ	Value/off		Description
L.2	r/w	base	2200	37168	Float	-1999...9999	☑	Lower limit value. The alarm is triggered if the value falls below the limit, and is reset with lower limit value plus hysteresis.
		1dP	10392					
		2dP	18584					
		3dP	26776					
H.2	r/w	base	2201	37170	Float	-1999...9999	☑	Upper limit value. The alarm is triggered if the value rises above the limit, and is reset with upper lower limit value plus hysteresis.
		1dP	10393					
		2dP	18585					
		3dP	26777					
HYS.2	r/w	base	2202	37172	Float	0...9999	☐	Hysteresis of the limit value. Switching difference for upper and lower limit value. The limit value must change by this amount (rise above upper limit or fall below lower limit) before the limit value alarm is reset.
		1dP	10394					
		2dP	18586					
		3dP	26778					

5 Lim2

Signal

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
St.Lim	r	base 1dP 2dP 3dP	2270 10462 18654 26846	37308	Enum	*Enum_LimStatus*	Limit value status: No alarm present or stored.

0	no alarm
1	latched alarm
2	A limit value has been exceeded.

6 Lim3

ConF

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
Fnc.3	r/w	base 1dP 2dP 3dP	2350 10542 18734 26926	37468	Enum	*Enum_Fcn*	Activation and adjustment of the limit value alarm (e.g. for input circuit monitoring), e.g. with/without storage.

0	No limit value monitoring.
1	measured value monitoring. The alarm signal is generated, if the limit is exceeded. If the measured value is within the limits (including hysteresis) again, this alarm signal is resetted.
2	Measured value monitoring + alarm status latch. An alarm signal is generated, if the limit is exceeded. A latched alarm signal remains latched until it is manually resetted.

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
Src.3	r/w	base 1dP 2dP 3dP	2351 10543 18735 26927	37470	Enum	*Enum_Src*	Source for limit value. Selection of which value is to be monitored.

0	Process value = absolute alarm
1	control deviation xw (process value - set-point) = relative alarm Note: Monitoring with the effective set-point Weff. For example using a ramp it is the changing set-point, not the target set-point of the ramp.
2	Control deviation Xw (= relative alarm) with suppression during start-up and setpoint changes. Limit value monitoring is continued as soon as the control deviation comes within the alarm limits again, at the latest after 10 * Tn.
6	effective set-point Weff. For example the ramp-function changes the effective set-point untill it matches the internal (target) set-point.
7	correcting variable y (controller output)
11	Control deviation Xw (= relative alarm) with suppression during start-up and setpoint change. Limit value monitoring is continued as soon as the control deviation comes within the alarm limits again.

PArA

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
L.3	r/w	base 1dP 2dP 3dP	2300 10492 18684 26876	37368	Float	-1999...9999 ☑	Lower limit value. The alarm is triggered if the value falls below the limit, and is reset with lower limit value plus hysteresis.

6 Lim3

PArA

Name	r/w	Adr. Integer		real	Typ	Value/off		Description
H.3	r/w	base 1dP 2dP 3dP	2301 10493 18685 26877	37370	Float	-1999...9999	☑	Upper limit value. The alarm is triggered if the value rises above the limit, and is reset with upper lower limit value plus hysteresis.
HYS.3	r/w	base 1dP 2dP 3dP	2302 10494 18686 26878	37372	Float	0...9999	☐	Hysteresis of the limit value. Switching difference for upper and lower limit value. The limit value must change by this amount (rise above upper limit or fall below lower limit) before the limit value alarm is reset.

Signal

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
St.Lim	r	base 1dP 2dP 3dP	2370 10562 18754 26946	37508	Enum	*Enum_LimStatus*	Limit value status: No alarm present or stored.

0	no alarm
1	latched alarm
2	A limit value has been exceeded.

7 LOGI

ConF

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
L_r	r/w	base 1dP 2dP 3dP	1051 9243 17435 25627	34870	Enum	*Enum_dInP1*	Local / remote switchover (Remote: Adjustment of all values via the front panel is blocked).

0	no function (switch-over via interface is possible)
1	always active
2	Digital Input DI1 switches
3	DI2 switches (only visible with OPTION)
4	DI3 switches (only visible with OPTION)

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
SP.2	r/w	base 1dP 2dP 3dP	1052 9244 17436 25628	34872	Enum	*Enum_dInP4*	Source of the control signal for activating the second (safety) setpoint (SP.2=) W2. Note: W2 is not restricted by the setpoint limits.

0	no function (switch-over via interface is possible)
2	Digital Input DI1 switches
3	DI2 switches (only visible with OPTION)
4	DI3 switches (only visible with OPTION)

7 LOGI

ConF

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
SP.E	r/w	base 1dP 2dP 3dP	1053 9245 17437 25629	34874	Enum	*Enum_dInP1*	Switching between internal set-point an external setpoint SP.E. The external SP.E is either the absolute set-point Wext or the offset to the set-point (dependent on instrument and configuration).

0	no function (switch-over via interface is possible)
1	always active
2	Digital Input DI1 switches
3	DI2 switches (only visible with OPTION)
4	DI3 switches (only visible with OPTION)

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
Y2	r/w	base 1dP 2dP 3dP	1054 9246 17438 25630	34876	Enum	*Enum_dInP3*	Source of the control signal for activating the second positioning output Y2. Activated Y2 = positioner control. Caution: The parameter 'positioning output Y2' must not be confused with the controller output Y2!

0	no function (switch-over via interface is possible)
2	Digital Input DI1 switches
3	DI2 switches (only visible with OPTION)
4	DI3 switches (only visible with OPTION)
6	Auto/manual key switches (A/M key)

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
mAn	r/w	base 1dP 2dP 3dP	1056 9248 17440 25632	34880	Enum	*Enum_dInp2*	Source of the control signal for auto/manual switchover. In the automatic mode, the controller is in charge. In the manual mode, the outputs can be varied independently of the process.

0	no function (switch-over via interface is possible)
1	always activated (manual station)
2	Digital Input DI1 switches
3	DI2 switches (only visible with OPTION)
4	DI3 switches (only visible with OPTION)
6	Auto/manual key switches (A/M key)

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
C.oFF	r/w	base 1dP 2dP 3dP	1057 9249 17441 25633	34882	Enum	*Enum_dInP3*	Source of the control signal for disabling all the controller outputs.Note: Forcing has priority, and remains active; alarm processing also remains active.

0	no function (switch-over via interface is possible)
2	Digital Input DI1 switches
3	DI2 switches (only visible with OPTION)
4	DI3 switches (only visible with OPTION)
6	Auto/manual key switches (A/M key)

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
m.Loc	r/w	base 1dP 2dP 3dP	1058 9250 17442 25634	34884	Enum	*Enum_dInp4*	Source of the control signal to disable the auto/manual key. If the A/M key is disabled, switchover to manual operation is not possible.

