

# Installation & Operation Manual

# 6060 MODBUS PROTOCOL

This manual is a supplement to the 6060  
Full Installation & Operation Manual PK514



PK539  
0037-75565  
July 2016

**Explanation of Symbols:**



**General Information**



**Caution: ESD-Sensitive Components**



**General Warning**

---

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

# Table of Contents

<b>1. General .....</b>	<b>5</b>
1.1 References .....	5
<b>2. Commissioning the interface .....</b>	<b>6</b>
2.1 Mounting hints.....	6
2.2 Electrical connections .....	6
2.2.1 RS 485 version (two-wire ).....	6
2.2.2 RS 422 version (four-wire - RS 485) .....	6
2.2.3 Cable installation .....	8
2.2.4 Screening.....	8
2.2.5 Terminating resistors .....	8
2.2.6 Installation notes .....	10
2.3 Bus settings.....	10
2.3.1 Bus address.....	10
2.3.2 Transmission parameters .....	11
2.4 Master operation (MASt).....	12
2.5 System layout.....	13
2.5.1 Minimum configuration of a MODBUS installation.....	13
2.5.2 Maximum configuration of a MODBUS installation.....	13
2.5.3 Wiring inside buildings.....	14
<b>3. Bus Protocol .....</b>	<b>15</b>
3.1 Composition of a transmission byte.....	15
3.2 General message frame .....	15
3.2.1 CRC .....	16
3.2.2 End of frame detection .....	16
3.3 Transmission principles .....	16
3.4 Response delay (dELY) .....	16
3.5 Modem operation (C.dEL) .....	16
3.6 Function codes.....	17
3.6.1 Reading several values .....	17
3.6.2 Writing a single value .....	18
3.7 Writing several values .....	19
3.8 Error record .....	20
3.8.1 Error codes.....	20
3.9 Diagnosis.....	21
3.9.1 Return transmission of the received message (0x00) .....	22
3.9.2 Restart of communication (terminates the Listen Only mode) (0x01) .....	22
3.9.3 Return transmission of the diagnosis register (0x02) .....	22
3.9.4 Change to the Listen Only mode (0x04) .....	22
3.9.5 Delete the counter and reset the diagnosis register (0x0A) .....	22
3.9.6 Return transmission of the message counter (0x0B) .....	22
3.9.7 Return transmission of the counter for faulty message transmissions .....	23
3.9.8 Return transmission of the counter for messages with error code .....	23
3.9.9 Return transmission of the message counter for this slave .....	23
3.9.10 Return transmission of the counter for unanswered messages.....	23

3.9.11 Return transmission of the counter for messages answered with NAK.....	23
3.9.12 Return transmission of the counter for messages answered with Busy.....	24
3.9.13 Return transmission of the parity error counter .....	24
3.9.14 Return transmission of the framing error counter .....	24
3.9.15 Return transmission of the counter for too long messages .....	24
<b>4. MODBUS addresses, address areas, and address formats.....</b>	<b>25</b>
4.1 Area definitions.....	25
4.2 Special values.....	25
4.3 Composition of the address tables .....	25
4.4 Internal data types .....	26
<b>5. Index .....</b>	<b>27</b>
<b>6. Address tables.....</b>	<b>28</b>

# 1. General

This document describes the implementation and operation of the MODBUS interface used with the industrial controller 6060 which will be called 'device' in the rest of this document.

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 CommPort® (PC) interface. This interface is not bussable, and serves for a direct connection with the ChromaloxPro® 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-Protocol:

### [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

## 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 and 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, 1/4 watt).

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

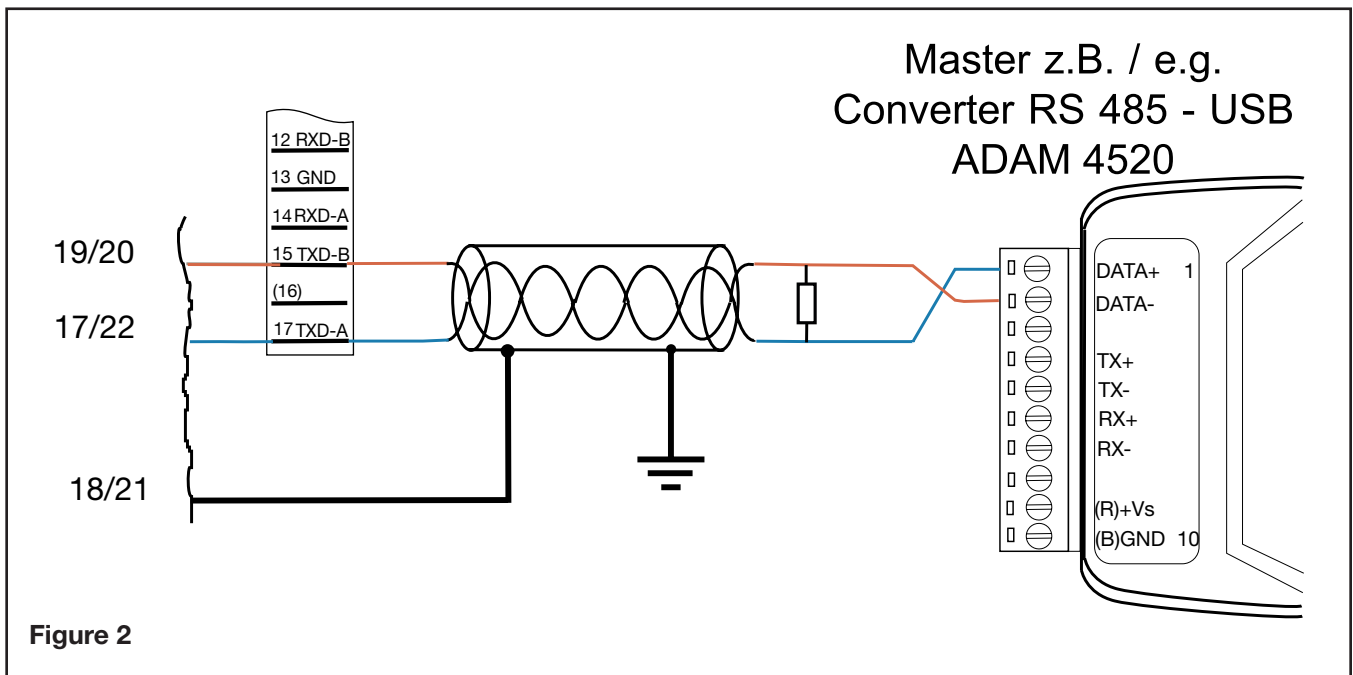
Definition MODBUS	According to Unit
D1	Data A
D0	Data B
Common	RGND

**Notes:**

1. Terminating resistors between Data A and B at the cable ends (see figure 2 on page 8)
2. Screening (see figure 2 on page 8)
3. GND lead (see Fig. 6)

Signal	Terminal
<b>Option 3 Position</b>	
TXD-B	19
TXD-A	17
GDN	18
<b>Option A Position</b>	
TXD-B	20
TXD-A	22
GDN	21

The following cable connection methods are possible.



### 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

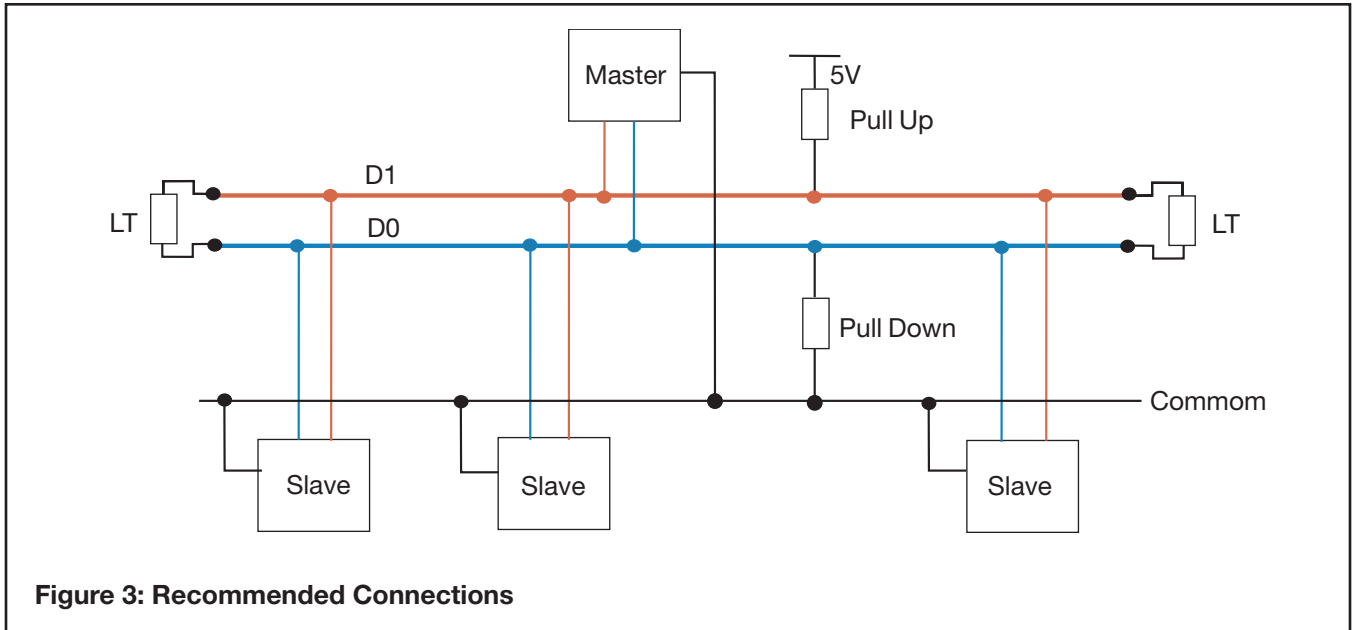
The widespread US Standard EIA RS 485 recommends fitting terminating resistors at each end of the bus cable. Terminating resistors usually have a value of approx. 120 ohms, and are connected in parallel between the data lines A and B (depending on the cable impedance; for details, see the cable manufacturer's 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 is 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  $\pm 200\text{mV}$  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].



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:

Baud Rate	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.