0	no function (switch-over via interface is possible)
2	Digital Input DI1 switches
3	DI2 switches (only visible with OPTION)
4	DI3 switches (only visible with OPTION)

7 LOGI

ConF

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
Err.r	r/w	base	1059	34886	Enum	*Enum_dInP3*	Source of the control signal for resetting all stored entries in the error list (the list contains all error messages and alarms). If an alarm is still present, i.e. the source of trouble has not been remedied, stored alarms cannot be acknowledged (reset).
		1dP	9251				
		2dP	17443				
		3dP	25635				

0	no function (switch-over via interface is possible)
2	Digital Input DI1 switches
3	DI2 switches (only visible with OPTION)
4	DI3 switches (only visible with OPTION)
6	Auto/manual key switches (A/M key)

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
P.run	r/w	base	1062	34892	Enum	*Enum_dInP6*	Source of the control signal for switching the programmer between Run and Stop. On units with a simple programmer (only 1 program), a stop immediately causes a reset, followed by a new start. With units that have been defined as program controllers (several programs), the program is stopped, and then continued.
		1dP	9254				
		2dP	17446				
		3dP	25638				

0	no function
2	Digital Input DI1 switches
3	DI2 switches (only visible with OPTION)
4	DI3 switches (only visible with OPTION)

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
di.Fn	r/w	base	1050	34868	Enum	*Enum_diFn*	Function of digital inputs (valid for all inputs)
		1dP	9242				
		2dP	17434				
		3dP	25626				

0	Basic setting 'Off': A permanent positive signal switches this function 'On', which is connected to the digital input. Removal of the signal switches the function 'Off' again.
1	Basic setting 'On': A permanent positive signal switches this function 'Off', which is connected to the digital input. Removal of the signal switches the function 'On' again.
2	Push-button function. Basic setting 'Off'. Only positive signals are effective. The first positive signal switches 'On'. Removal of the signal is necessary before the next positive signal can switch 'Off'.

Signal

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
St.Di	r	base	1070	34908	Int	0...7 ☑	Status of the digital inputs or of push-buttons (binary coded).
		1dP	9262				
		2dP	17454				
		3dP	25646				

Bit 0 Input 1
Bit 1 Input 2
Bit 2 Input 3
Bit 8 Status of 'F' key
Bit 9 Status of 'A/M' key
Bit 10 Status of 'Sel' key
Bit 11 Status of 'Down' key
Bit 12 Status of 'Up' key
Bit 13 Status of 'Loc' key

7 LOGI

Signal

Name	r/w	Adr. Integer		real	Typ	Value/off		Description
L-R	r/w	base 1dP 2dP 3dP	1080 9272 17464 25656	34928	Int	0...1	☐	Remote operation. Remote means that all values can only be adjusted via the interface. Adjustments via the front panel are blocked.
W_W2	r/w	base 1dP 2dP 3dP	1081 9273 17465 25657	34930	Int	0...1	☐	Signal for activating the second (safety) setpoint (SP.2=) W2. Note: Setpoint W2 is not restricted by the setpoint limits!
Wi_We	r/w	base 1dP 2dP 3dP	1082 9274 17466 25658	34932	Int	0...1	☐	Signal for activating the external setpoint value. SP.E is the external setpoint, or dependent on the device and configuration of the setpoint shift.
Y_Y2	r/w	base 1dP 2dP 3dP	1083 9275 17467 25659	34934	Int	0...1	☐	Signal for activating the 2nd output value Y2. With selected Y2, the output is operated as a positioner.Caution: Do not confuse the parameter 'fixed output Y2' with the controller output Y2!
A-M	r/w	base 1dP 2dP 3dP	1085 9277 17469 25661	34938	Int	0...1	☐	Signal for activating manual operation. In the manual mode, the controller provides output signals independent of the process.
C.Off	r/w	base 1dP 2dP 3dP	1086 9278 17470 25662	34940	Int	0...1	☐	Signal for disabling all the controller outputs. Note: Forcing has priority; alarm processing remains active.
L.AM	r/w	base 1dP 2dP 3dP	1087 9279 17471 25663	34942	Int	0...1	☐	Signal for disabling manual operation. Triggers a forced switchover to automatic mode, and disables the front panel A/M key (also if other functions have been assigned to the key).
Err.r	r/w	base 1dP 2dP 3dP	1088 9280 17472 25664	34944	Int	0...1	☐	Signal for resetting the entire error list. The error list contains all errors that are reported, e.g. device faults and limit values. It also contains queued as well as stored errors after their correction. The reset acknowledges all errors, whereby queued errors will reappear after the next error detection (measurement).
SSR.Res	r/w	base 1dP 2dP 3dP	1089 9281 17473 25665	34946	Int	0...1	☐	Reset of the alarm triggered by a solid-state relay (SSR). SSRs are mostly used for frequent switching of heating elements, because they have no mechanical contacts that can wear out. However, an unnoticed short circuit could lead to overheating of the machine.
Prg.R.S	r/w	base 1dP 2dP 3dP	1092 9284 17476 25668	34952	Int	0...1	☐	Signal for starting the programmer. On units with a simple programmer (only 1 program), a stop immediately causes a reset, followed by a new start. With units that have been defined as program controllers (several programs), the program is stopped, and then continued.

7 LOGI

Signal

Name	r/w	Adr. Integer		real	Typ	Value/off		Description
F.Di	r/w	base	1094	34956	Int	0...7	☐	Forcing of digital inputs. Forcing involves the external operation of at least one input. The instrument takes over this input value (preset value for inputs from a superordinate system, e.g. for a function test.)
		1dP	9286					
		2dP	17478					
		3dP	25670					

Bit 0 Forcing of digital input 1
Bit 1 Forcing of digital input 2
Bit 2 Forcing of digital input 3
Bit 3 Forcing of digital input 4
Bit 4 Forcing of digital input 5

8 ohnE

PArA

Name	r/w	Adr. Integer		real	Typ	Value/off		Description
Conf	r/w	base	1	32770	Int	0...2	☐	Start/Stop and abortion of the configuration mode 0 = End of configuration 1 = Start of configuration 2 = Abort configuration
		1dP	8193					
		2dP	16385					
		3dP	24577					

Signal

Name	r/w	Adr. Integer		real	Typ	Value/off		Description
UPD	r/w	base	95	32958	Enum	*Enum_Aenderungsflag*		Status message indicating that parameter / configuration have been changed via the front panel.
		1dP	8287					
		2dP	16479					
		3dP	24671					

0	No change via the front panel keys.
1	A change has been made via the front panel keys, which must be processed.

Name	r/w	Adr. Integer		real	Typ	Value/off		Description
Hw.Opt	r	base	200	33168	Int	0...65535	☑	
		1dP	8392					
		2dP	16584					
		3dP	24776					
Sw.Op	r	base	201	33170	Int	0...255	☐	Software version XY Major and Minor Release (e.g. 21 = Version 2.1). The software version specifies the firmware in the unit. For the correct interaction of E-Tool and device, it must match the operating version (OpVersion) in the E-Tool.
		1dP	8393					
		2dP	16585					
		3dP	24777					
Bed.V	r	base	202	33172	Int	0...255	☐	Operating version (numeric value). For the correct interaction of E-Tool and device, the software version and operating version must match.
		1dP	8394					
		2dP	16586					
		3dP	24778					
Unit	r	base	203	33174	Int	0...255	☐	Identification of the device.
		1dP	8395					
		2dP	16587					
		3dP	24779					