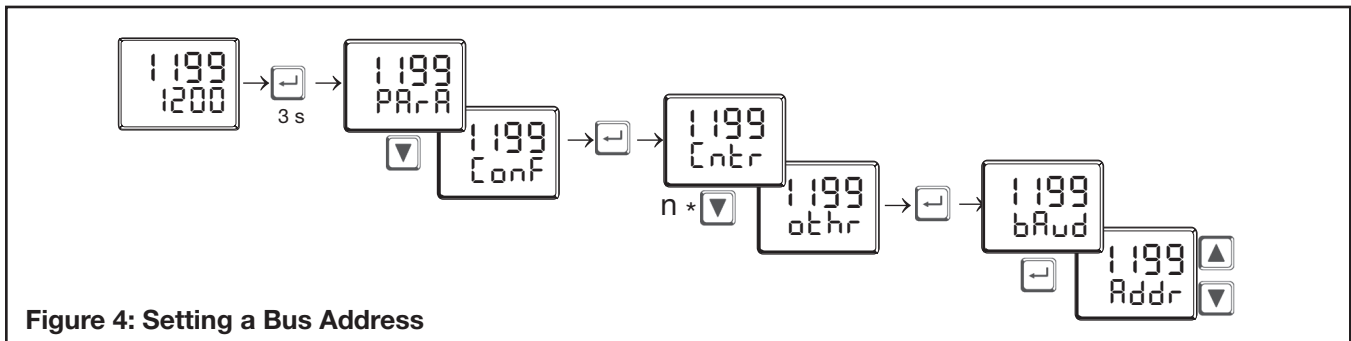
---

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



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 behavior 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 Master Operation (MASt)

The 6060 master function is limited to broadcast messages (data transmission to all connected slaves).

For operation as a master, the instrument must be configured accordingly by means of ChromaloxPro® (engineering software for 6060).

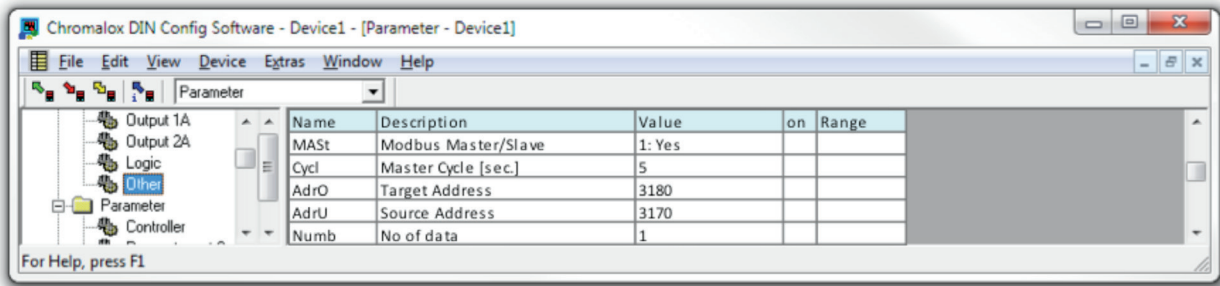
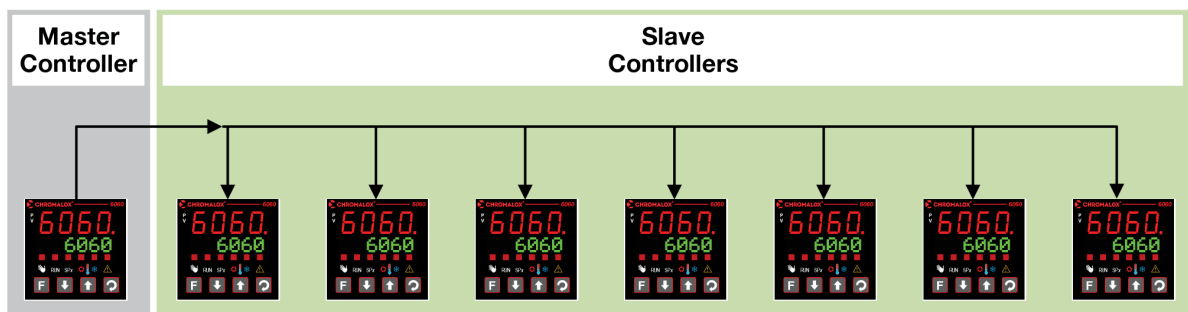


Figure 5: Master Function Parameter Setting

A possible MODBUS master configuration is given in the drawing shown above. In this example, the actual master set-point (source address 3170) is transmitted to the slaves (target address 3180) at intervals of 5 seconds.



Controller	Master	Slave 1	Slave 2	Slave 3	Slave 4	Slave 5	Slave 6	Slave 7
Set Point	180°C	185°C	190°C	195°C	200°C	195°C	200°C	190°C
Master Change	+5°C	↓	↓	↓	↓	↓	↓	↓
New Set Point	185°C	190°C	195°C	200°C	205°C	200°C	205°C	195°C