8 ohnE

Signal

Name	r/w	Adr. Integer		real	Typ	Value/off		Description
S.Vers	r	base 1dP 2dP 3dP	204 8396 16588 24780	33176	Int	100...255	☐	The sub-version number is given as an additional index for precise definition of software version.
Uident	r	base 1dP 2dP 3dP	910 9102 17294 25486	34588	Text	...	☐	Device identification. Via this Modbus address, up to 14 data units (28 bytes) can be defined. Bytes 1 - 15 order number of the device Bytes 16 - 19 Ident number 1 Bytes 20 + 21 Ident number 2 Bytes 22 - 25 OEM number Bytes 26 - 28 Software order number
St.Ala	r	base 1dP 2dP 3dP	250 8442 16634 24826	33268	Int	0...31	☐	Alarm status: Bit-wise coded status of the individual alarms, e.g. exceeded limit value or Loop.
						Bit 0 Existing/stored exceeded limit 1 Bit 1 Existing/stored exceeded limit 2 Bit 2 Existing/stored exceeded limit 3 Bit 3 Not used Bit 4 Existing/stored loop alarm Bit 5 Existing/stored heating current alarm Bit 6 Existing/stored SSR alarm Bit 7 Not used Bit 8 Existing exceeded limit 1 Bit 9 Existing exceeded limit 2 Bit 10 Existing exceeded limit 3 Bit 11 Not used Bit 12 Existing loop alarm Bit 13 Existing heating current alarm Bit 14 Existing SSR alarm Bit 15 Not used		
St.Do	r	base 1dP 2dP 3dP	251 8443 16635 24827	33270	Int	0...31	☐	Status of the digital outputs Bit 0 digital output 1 Bit 1 digital output 2 Bit 2 digital output 3 Bit 3 digital output 4 Bit 4 digital output 5 Bit 5 digital output 6

8 ohnE

Signal

Name	r/w	Adr. Integer		real	Typ	Value/off		Description
St.Ain	r	base	252	33272	Int	0...7	☐	Bit-coded status of the analog input (fault, e.g. short circuit)
		1dP	8444					
		2dP	16636					
		3dP	24828					
								Bit 0 Break at Input 1 Bit 1 Reversed polarity at Input 1 Bit 2 Short circuit at Input 1 Bit 3 Not used Bit 4 Break at Input 2 Bit 5 Reversed polarity at Input 2 Bit 6 Short-circuit at Input 2 Bit 7 Not used Bit 8 Break at Input 3 (only KS 90) Bit 9 Reversed polarity at Input 3 (only KS 90) Bit 10 Short-circuit at Input 3 (only KS 90) Bit 11 Not used
St.Di	r	base	253	33274	Int	0...7	☐	Status of the digital inputs or of push-buttons (binary coded).
		1dP	8445					
		2dP	16637					
		3dP	24829					
								Bit 0 Input 1 Bit 1 Input 2 Bit 2 Input 3 Bit 8 Status of 'F' key Bit 9 Status of 'A/M' key Bit 10 Status of 'Sel' key Bit 11 Status of 'Down' key Bit 12 Status of 'Up' key Bit 13 Status of 'Loc' key
F.Di	r/w	base	303	33374	Int	0...1	☐	Forcing of digital inputs. Forcing involves the external operation of at least one input. The instrument takes over this input value (preset value for inputs from a superordinate system, e.g. for a function test.)
		1dP	8495					
		2dP	16687					
		3dP	24879					
								Bit 0 Forcing of digital input 1 Bit 1 Forcing of digital input 2 Bit 2 Forcing of digital input 3 Bit 3 Forcing of digital input 4 Bit 4 Forcing of digital input 5
F.Do	r/w	base	304	33376	Int	0...15	☐	Forcing of digital outputs. Forcing involves the external operation of at least one output. The instrument has no influence on this output (use of free outputs by superordinate system).
		1dP	8496					
		2dP	16688					
		3dP	24880					

9 ohnE1

Signal

Name	r/w	Adr. Integer		real	Typ	Value/off		Description
In.1	r	base 1dP 2dP 3dP	232 8424 16616 24808	33232	Float	-1999...9999	☐	Measurement value after the measurement value correction (e.g. with offset or 2-point correction, and scaling).
In.1r	r	base 1dP 2dP 3dP	240 8432 16624 24816	33248	Float	-1999...9999	☐	Measurement value before the measurement value correction (unprocessed).
F.Inp	r/w	base 1dP 2dP 3dP	300 8492 16684 24876	33368	Float	-1999...9999	☐	Forcing the value for an analog input INP. Forcing involves the external operation of an input. The instrument takes over the value at this input like a measurement value (preset value for inputs from a superordinate system, e.g. for a function test.)

10 ohnE2

Signal

Name	r/w	Adr. Integer		real	Typ	Value/off		Description
In.2	r	base 1dP 2dP 3dP	233 8425 16617 24809	33234	Float	-1999...9999	☐	Measurement value after the measurement value correction (e.g. with offset or 2-point correction, and scaling).
In.2r	r	base 1dP 2dP 3dP	241 8433 16625 24817	33250	Float	-1999...9999	☐	Measurement value before the measurement value correction (unprocessed).
F.Inp	r/w	base 1dP 2dP 3dP	301 8493 16685 24877	33370	Float	-1999...9999	☐	Forcing the value for an analog input INP. Forcing involves the external operation of an input. The instrument takes over the value at this input like a measurement value (preset value for inputs from a superordinate system, e.g. for a function test.)

11 ohnE3

Signal

Name	r/w	Adr. Integer		real	Typ	Value/off		Description
F.Out1	r/w	base 1dP 2dP 3dP	305 8497 16689 24881	33378	Float	0...120	☐	Forcing value of the analog output. Forcing involves the external operation of an output, i.e. the instrument has no influence on this output. (Used for the operation of free outputs e.g. by a supervisory PLC.)

12 othr

ConF

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
bAud	r/w	base	180	33128	Enum	*Enum_Baud*	Bit rate of the interface (only visible with OPTION). The bit rate determines the transmission speed.
		1dP	8372				
		2dP	16564				
		3dP	24756				

0	2400 Baud
1	4800 Baud
2	9600 Baud
3	19200 Baud

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
Addr	r/w	base	181	33130	Int	1...247 ☐	Address on the interface (only visible with OPTION)
		1dP	8373				
		2dP	16565				
		3dP	24757				
PrtY	r/w	base	182	33132	Enum	*Enum_Parity*	Parity of data on the interface (only visible with OPTION). Simple possibility of checking that transferred data is correct.
		1dP	8374				
		2dP	16566				
		3dP	24758				

0	No parity, with 2 stop bits.
1	even parity
2	odd parity
3	no parity (1 stop bit)

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
dELY	r/w	base	183	33134	Int	0...200 ☐	Response delay [ms] (only visible with OPTION). Additional delay time before the received message may be answered on the Modbus. (Might be necessary, if the same line is used for transmit/receive.)
		1dP	8375				
		2dP	16567				
		3dP	24759				
Unit	r/w	base	170	33108	Enum	*Enum_Unit*	Physical unit (temperature), f.e.°C
		1dP	8362				
		2dP	16554				
		3dP	24746				

0	without unit
1	°C
2	°F

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
dP	r/w	base	171	33110	Enum	*Enum_dP*	Decimal point (max. no of decimals). Format of the measured value display.
		1dP	8363				
		2dP	16555				
		3dP	24747				

0	no digit behind the decimal point
1	Display has one decimal.
2	Display has two decimals.
3	Display has three decimals.

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
C.dEL	r/w	base	184	33136	Int	0...200 ☐	For both interfaces, Modbus only. Additional acceptable delay time between 2 received bytes, before "end of message" is assumed. This time is needed if data is not transmitted continousely by the modem.
		1dP	8376				
		2dP	16568				
		3dP	24760				

12 othr

ConF

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
FrEq	r/w	base	150	33068	Enum	*Enum_FrEq*	Switchover of the applied mains frequency 50 / 60 Hz, thereby better adaptation of the input filter for hum suppression.
		1dP	8342				
		2dP	16534				
		3dP	24726				

0	Mains frequency is 50 Hz.
1	Mains frequency is 60 Hz.

Signal

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
E.1	r/w	base	210	33188	Enum	*Defect*	Err 1 (internal error) Contact Service.
		1dP	8402				
		2dP	16594				
		3dP	24786				

0	No fault exists (Reset).
2	The device is defective.

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
E.2	r/w	base	211	33190	Enum	*Problem*	Err 2 (internal error, resettable) (As a process value via fieldbus interface not writable!)
		1dP	8403				
		2dP	16595				
		3dP	24787				

0	No fault, resetting possible (Reset).
1	A fault has occurred and has been stored.

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
FbF.1	r/w	base	212	33192	Enum	*Break*	Sensor break at input INP1. Typical causes and suggested remedies: Sensor fault: replace INP1 sensor. Wiring fault: check connections of INP1. (As a process value via fieldbus interface not writable!)
		1dP	8404				
		2dP	16596				
		3dP	24788				

0	No fault, resetting of the sensor break alarm possible (Reset).
1	The sensor fault alarm has been triggered and stored; the fault is no longer present. The operator must acknowledge the error message in order to delete it from the error list.
2	Sensor break: The sensor is defective or there is a wiring fault.

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
Sht.1	r/w	base	213	33194	Enum	*Short*	Short circuit at input INP1. Typical causes and suggested remedies: Sensor fault: replace INP1 sensor. Wiring fault: check connections of INP1. (As a process value via fieldbus interface not writable!)
		1dP	8405				
		2dP	16597				
		3dP	24789				

0	No fault, resetting of the short-circuit alarm possible (Reset).
1	A short-circuit fault has occurred and has been stored.
2	A short-circuit fault has occurred.

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
POL.1	r/w	base	214	33196	Enum	*Polarity*	Incorrect polarity at input INP1. Suggested remedy: reverse the polarity at INP1. (As a process value via fieldbus interface not writable!)
		1dP	8406				
		2dP	16598				
		3dP	24790				

0	No fault, resetting of the incorrect polarity alarm possible (Reset).
1	An incorrect polarity fault has occurred and has been stored.
2	Incorrect polarity. The wiring of the input circuit is not correct.

12 othr

Signal

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
FbF.2	r/w	base 1dP 2dP 3dP	215 8407 16599 24791	33198	Enum	*Break*	Sensor break at input INP2. Typical causes and suggested remedies: Sensor fault: replace INP2 sensor. Wiring fault: check connections of INP2. (As a process value via fieldbus interface not writable!)

0	No fault, resetting of the sensor break alarm possible (Reset).
1	The sensor fault alarm has been triggered and stored; the fault is no longer present. The operator must acknowledge the error message in order to delete it from the error list.
2	Sensor break: The sensor is defective or there is a wiring fault.

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
Sht.2	r/w	base 1dP 2dP 3dP	216 8408 16600 24792	33200	Enum	*Short*	Short circuit at input INP2. Typical causes and suggested remedies: Sensor fault: replace INP2 sensor. Wiring fault: check connections of INP2. (As a process value via fieldbus interface not writable!)

0	No fault, resetting of the short-circuit alarm possible (Reset).
1	A short-circuit fault has occurred and has been stored.
2	A short-circuit fault has occurred.

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
POL.2	r/w	base 1dP 2dP 3dP	217 8409 16601 24793	33202	Enum	*Polarity*	Incorrect polarity at input INP2. Suggested remedy: reverse the polarity at INP2. (As a process value via fieldbus interface not writable!)

0	No fault, resetting of the incorrect polarity alarm possible (Reset).
1	An incorrect polarity fault has occurred and has been stored.
2	Incorrect polarity. The wiring of the input circuit is not correct.

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
HCA	r/w	base 1dP 2dP 3dP	218 8410 16602 24794	33204	Enum	*HeatCurr*	Heating current alarm.Possible fault s are an open heating current circuit with current I < heating current limit, or current I > heating current limit (depending on configuration), or defective heater band.Suggested remedy: check heating current circuit, replace heater band if necessary. (As a process value via fieldbus interface not writable!)

0	No fault, resetting of the heating current alarm possible (Reset).
1	A heating current fault has occurred and has been stored.

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
SSr	r/w	base 1dP 2dP 3dP	219 8411 16603 24795	33206	Enum	*Short*	Alarm message: SSr Possible causes: a current flow in the heating circuit although controller is 'off', or the SSR is defective. Suggested remedy: check heating current circuit, replace the solid-state relay, if necessary. (As a process value via fieldbus interface not writable!)

0	No fault, resetting of the short-circuit alarm possible (Reset).
1	A short-circuit fault has occurred and has been stored.
2	A short-circuit fault has occurred.

12 othr

Signal

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
LooP	r/w	base 1dP 2dP 3dP	220 8412 16604 24796	33208	Enum	*LoopAlarm*	Alarm message: LooP Possible causes: faulty or incorrectly connected input circuit, or output not connected correctly. Suggested remedy: check heating or cooling circuit, check sensor function and replace if necessary, check controller and output switching actuator. (As a process value via fieldbus interface not writable!)

0	No fault, resetting of the loop alarm possible (Reset).
1	A control loop fault has occurred and has been stored.
2	A control loop fault has occurred, there was no clear process response following a step change of the output.

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
AdA.H	r/w	base 1dP 2dP 3dP	221 8413 16605 24797	33210	Enum	*Tune*	Error message from "heating" self-tuning and reason for aborted tuning attempt. Hints for trouble-shooting: Check operating sense of actuator. Is the loop closed? Is there an output limit? Adapt the setpoint. Increase step output for Yopt. (As a process value via fieldbus interface not writable!)

0	no error
3	Process responds in the wrong direction. Possible remedy: Check the output signal sense (inverse <-> direct), and re-configure the controller if necessary (inverse <-> direct).
4	No response from the process. Perhaps the control loop is open. Possible remedy: Check sensor, connections, and process.
5	The process value turning point of the step response is too low. Possible remedy: Increase the permitted step output range, i.e. increase the parameter Y.Hi ('heating') or reduce the parameter Y.Lo ('cooling').
6	Self-tuning was aborted due to the risk of an exceeded setpoint. Possible remedy: Repeat the attempt with an increased setpoint reserve.
7	The step output change is not large enough (minimum change > 5 %). Possible remedy: Increase the permitted step output range, i.e. increase the parameter Y.Hi ('heating') or reduce the parameter Y.Lo ('cooling').
8	Setpoint reserve must be given before generating the step output change. Possible remedy: decrease set-point range, change set-point, or change process value.
9	The pulse response attempt has failed. No useful parameters were determined. Perhaps the control loop is open. Possible remedy: Check sensor, connections, and process.