Figure 6: Example

## 2.5 System Layout



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

### 2.5.1 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).

### 2.5.2 Maximum configuration of a MODBUS installation

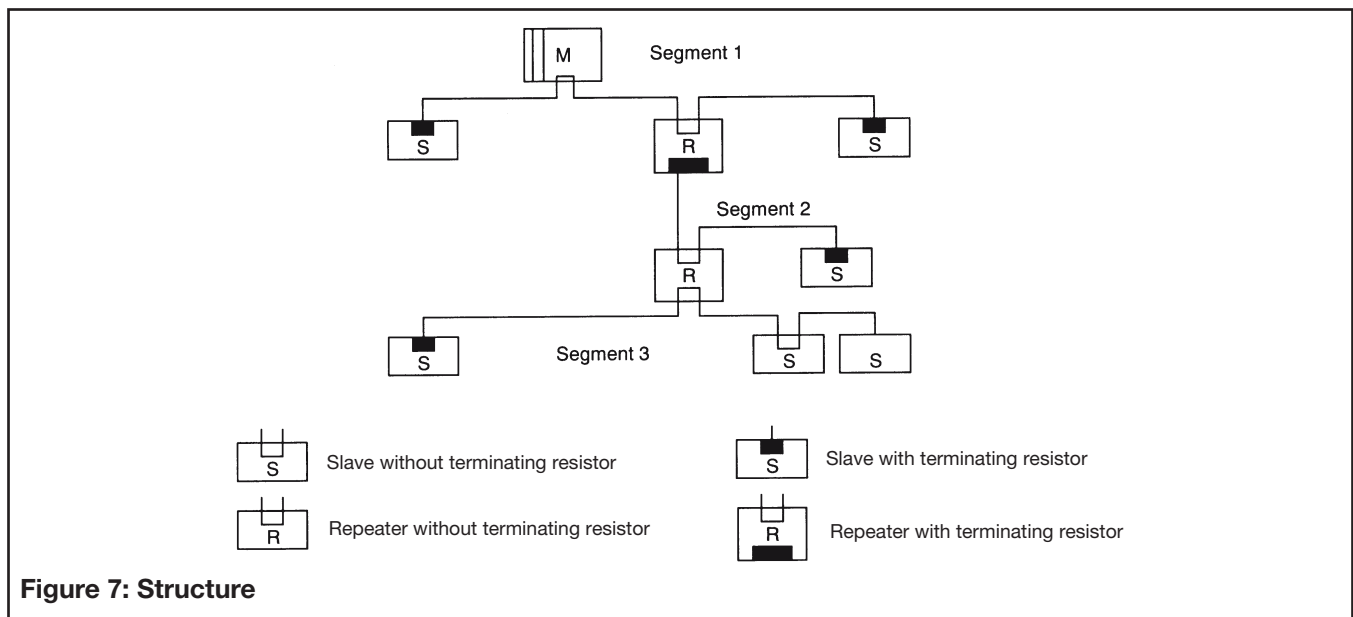
A bus segment consists of max. 32 field units (active and passive). The greatest number of slave participants that can be operated by one MODBUS master via several segments, is determined by the internal memory structure of the master. Therefore, you should know the specifications of the master when 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.



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.

### 2.5.3 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.

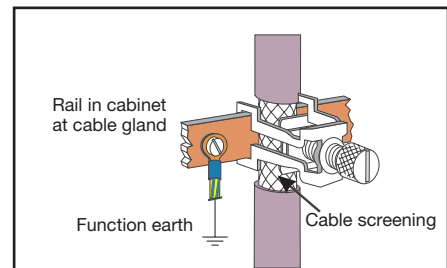
The screen must be taken right up to the field unit, where it is to be connected to the conductive housing and/or the metal connector. Hereby, it must be ensured that the device housing (and possibly the control cabinet in which the device is installed), are held at equal ground potential by means of 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<sup>2</sup> 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.



**Figure 8: Screen Connection**

# 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:

<b>Start bit</b>	<b>8 data bits</b>	<b>Parity/Stop bit</b>	<b>Stop bit</b>
------------------	--------------------	------------------------	-----------------

## 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:

<b>Device address</b>	<b>Function code</b>	<b>Data field</b>	<b>CRC</b>	<b>End of frame detection</b>
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		Description	Explanation
Hex	Dez		
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 behavior 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	02	Starting address 650
Start address Low	8A	Reading the MODBUS diagnostic register
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 or 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 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 message 'Writing several parameters or configuration data'. Composition: 80hex + 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

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 with Error Code

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 message answered with error code

### 3.9.9 Return Transmission of the Message Counter for this Slave

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

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

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

### 3.9.12 Return Transmission of the Counter for Messages Answered with Busy

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

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

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

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 & 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		Data Transfer Format	Smallest Transferable Value	Largest Transferable Value	Resolution
hex	dez.				
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/                         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

## A

address areas.....	27 - 28
address formats.....	27 - 28
address tables.....	31 - 32
addressing.....	5
area definitions.....	27

## B

Baudrate.....	13
broadcast.....	17
broadcast – mode.....	18
bus address.....	13
bus protocol.....	17 - 26
bus segment.....	15
busprotocol.....	17 - 26

## C

cable installation.....	11
cable screen.....	16
commissioning the interface.....	6 - 14
composition of the address tables.....	28
CRC.....	17 - 18

## D

data.....	17
device address.....	17
diagnosis.....	23 - 26

## E

electrical connections.....	6 - 12
end of frame detection.....	18
error codes.....	22
error record.....	22

## F

format	
float.....	27
float acc. to ieee.....	27
integer.....	27
Motorola.....	27
four-wire.....	9
function code.....	17, 19 - 20
function codes.....	19 - 20