12 othr

Signal

Name	r/w	Adr.	Integer	real	Typ	Value/off	Description
AdA.C	r/w	base	222	33212	Enum	*Tune*	Error message from "cooling" self-tuning and reason for aborted tuning attempt. Hints for trouble-shooting: Check operating sense of actuator. Is the loop closed? Is there an output limit? Adapt the setpoint. Increase step output for Yopt. (As a process value via fieldbus interface not writable!)
		1dP	8414				
		2dP	16606				
		3dP	24798				

0	no error
3	Process responds in the wrong direction. Possible remedy: Check the output signal sense (inverse <-> direct), and re-configure the controller if necessary (inverse <-> direct).
4	No response from the process. Perhaps the control loop is open. Possible remedy: Check sensor, connections, and process.
5	The process value turning point of the step response is too low. Possible remedy: Increase the permitted step output range, i.e. increase the parameter Y.Hi ('heating') or reduce the parameter Y.Lo ('cooling').
6	Self-tuning was aborted due to the risk of an exceeded setpoint. Possible remedy: Repeat the attempt with an increased setpoint reserve.
7	The step output change is not large enough (minimum change > 5 %). Possible remedy: Increase the permitted step output range, i.e. increase the parameter Y.Hi ('heating') or reduce the parameter Y.Lo ('cooling').
8	Setpoint reserve must be given before generating the step output change. Possible remedy: decrease set-point range, change set-point, or change process value.
9	The pulse response attempt has failed. No useful parameters were determined. Perhaps the control loop is open. Possible remedy: Check sensor, connections, and process.

Name	r/w	Adr.	Integer	real	Typ	Value/off	Description
Lim.1	r/w	base	223	33214	Enum	*Limit*	Limit value 1 exceeded. Hint for trouble-shooting: check the process. (As a process value via fieldbus interface not writable!)
		1dP	8415				
		2dP	16607				
		3dP	24799				

0	No fault, resetting of the limit value alarm possible (Reset).
1	The limit value has been exceeded, and the fault has been stored.
2	The limit value has been exceeded; the monitored (measurement) value is outside the set limits.

Name	r/w	Adr.	Integer	real	Typ	Value/off	Description
Lim.2	r/w	base	224	33216	Enum	*Limit*	Limit value 2 exceeded. Hint for trouble-shooting: check the process. (As a process value via fieldbus interface not writable!)
		1dP	8416				
		2dP	16608				
		3dP	24800				

0	No fault, resetting of the limit value alarm possible (Reset).
1	The limit value has been exceeded, and the fault has been stored.
2	The limit value has been exceeded; the monitored (measurement) value is outside the set limits.

Name	r/w	Adr.	Integer	real	Typ	Value/off	Description
Lim.3	r/w	base	225	33218	Enum	*Limit*	Limit value 3 exceeded. Hint for trouble-shooting: check the process. (As a process value via fieldbus interface not writable!)
		1dP	8417				
		2dP	16609				
		3dP	24801				

0	No fault, resetting of the limit value alarm possible (Reset).
1	The limit value has been exceeded, and the fault has been stored.
2	The limit value has been exceeded; the monitored (measurement) value is outside the set limits.

12 othr

Signal

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
InF.1	r/w	base	226	33220	Enum	*Time*	Message from the operating hours counter that the preset no. of hours for this maintenance period has been reached. The op-hours counter for the maintenance period is reset when this message is acknowledged. Counting the operating hours is used for preventive maintenance. - Acknowledge the error to reset it. (As a process value via fieldbus interface not writable!)
		1dP	8418				
		2dP	16610				
		3dP	24802				

0	No signal, resetting of the time limit signal possible (Reset).
1	Operating hours - limit value (maintenance period) reached: please acknowledge.

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
InF.2	r/w	base	227	33222	Enum	*Switch*	Message from the switching cycle counter that the preset no. of switch cycles for this maintenance period has been reached. The cycle counter for the maintenance period is reset when this message is acknowledged. Counting the switching cycles is used for preventive maintenance. - Acknowledge the error to reset it. (As a process value via fieldbus interface not writable!)
		1dP	8419				
		2dP	16611				
		3dP	24803				

0	No error message, resetting of the switching cycle counter possible (Reset).
1	Set limit of the switching cycle counter (maintenance period) has been reached: please acknowledge.

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
E.4	r/w	base	228	33224	Enum	*Problem*	Hardware fault.Cause: Code number and hardware are not identical. Remedy: Contact Service. (As a process value via fieldbus interface not writable!)
		1dP	8420				
		2dP	16612				
		3dP	24804				

0	No fault, resetting possible (Reset).
1	A fault has occurred and has been stored.

13 Out.1

ConF

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
O.Act	r/w	base	4150	41068	Enum	*Enum_OAct*	Operating sense of the switching output. Direct: Active function (e.g. limit value) switches the output ON; Inverse: Active function (e.g. limit value) switches the output OFF.
		1dP	12342				
		2dP	20534				
		3dP	28726				

0	direct / normally open
1	inverse / normally closed

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
Y.1	r/w	base	4151	41070	Enum	*Enum_Y1*	Output function: Controller output Y1
		1dP	12343				
		2dP	20535				
		3dP	28727				

0	not active
1	This output provides the controller output Y1.

13 Out.1

ConF

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
Y.2	r/w	base	4152	41072	Enum	*Enum_Y2*	Output function: Controller output Y2. Caution: Do not confuse the controller output Y2 with the parameter 'Fixed output Y2' !
		1dP	12344				
		2dP	20536				
		3dP	28728				
						0	not active
						1	This output provides the controller output Y2.
Lim.1	r/w	base	4153	41074	Enum	*Enum_Lim1*	Output function: Signal limit 1
		1dP	12345				
		2dP	20537				
		3dP	28729				
						0	not active
						1	The output is activated by an alarm from limit value 1.
Lim.2	r/w	base	4154	41076	Enum	*Enum_Lim2*	Output function: Signal limit 2
		1dP	12346				
		2dP	20538				
		3dP	28730				
						0	not active
						1	The output is activated by an alarm from limit value 2.
Lim.3	r/w	base	4155	41078	Enum	*Enum_Lim3*	Output function: Signal limit 3
		1dP	12347				
		2dP	20539				
		3dP	28731				
						0	not active
						1	The output is activated by an alarm from limit value 3.
LP.AL	r/w	base	4157	41082	Enum	*Enum_OUT_LPAL*	Output function: Signal Interruption alarm (LOOP) The overall control loop is monitored and the process value has to change with an output signal of maximum value, else loop alarm is generated.
		1dP	12349				
		2dP	20541				
		3dP	28733				
						0	not active
						1	The loop alarm (= open loop alarm) is assigned to this output.
HC.AL	r/w	base	4158	41084	Enum	*Enum_OUT_HCAL*	Output function: Signal Heat current alarm. Either break (= current I < heating current limit) can be monitored or overload (= current I > heating current limit), dependent on configuration.
		1dP	12350				
		2dP	20542				
		3dP	28734				
						0	not active
						1	The heating current alarm is assigned to this output.
HC.SC	r/w	base	4159	41086	Enum	*Enum_HCSC*	Output function: Signal Solid-state relay (SSR) short circuit. The short circuit alarm of the SSR is triggered, if a current is detected in the heating circuit, although the controller output is switched off.
		1dP	12351				
		2dP	20543				
		3dP	28735				
						0	not active
						1	Output activated by an SSR fault.

13 Out.1

ConF

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
timE	r/w	base 1dP 2dP 3dP	4160 12352 20544 28736	41088	Enum	*Enum_time*	Output function: Signal Timer running. This message is generated by the setpoint processing, if a timer mode has been configured, and the time has elapsed.
						0	not active
						1	activated
t.End	r/w	base 1dP 2dP 3dP	4176 12368 20560 28752	41120	Enum	*Enum_TEnd*	Output function: Signal Timer end. This message is available when the timer has been completed (only when configured as a timer).
						0	not active
						1	This output is activated by the message 'Timer end'.
P.End	r/w	base 1dP 2dP 3dP	4161 12353 20545 28737	41090	Enum	*Enum_PEnd*	Output function: Signal Program end. This message is available when the program has been completed (only when configured as a program controller).
						0	not active
						1	This output is activated by the message 'Program end'.
FAi.1	r/w	base 1dP 2dP 3dP	4162 12354 20546 28738	41092	Enum	*Enum_FAi1*	Output function: Signal INP1 fault. The fail signal is generated, if a fault occurs at the analog Input INP1.
						0	not active
						1	The output sends the error message 'INP1 fault'.
FAi.2	r/w	base 1dP 2dP 3dP	4163 12355 20547 28739	41094	Enum	*Enum_FAi2*	Output function: Signal INP2 fault. The fail signal is generated, if a fault occurs at the analog Input INP2.
						0	not active
						1	The output sends the error message 'INP2 fault'.

Signal

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
Out1	r	base 1dP 2dP 3dP	4180 12372 20564 28756	41128	Enum	*Enum_Ausgang*	Status of the digital output
						0	off
						1	on

13 Out.1

Signal

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
F.Do1	r/w	base	4181	41130	Enum	*Enum_Ausgang*	Forcing of this digital output. Forcing involves the external operation of an output. The instrument has no influence on this output (use of free outputs by superordinate system).
		1dP	12373				
		2dP	20565				
		3dP	28757				

0	off
1	on

14 Out.2

ConF

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
O.Act	r/w	base	4250	41268	Enum	*Enum_OAct*	Operating sense of the switching output. Direct: Active function (e.g. limit value) switches the output ON; Inverse: Active function (e.g. limit value) switches the output OFF.
		1dP	12442				
		2dP	20634				
		3dP	28826				

0	direct / normally open
1	inverse / normally closed

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
Y.1	r/w	base	4251	41270	Enum	*Enum_Y1*	Output function: Controller output Y1
		1dP	12443				
		2dP	20635				
		3dP	28827				

0	not active
1	This output provides the controller output Y1.

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
Y.2	r/w	base	4252	41272	Enum	*Enum_Y2*	Output function: Controller output Y2. Caution: Do not confuse the controller output Y2 with the parameter 'Fixed output Y2' !
		1dP	12444				
		2dP	20636				
		3dP	28828				

0	not active
1	This output provides the controller output Y2.

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
Lim.1	r/w	base	4253	41274	Enum	*Enum_Lim1*	Output function: Signal limit 1
		1dP	12445				
		2dP	20637				
		3dP	28829				

0	not active
1	The output is activated by an alarm from limit value 1.

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
Lim.2	r/w	base	4254	41276	Enum	*Enum_Lim2*	Output function: Signal limit 2
		1dP	12446				
		2dP	20638				
		3dP	28830				

0	not active
1	The output is activated by an alarm from limit value 2.

14 Out.2

ConF

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
Lim.3	r/w	base	4255	41278	Enum	*Enum_Lim3*	Output function: Signal limit 3
		1dP	12447				
		2dP	20639				
		3dP	28831				

0	not active
1	The output is activated by an alarm from limit value 3.

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
LP.AL	r/w	base	4257	41282	Enum	*Enum_OUT_LPAL*	Output function: Signal Interruption alarm (LOOP) The overall control loop is monitored and the process value has to change with an output signal of maximum value, else loop alarm is generated.
		1dP	12449				
		2dP	20641				
		3dP	28833				

0	not active
1	The loop alarm (= open loop alarm) is assigned to this output.

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
HC.AL	r/w	base	4258	41284	Enum	*Enum_OUT_HCAL*	Output function: Signal Heat current alarm. Either break (= current I < heating current limit) can be monitored or overload (= current I > heating current limit), dependent on configuration.
		1dP	12450				
		2dP	20642				
		3dP	28834				

0	not active
1	The heating current alarm is assigned to this output.

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
HC.SC	r/w	base	4259	41286	Enum	*Enum_HCSC*	Output function: Signal Solid-state relay (SSR) short circuit. The short circuit alarm of the SSR is triggered, if a current is detected in the heating circuit, although the controller output is switched off.
		1dP	12451				
		2dP	20643				
		3dP	28835				

0	not active
1	Output activated by an SSR fault.

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
timE	r/w	base	4260	41288	Enum	*Enum_time*	Output function: Signal Timer running. This message is generated by the setpoint processing, if a timer mode has been configured, and the time has elapsed.
		1dP	12452				
		2dP	20644				
		3dP	28836				

0	not active
1	activated

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
t.End	writ	base	4276	41320	Enum	*Enum_TEnd*	Output function: Signal Timer end. This message is available when the timer has been completed (only when configured as a timer).
		1dP	12468				
		2dP	20660				
		3dP	28852				

0	not active
1	This output is activated by the message 'Timer end'.

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
P.End	r/w	base	4261	41290	Enum	*Enum_PEnd*	Output function: Signal Program end. This message is available when the program has been completed (only when configured as a program controller).
		1dP	12453				
		2dP	20645				
		3dP	28837				

0	not active
1	This output is activated by the message 'Program end'.

14 Out.2

ConF

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
FAi.1	r/w	base 1dP 2dP 3dP	4262 12454 20646 28838	41292	Enum	*Enum_FAi1*	Output function: Signal INP1 fault. The fail signal is generated, if a fault occurs at the analog Input INP1.

0	not active
1	The output sends the error message 'INP1 fault'.

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
FAi.2	r/w	base 1dP 2dP 3dP	4263 12455 20647 28839	41294	Enum	*Enum_FAi2*	Output function: Signal INP2 fault. The fail signal is generated, if a fault occurs at the analog Input INP2.

0	not active
1	The output sends the error message 'INP2 fault'.

Signal

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
Out2	r	base 1dP 2dP 3dP	4280 12472 20664 28856	41328	Enum	*Enum_Ausgang*	Status of the digital output

0	off
1	on

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
F.Do2	r/w	base 1dP 2dP 3dP	4281 12473 20665 28857	41330	Enum	*Enum_Ausgang*	Forcing of this digital output. Forcing involves the external operation of an output. The instrument has no influence on this output (use of free outputs by superordinate system).

0	off
1	on

15 Out.3

ConF

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
O.tYP	r/w	base 1dP 2dP 3dP	4370 12562 20754 28946	41508	Enum	*Enum_OtYP*	Signal type selection OUT

0	Relay / logic
1	0 ... 20 mA continuous
2	4 ... 20 mA continuous
3	0...10 V continuous
4	2...10 V continuous
5	transmitter supply

15 Out.3

ConF

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
O.Act	r/w	base 1dP 2dP 3dP	4350 12542 20734 28926	41468	Enum	*Enum_OAct*	Operating sense of the switching output. Direct: Active function (e.g. limit value) switches the output ON; Inverse: Active function (e.g. limit value) switches the output OFF.