## G

general message frame.....	17
----------------------------	----

## I

installation notes.....	13
internal data types.....	28

## L

lead length.....	5
------------------	---

## M

max. length.....	14
maximum configuration.....	15
modbus addresses.....	27 - 28
modem operation.....	18
mounting.....	6

## P

parity error.....	14
-------------------	----

## R

reading values.....	19
references.....	5
repeater.....	15
response delay.....	18
RS 422.....	9
RS 485.....	7

## S

Screening.....	11
special values.....	27
sensor fault.....	27
undefined value.....	27
stop bit.....	13

## T

terminating resistors.....	11
transmission byte.....	17
transmission format.....	27
two-wire.....	7

## U

unicast – mode.....	18
---------------------	----

## W

wiring.....	16
writing a value.....	20
writing several values.....	21

# 6. Address Tables

The following sections describe the address tables for:

- **industrial controller 6060**

# Code Table

## Table of Contents

<b>1 Cntr</b>		<b>12 ohnE3</b>	
Con .....	1	Signal .....	34
PAr .....	4	<b>13 ohnE4</b>	
Signal .....	7	Signal .....	34
<b>2 InP.1</b>		<b>14 othr</b>	
Con .....	13	Con .....	34
Par.....	14	Signal .....	37
Signal .....	14	<b>15 Out.1</b>	
<b>3 InP.2</b>		Con .....	42
Con .....	15	Signal .....	45
Par.....	16	<b>16 Out.2</b>	
Signal .....	16	Con .....	46
<b>4 InP.3</b>		Signal .....	49
Con .....	17	<b>17 Out.3</b>	
Par.....	18	Con .....	49
Signal .....	18	Signal .....	53
<b>5 Lim</b>		<b>18 Out.4</b>	
Con .....	19	Con .....	54
Par.....	20	Signal .....	56
Signal .....	21	<b>19 Out.5</b>	
<b>6 Lim2</b>		Con .....	57
Con .....	22	Signal .....	60
Par.....	22	<b>20 Out.6</b>	
<b>7 Lim3</b>		Con .....	61
Con .....	23	Signal .....	64
Signal .....	23	<b>21 PAr.2</b>	
Par.....	24	Par.....	64
Signal .....	24	<b>22 ProG</b>	
<b>8 LOGI</b>		Con .....	65
Con .....	24	Par.....	66
Signal .....	29	Signal .....	99
<b>9 ohnE</b>		<b>23 SETP</b>	
Par.....	31	Par.....	100
Signal .....	31	Signal .....	101
<b>10 ohnE1</b>		<b>24 Tool</b>	
Signal .....	33	Con .....	102
<b>11 ohnE2</b>			
Signal .....	34		

## Code Table

### 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	Setpoint controller can be switched over to external setpoint (->LOGI/SPE)
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 ( $x = SP \pm b.t$ ). 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 ( $x = SP \pm b.t$ ). 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 switch-over. 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 switch-over. 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 (= fixed positioning value). When the timer (t.SP) has elapsed, the controller switches over to the adjusted setpoint.
7	Timer, operating mode 6 (setpoint switch-over). 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 ( $x = SP \pm b.t$ ). When the timer has elapsed, the controller switches back to setpoint SP, and the lower display alternates between 'End' and the setpoint.
10	Setpoint controller with start-up function. The start-up function is a protective function, e.g. with hot runner control. To prevent destruction of high-performance heating elements, they must be heated slowly to remove any humidity. With activated start-up function, the controller maintains the reduced starting temperature for a defined dwell period. Subsequently, the controller switches over to the main setpoint.
11	Setpoint controllers are switchable to external setpoint and to a second setpoint, always with the start-up function. The start-up function is a protective function, e.g. with hot runner control. To prevent destruction of high-performance heating elements, they must be heated slowly to remove any humidity. With activated start-up function, the controller maintains the reduced starting temperature for a defined dwell period. Subsequently, the controller switches over to the main setpoint.

# 1 Cntr

- ConF

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 <input type="checkbox"/>	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 switch over). 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 behavior (algorithm) referred to output value: e.g. 2- or 3-point controller, signaller, 3-point stepping control.
							0 On/Off (2-point) controller or signaller with one output. The on/off controller or the signaller switches its output when the process value leaves the defined hysteresis band around the setpoint.
							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.
							7 3-point signaller. The 3-point signaller switches two digital outputs, depending on their switching difference (Sd1 and Sd2), the trigger point separation, and from the control deviation.
							8 3-point stepping controller that can be switched over to signaller operation. Via interface or front panel key (depending on configuration), the operating mode can be switched between 3-point stepping controller and signaller (1 output).
							9 3-point stepping controller can be switched over to 3-point-signaller operation. Via interface or front panel key (depending on configuration), the operating mode can be switched between 3-point stepping controller and 3-point-signaller (2 outputs).
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).