0	direct / normally open
1	inverse / normally closed

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
Y.1	r/w	base 1dP 2dP 3dP	4351 12543 20735 28927	41470	Enum	*Enum_Y1*	Output function: Controller output Y1

0	not active
1	This output provides the controller output Y1.

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
Y.2	r/w	base 1dP 2dP 3dP	4352 12544 20736 28928	41472	Enum	*Enum_Y2*	Output function: Controller output Y2. Caution: Do not confuse the controller output Y2 with the parameter 'Fixed output Y2' !

0	not active
1	This output provides the controller output Y2.

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
Lim.1	r/w	base 1dP 2dP 3dP	4353 12545 20737 28929	41474	Enum	*Enum_Lim1*	Output function: Signal limit 1

0	not active
1	The output is activated by an alarm from limit value 1.

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
Lim.2	r/w	base 1dP 2dP 3dP	4354 12546 20738 28930	41476	Enum	*Enum_Lim2*	Output function: Signal limit 2

0	not active
1	The output is activated by an alarm from limit value 2.

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
Lim.3	r/w	base 1dP 2dP 3dP	4355 12547 20739 28931	41478	Enum	*Enum_Lim3*	Output function: Signal limit 3

0	not active
1	The output is activated by an alarm from limit value 3.

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
LP.AL	r/w	base 1dP 2dP 3dP	4357 12549 20741 28933	41482	Enum	*Enum_OUT_LPAL*	Output function: Signal Interruption alarm (LOOP) The overall control loop is monitored and the process value has to change with an output signal of maximum value, else loop alarm is generated.

0	not active
1	The loop alarm (= open loop alarm) is assigned to this output.

15 Out.3

• ConF

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
HC.AL	r/w	base 1dP 2dP 3dP	4358 12550 20742 28934	41484	Enum	*Enum_OUT_HCAL*	Output function: Signal Heat current alarm. Either break (= current I < heating current limit) can be monitored or overload (= current I > heating current limit), dependent on configuration.
						0	not active
						1	The heating current alarm is assigned to this output.
HC.SC	r/w	base 1dP 2dP 3dP	4359 12551 20743 28935	41486	Enum	*Enum_HCSC*	Output function: Signal Solid-state relay (SSR) short circuit. The short circuit alarm of the SSR is triggered, if a current is detected in the heating circuit, although the controller output is switched off.
						0	not active
						1	Output activated by an SSR fault.
timE	r/w	base 1dP 2dP 3dP	4360 12552 20744 28936	41488	Enum	*Enum_time*	Output function: Signal Timer running. This message is generated by the setpoint processing, if a timer mode has been configured, and the time has elapsed.
						0	not active
						1	activated
t.End	r/w	base 1dP 2dP 3dP	4376 12568 20760 28952	41520	Enum	*Enum_TEnd*	Output function: Signal Timer end. This message is available when the timer has been completed (only when configured as a timer).
						0	not active
						1	This output is activated by the message 'Timer end'.
P.End	r/w	base 1dP 2dP 3dP	4361 12553 20745 28937	41490	Enum	*Enum_PEnd*	Output function: Signal Program end. This message is available when the program has been completed (only when configured as a program controller).
						0	not active
						1	This output is activated by the message 'Program end'.
FAi.1	r/w	base 1dP 2dP 3dP	4362 12554 20746 28938	41492	Enum	*Enum_FAi1*	Output function: Signal INP1 fault. The fail signal is generated, if a fault occurs at the analog Input INP1.
						0	not active
						1	The output sends the error message 'INP1 fault'.
FAi.2	r/w	base 1dP 2dP 3dP	4363 12555 20747 28939	41494	Enum	*Enum_FAi2*	Output function: Signal INP2 fault. The fail signal is generated, if a fault occurs at the analog Input INP2.
						0	not active
						1	The output sends the error message 'INP2 fault'.

15 Out.3

ConF

Name	r/w	Adr. Integer		real	Typ	Value/off		Description
Out.0	r/w	base 1dP 2dP 3dP	4371 12563 20755 28947	41510	Float	-1999...9999	☐	Lower scaling limit of the analog output (corresponds to 0%). If current and voltage signals are used as output values, the display can be scaled to the output value in the Parameter Level. The output value of the lower scaling point is indicated in the respective electrical unit (mA / V).
Out.1	r/w	base 1dP 2dP 3dP	4372 12564 20756 28948	41512	Float	-1999...9999	☐	Upper scaling limit of the analog output (corresponds to 100%). If current and voltage signals are used as output values, the display can be scaled to the output value in the Parameter Level. The output value of the upper scaling point is indicated in the respective electrical unit (mA / V).
O.Src	r/w	base 1dP 2dP 3dP	4373 12565 20757 28949	41514	Enum	*Enum_OSrc*		Signal source of the analog output (visible not with all output signal types O.TYP).

0	not used
1	Controller output y1 (continuous)
2	Controller output y2 (continuous)
3	process value
4	The effective setpoint Weff, which is used for control. Example: The gradient changes the effective setpoint until it reaches the internal (target) setpoint.
5	control deviation xw (process value - set-point)= relative alarm Note: Monitoring with the effective set-point Weff. For example using a ramp it is the changing set-point, not the target set-point of the ramp.

Signal

Name	r/w	Adr. Integer		real	Typ	Value/off		Description
Out1	r	base 1dP 2dP 3dP	4380 12572 20764 28956	41528	Enum	*Enum_Ausgang*		Status of the digital output

0	off
1	on

Name	r/w	Adr. Integer		real	Typ	Value/off		Description
F.Do1	r/w	base 1dP 2dP 3dP	4381 12573 20765 28957	41530	Enum	*Enum_Ausgang*		Forcing of this digital output. Forcing involves the external operation of an output. The instrument has no influence on this output (use of free outputs by superordinate system).

0	off
1	on

Name	r/w	Adr. Integer		real	Typ	Value/off		Description
F.Out1	r/w	base 1dP 2dP 3dP	4382 12574 20766 28958	41532	Float	0...120	☐	Forcing value of the analog output. Forcing involves the external operation of an output, i.e. the instrument has no influence on this output. (Used for the operation of free outputs e.g. by a supervisory PLC.)

16 ProG

PArA

Name	r/w	Adr. Integer		real	Typ	Value/off		Description
SP.01	r/w	base 1dP 2dP 3dP	6100 14292 22484 30676	44968	Float	-1999...9999	☑	End setpoint of segment 1. This is the target setpoint that is reached at the end of the first segment. The target setpoint is approached from the previous valid setpoint (when starting the 1st segment, matching to process value!). When the program is completed, the controller continues with the last target setpoint reached.
Pt.01	r/w	base 1dP 2dP 3dP	6101 14293 22485 30677	44970	Float	0...9999	☐	Segment time 1 defines the duration of the first segment. The gradient of this segment is calculated using the segment time and the setpoint difference (SP – segment starting setpoint).Note: The 1st segment is started at process value.
SP.02	r/w	base 1dP 2dP 3dP	6102 14294 22486 30678	44972	Float	-1999...9999	☑	End setpoint of segment 2. This is the target setpoint that is reached at the end of the second segment. The target setpoint is approached from the previous valid setpoint. When the program is completed, the controller continues with the last target setpoint reached.
Pt.02	r/w	base 1dP 2dP 3dP	6103 14295 22487 30679	44974	Float	0...9999	☐	Segment time 2 defines the duration of the second segment. The gradient of this segment is calculated using the segment time and the setpoint difference (SP – segment starting setpoint).Note: The 1st segment is started at process value.
SP.03	r/w	base 1dP 2dP 3dP	6104 14296 22488 30680	44976	Float	-1999...9999	☑	End setpoint of segment 3. This is the target setpoint that is reached at the end of the third segment. The target setpoint is approached from the previous valid setpoint. When the program is completed, the controller continues with the last target setpoint reached.
Pt.03	r/w	base 1dP 2dP 3dP	6105 14297 22489 30681	44978	Float	0...9999	☐	Segment time 3 defines the duration of the third segment. The gradient of this segment is calculated using the segment time and the setpoint difference (SP – segment starting setpoint).Note: The 1st segment is started at process value.
SP.04	r/w	base 1dP 2dP 3dP	6106 14298 22490 30682	44980	Float	-1999...9999	☑	End setpoint of segment 4. This is the target setpoint that is reached at the end of the fourth segment. The target setpoint is approached from the previous valid setpoint. When the program is completed, the controller continues with the last target setpoint reached.
Pt.04	r/w	base 1dP 2dP 3dP	6107 14299 22491 30683	44982	Float	0...9999	☐	Segment time 4 defines the duration of the fourth segment. The gradient of this segment is calculated using the segment time and the setpoint difference (SP – segment starting setpoint).Note: The 1st segment is started at process value.

16 ProG

Signal

Name	r/w	Adr. Integer		real	Typ	Value/off		Description
St.Prog	r	base 1dP 2dP 3dP	6170 14362 22554 30746	45108	Int	0...255	☐	The programmer's status contains bit-wise coded data, e.g. which point of the program sequence the program has reached.
								Bit 0,1,2 Type of segment 0: rising 1: falling 2: hold (dwell) Bit 3 Program 'Run' Bit 4 Program 'End' Bit 5 Program 'Reset' Bit 6 Program 'StartFlankMissing' Bit 7 Program 'BandHold + FailHold' Bit 8 Program active
SP.Pr	r	base 1dP 2dP 3dP	6171 14363 22555 30747	45110	Float	-1999...9999	☐	The programmer's setpoint is displayed as the effective setpoint while the program is running.
T1.Pr	r	base 1dP 2dP 3dP	6172 14364 22556 30748	45112	Float	0...9999	☐	Only with a running program. The net (elapsed) time of the programmer is shown in a simplified form as time elapsed since program start.Caution: Stop times are not counted! If the first segment is defined as a gradient, the program starts at the process value, whereby the offset is defined as the time that the controller would have needed with the gradient beginning at the setpoint valid at program start.
T3.Pr	r	base 1dP 2dP 3dP	6173 14365 22557 30749	45114	Float	0...9999	☐	Only with running program. The remaining programmer time is given by the sum of the currently running segment plus the times of the remaining program segments (without hold times).
T2.Pr	r	base 1dP 2dP 3dP	6174 14366 22558 30750	45116	Float	0...9999	☐	Only while program is running. The net segment time corresponds to the elapsed segment time.Caution: Stop times are not counted! If the first segment has been defined as a gradient, the start commences at process value, and the offset specified for the first segment corresponds to the time that the controller would have required with a gradient beginning at the actual process value when the program was started.
T4.Pr	r	base 1dP 2dP 3dP	6175 14367 22559 30751	45118	Float	0...9999	☐	Only with running program. The remaining time of the running program segment (without hold times).
SG.Pr	r	base 1dP 2dP 3dP	6176 14368 22560 30752	45120	Int	0...4	☐	A program consists of one or more segments which are arranged and defined by means of the segment numbers. By means of the segment number(s), the program can be changed quickly and specifically at the required point.

17 SEtP

PArA

Name	r/w	Adr. Integer		real	Typ	Value/off		Description
SP.LO	r/w	base 1dP 2dP 3dP	3100 11292 19484 27676	38968	Float	-1999...9999	☐	Lower setpoint limit. The setpoint is raised to this value automatically, if a lower setpoint is adjusted. BUT: The (safety) setpoint W2 is not restricted by the setpoint limits! The setpoint reserve for the step function is 10% of SPHi - SPLo.
SP.Hi	r/w	base 1dP 2dP 3dP	3101 11293 19485 27677	38970	Float	-1999...9999	☐	Upper setpoint limit. The setpoint is reduced to this value automatically, if a higher setpoint is adjusted. BUT: The (safety) setpoint W2 is not restricted by the setpoint limits! The setpoint reserve for the step function is 10% of SPHi - SPLo.
SP.2	r/w	base 1dP 2dP 3dP	3102 11294 19486 27678	38972	Float	-1999...9999	☐	Second (safety) setpoint. Ramp function as with other setpoints (effective, external). However, SP2 is not restricted by the setpoint limits.
r.SP	r/w	base 1dP 2dP 3dP	3103 11295 19487 27679	38974	Float	0,01...9999	☑	Setpoint gradient [/min] or ramp. Max. rate of change in order to avoid step changes of the setpoint. The gradient acts in the positive and negative directions. Note for self-tuning: with activated gradient function, the setpoint gradient is started from the process value, so that there is no sufficient setpoint reserve.
t.SP	r/w	base 1dP 2dP 3dP	3104 11296 19488 27680	38976	Float	0...9999	☐	The timer (preset) value is entered in minutes with one decimal digit (0,1 minute = 6 seconds). With an activated timer, the preset value is displayed automatically in the extended Operating Level, where it can be changed by means of the parameter t.ti.

Signal

Name	r/w	Adr. Integer		real	Typ	Value/off		Description
SP.EF	r	base 1dP 2dP 3dP	3170 11362 19554 27746	39108	Float	-1999...9999	☐	Effective setpoint. The value reached at the end of setpoint processing, after taking W2, external setpoint, gradient, boost function, programmer settings, start-up function, and limit functions into account. Comparison with the effective process value leads to the control deviation, from which the necessary controller response is derived.
SP	r/w	base 1dP 2dP 3dP	3180 11372 19564 27756	39128	Float	-1999...9999	☐	Setpoint for the interface (without the additional function 'Controller off'). SetpInterface acts on the internal setpoint before the setpoint processing stage. Note: The value in RAM is always updated. To protect the EEPROM, storage of the value in the EEPROM is timed (at least one value per half hour).
SP.d	r/w	base 1dP 2dP 3dP	3181 11373 19565 27757	39130	Float	-1999...9999	☐	The effective setpoint is shifted by this value. In this way, the setpoints of several controllers can be shifted together, regardless of the individually adjusted effective setpoints.
t.ti	r/w	base 1dP 2dP 3dP	3182 11374 19566 27758	39132	Float	0...9999	☐	Current timer count in minutes. Count-down timer. The run time is only visible, if the timer is active. Configuration in the extended Operating Level.

18 Tool

ConF

Name	r/w	Adr. Integer		real	Typ	Value/off	Description
U.LinT	r/w	base	634	34036	Enum	*Enum_Unit*	Engineering unit of linearization table (temperature).
		1dP	8826				
		2dP	17018				
		3dP	25210				

0	without unit
1	°C
2	°F

9499-040-63511

Subject to alterations without notice
Änderungen vorbehalten
Sous réserve de toutes modifications

P.O.B. 310 229, D-34058 Kassel, Germany
Printed in Germany 9499-040-63511 (02/2005)

A4, unibind, SW-Druck, 80g² weiß