# 1 Cntr

ConF							
Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
C.Act	r/w	Base	5052	42872	Enum	Enum_CAct	Operating sense of the controller. Inverse operation (e.g. heating) means increased actuator value when the process value falls. Direct operation (e.g. cooling) means increased actuator value when the process value increases.
		1dP	13244				
		2dP	21436				
		3dP	29628				
						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.
FAIL	r/w	Base	5053	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.
		1dP	13243				
		2dP	21435				
		3dP	29627				
						0	Controller outputs disabled.
						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.
rnG.L	r/w	Base	5059	42886	Float	-1999...9999 <input type="checkbox"/>	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.
		1dP	13251				
		2dP	21443				
		3dP	29635				
rnG.H	r/w	Base	5060	42888	Float	-1999...9999 <input type="checkbox"/>	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.
		1dP	13252				
		2dP	21444				
		3dP	29636				
SP2C	r/w	Base	5054	42876	Enum	Enum_S2PC	When switching over to the 2nd setpoint SP.2, control is performed without cooling.
		1dP	13246				
		2dP	21438				
		3dP	29630				
						0	Standard (cooling permitted with all setpoints).
SP2C	r/w	Base	5055	42878	Enum	Enum_CYCL	Duty cycle for 2-point and 3-point controllers. Internally, the controller calculates a continuous output value, which is converted into switching pulses for digital outputs. The user can adapt the setting to calculate various duty cycles (on/off ratio).
		1dP	13247				
		2dP	21439				
		3dP	29631				
						0	Standard. 'Bathtub curve'. The adjusted duty cycles t1 and t2 are valid for ± 50% control output. With very small and very large control outputs, the effective duty cycle is increased sufficiently to prevent nonsensically short operating pulses. The shortest pulses are limited to 1/4 of t1 and 1/4 of t2.

# 1 Cntr

- ConF

Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
tunE	r/w	Base 1dP 2dP 3dP	5056 13248 21440 29632	42880	Enum	Enum_tune	Self-tuning procedure / sequence. Choice between: step response tuning during start-up and pulse response tuning at setpoint; or pulse response tuning during start-up and at setpoint; or only step response tuning during start-up, and no tuning at setpoint (no pulse).
							<p>0 At start-up with step function, impulse function at setpoint. The step function at start up requires a control deviation of more than 10% of the control range. At setpoint, with control deviation less than 10% of the control range, tuning is done with the impulse function.</p> <p>1 At start-up with impulse function. Setting for fast controlled systems (e.g. hot runner control). Always tuning with impulse function. At start up, with a control deviation of more than 10% of the control range, the control loop is optimized for a wide control range. At setpoint the control deviation during self-tuning is small.</p> <p>2 At start up and at setpoint always tune step function at start up. Tuning is done with step function at start up, regardless of the control deviation.</p>
Strt	r/w	Base 1dP 2dP 3dP	5057 13249 21441 29633	42882	Enum	Enum_Strt	Start of self-tuning. Self-tuning can always be started manually at the request of the operator. Here, it is possible to determine that self-tuning is started automatically under the following conditions: On power-up or when an oscillation of the process value is detected.
							<p>0 No automatic start (manual start via front interface)</p> <p>1 Manual or automatic start of auto-tuning at power on or when oscillating is detected (oscillating of process value by more than <math>\pm 0.5\%</math> of the control range, and simultaneously the output value by more than 20%.) Note: Though the process is unchanged, at power on always the (time-consuming) auto-tuning is started.</p>
Adt.0	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.
							<p>0 The duty cycle is determined by the self-tuning procedure. This ensures the best control results</p> <p>1 The cycle duration is not determined 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).</p>

- 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 <input type="checkbox"/>	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).

# 1 Cntr

## • PArA

Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
Pb2	r/w	Base 1dP 2dP 3dP	5001 13193 21385 29577	42770	Float	1...9999 <input type="checkbox"/>	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	42772	Float	1...9999 <input checked="" type="checkbox"/>	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 <input checked="" type="checkbox"/>	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 <input checked="" type="checkbox"/>	Derivative action time 1 (heating) [s]. 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 <input checked="" type="checkbox"/>	Derivative action time 2 (cooling) [s]. 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 <input type="checkbox"/>	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 <input type="checkbox"/>	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	0.4...9999 <input type="checkbox"/>	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.
D.SP	r/w	Base 1dP 2dP 3dP	5016	42800 13208 21400 29592	Float	-1999...9999 <input type="checkbox"/>	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	42800	Float	-1999...9999 <input checked="" type="checkbox"/>	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 <input type="checkbox"/>	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 <input type="checkbox"/>	Lower output limit [%] The range is dependent of the type of controller: 2 point controller: 0...ymax+1 3 point controller: -105 ymax-1

PArA							
Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
Y.Hi	r/w	Base 1dP 2dP 3dP	5019 13211 21403 29595	42806	Float	-105...105 <input type="checkbox"/>	Upper output limit [%] The range is ymin+1 ....105
Y2	r/w	Base 1dP 2dP 3dP	5017 13209 21401 29593	42802	Float	-100...100 <input type="checkbox"/>	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 <input type="checkbox"/>	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 <input type="checkbox"/>	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 <input type="checkbox"/>	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.
E.H2O	r/w	Base 1dP 2dP 3dP	5013 13205 21397 29589	42794	Float	-1999...9999 <input type="checkbox"/>	Min. temperature for water cooling. Below the set temperature no water cooling happens.
t.on	r/w	Base 1dP 2dP 3dP	5010 13202 21394 29586	42788	Float	0.1...9999 <input type="checkbox"/>	Impulse length for water cooling. Fixed for all values of controller output. The pause time is varied.
t.off	r/w	Base 1dP 2dP 3dP	5011 13203 21395 29587	42790	Float	1...9999 <input type="checkbox"/>	Min. pause time for water cooling. The max. effective controller output results from t.on/(t.on+t.off)·100%
F.H2O	r/w	Base 1dP 2dP 3dP	5012 13204 21396 29588	42792	Float	0.1...9999 <input type="checkbox"/>	Adaptation of the (non-linear) water-cooling characteristic. If the cooling action is very strong, and causes an unfavorable transition between heating and cooling, a non-linear characteristic can reduce the cooling action considerably. Adjust FH20 = 1 for output values up to -70%; FH20 = 2 for values up to approx. -80%, and FH20 = 0.5 for up to approx. -60%.
HYS.L	r/w	Base 1dP 2dP 3dP	5028 13220 21412 29604	42824	Float	0...9999 <input type="checkbox"/>	Switching hysteresis below the setpoint of the signaller [engineering unit].
HYS.H	r/w	Base 1dP 2dP 3dP	5029	42826	Float	0...9999 <input type="checkbox"/>	Switching hysteresis above the setpoint of the signaller [engineering unit].
Tu2	r	Base 1dP 2dP 3dP	5145 13337 21529 29721	43058	Float	0...9999 <input type="checkbox"/>	'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 <input type="checkbox"/>	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 <input type="checkbox"/>	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.

**Signal**

Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
St.Cntr	r	Base 1dP 2dP 3dP	5100 13292 21484 29676	42968	Int	0...65535 <input type="checkbox"/>	Status informations of the controller.f.e. switching signals, controller off or informations about selftuning. The controller status 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: Control signal: Manual/automatic 0: automatic 1: manual Bit 4: Control signal: Y2 0: Y2 not activ 1: Y2 activ Bit 5: Control signal: Ext. setting of output signal 0: not activ 1: activ Bit 6: Control signal: 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 active 1: active Bit 10: Rate to setpoint 0: not active 1: active 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 output signal 1 0 0 1 Outputs switched off (neutral) 1 0 1 0 Abortion of the selftuning (by control- or error-signal)
diFF	r	Base 1dP 2dP 3dP	5104 13296 21488 29680	42976	Float	-1999...9999 <input type="checkbox"/>	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 <input type="checkbox"/>	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 <input type="checkbox"/>	'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 <input type="checkbox"/>	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.

Signal							
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 <input type="checkbox"/>	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 <input type="checkbox"/>	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
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
SPEF	r	Base 1dP 2dP 3dP	5101 13293 21485 29677	42970	Float	-1999...9999 <input type="checkbox"/>	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.?	r	Base 1dP 2dP 3dP	5102 13294 21486 29678	42972	Float	-1999...9999 <input type="checkbox"/>	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 <input type="checkbox"/>	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)
Vmax1	r	Base 1dP 2dP 3dP	5142 13334 21526 29718	43052	Float	0...9999 <input type="checkbox"/>	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	4143 13335 21527 29719	43054	Float	0...9999 <input type="checkbox"/>	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	43604	Enum	Enum_Msg	The result of self-tuning for 'cooling' indicates whether self-tuning was successful, and with what result.
							0 no message/tuning is active
							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 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 Self-tuning was successful, but with a warning. The new parameters are valid. Note: The process value turning point of the step response is too low. Quality of control is limited. 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 Self-tuning was stopped before the output step change was made, because the setpoint reserve is too small (the controller waits). Confirming this error message aborts the self-tuning, and the controller continues operation in the automatic mode. Possible remedy: Reduce the setpoint adjustment range, change the setpoint, or reduce the process value.

# 1 Cntr

- Signal

Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
Msg1	r	Base	5144	43056	Enum	Enum_Msg	The result of self-tuning for 'heating' indicates whether self-tuning was successful, and with what result.
		1dP	13336				
		2dP	21528				
		3dP	29720				
						0	no message/tuning is active
						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	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	Self-tuning was successful, but with a warning. The new parameters are valid. Note: The process value turning point of the step response is too low. Quality of control is limited. 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	Self-tuning was stopped before the output step change was made, because the setpoint reserve is too small (the controller waits). Confirming this error message aborts the self-tuning, and the controller continues operation in the automatic mode. Possible remedy: Reduce the setpoint adjustment range, change the setpoint, or reduce the process value.

## 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	The result of self-tuning for 'cooling' indicates whether self-tuning was successful, and with what result.
							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-Pt 10% Fahrenheit: 32...3200°F 5 Thermocouple type R (0...1760°C), PtRh-Pt13% Fahrenheit: 32...3200°F 6 Thermocouple type T (-200...400°C), Cu-CuNi Fahrenheit: -328...752°F 7 Thermocouple type C (0...2315°C), W5%Re-W26%Re Fahrenheit: 32...4199°F 8 Thermocouple type D (0...2315°C), W3%Re-W25%Re Fahrenheit: 32...4199°F 9 Thermocouple type E (-100...1000°C), NiCr-CuNi Fahrenheit: -148...1832°F 10 Thermocouple type B (0/100...1820°C), PtRh-Pt6% Fahrenheit: 32/212 ... 3308°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 ... 850°C) Fahrenheit: -328...1562 °F 22 Special : 0...4500 Ohms. 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 30 current : 0/4...20 mA 40 40 voltage : 0...10 V
S.Lin	r/w	Base 1dP 2dP 3dP	1151 9343 17535 25727	35070	Enum	Enum_SLin	Special linearization (not adjustable for all sensor types S.tYP). The linearization table can be created with the Engineering Tool.
							0 No special linearization 1 Special linearization. Definition of the linearization table is possible with the Engineering Tool.

## 2 InP.1

- ConF

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) is 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 PAR-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.

- PARA

Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
InL.?	r/w	Base 1dP 2dP 3dP	1100 9292 17484 25676	34968	Float	-1999...9999 <input type="checkbox"/>	Input value of the lower scaling point. The display of the value is done using the corresponding measured electrical value
OuL.?	r/w	Base 1dP 2dP 3dP	1101 9293 17485 25677	34970	Float	-1999...9999 <input type="checkbox"/>	Display value of the lower scaling point. This is the physical value, which is assigned to the measured lower input value.
InH.?	r/w	Base 1dP 2dP 3dP	1102 9294 17486 25678	34972	Float	-1999...9999 <input type="checkbox"/>	Input value of the upper scaling point. The display of the value is done using the corresponding measured electrical value.
OuH.?	r/w	Base 1dP 2dP 3dP	1103 9295 17487 25679	34974	Float	-1999...9999 <input type="checkbox"/>	Display value of the upper scaling point. This is the physical value, which is assigned to the measured upper input value.
t.F?	r/w	Base 1dP 2dP 3dP	1104 9296 17488 25680	34976	Float	0...100 <input type="checkbox"/>	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.?r	r	Base 1dP 2dP 3dP	1170 9362 17554 25746	35108	Float	-1999...9999 <input type="checkbox"/>	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.
In.?	r	Base 1dP 2dP 3dP	1172 9364 17556 25748	35112	Float	-1999...9999 <input type="checkbox"/>	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 <input type="checkbox"/>	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 SPE or (depending on version) external setpoint shift SPE. (Switch-over is done via -> LOGI/SPE).
						5	Preset for external positioning value Y.E (switch-over via -> LOGI/Y.E)
S.tYP	r/w	Base 1dP 2dP 3dP	1250 9442 17634 25826	35268	Enum	Enum_StYP2_1	Sensor type selection
						30	Current : 0...20 mA / 4...20 mA
						31	0...30 mA current (AC)
Corr	r/w	Base 1dP 2dP 3dP	162 8354 16546 24738	33092	Enum	Enum_Corr	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) is 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.

PARA							
Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
InL.?	r/w	Base 1dP 2dP 3dP	1200 9392 17584 25776	35168	Float	-1999...9999 <input type="checkbox"/>	Input value of the lower scaling point. The display of the value is done using the corresponding measured electrical value.
OuL.?	r/w	Base 1dP 2dP 3dP	1201 9393 17585 25777	35170	Float	-1999...9999 <input type="checkbox"/>	Display value of the lower scaling point. This is the physical value, which is assigned to the measured lower input value.
InH.?	r/w	Base 1dP 2dP 3dP	1202 9394 17586 25778	35172	Float	-1999...9999 <input type="checkbox"/>	Input value of the upper scaling point. The display of the value is done using the corresponding measured electrical value.
OuH.?	r/w	Base 1dP 2dP 3dP	1203 9395 17587 25779	35175	Float	-1999...9999 <input type="checkbox"/>	Display value of the upper scaling point. This is the physical value, which is assigned to the measured upper input value.

Signal							
Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
In.?	r	Base 1dP 2dP 3dP	1270 9462 17654 25846	35308 9462 17654 25846	Float	-1999...9999 <input type="checkbox"/>	Measurement value after the measurement value correction (e.g. with offset or 2-point correction and scaling).
Fail	r	Base 1dP 2dP 3dP	1271 9463 17655 25847	35310	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
In.?	r	Base 1dP 2dP 3dP	1272 9464 17656 25848	35312	Float	-1999...9999 <input type="checkbox"/>	Measurement value before the measurement value correction (unprocessed).
F.Inp	r/w	Base 1dP 2dP 3dP	1280 9472 17664 25856	35328	Float	-1999...9999 <input type="checkbox"/>	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.)

## 4 InP.3

ConF							
Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
I.Fnc	r/w	Base 1dP 2dP 3dP	166 8358 16550 24742	33100	Enum	Enum_IFnc	Selection of the function assigned to the value at INP3, e.g. value at INP3 is the external setpoint.
						0	No function (subsequent input data are skipped)
						1	Heating current input.
						2	External setpoint SPE or (depending on version) external setpoint shift SPE. (Switch-over is done via -> LOGI/SPE).
						5	Preset for external positioning value Y.E (switch-over via -> LOGI/Y.E)

ConF							
Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
S.tYP	r/w	Base 1dP 2dP 3dP	1350 9542 17734 25926	35468	Enum	Enum_StYP2_3	Sensor type selection. For sensors with signals of resistance transducer, current or voltage measuring, scaling can be adjusted.
S.Lin	r/w	Base 1dP 2dP 3dP	1351 9543 17735 25927	35470	Enum	Enum_SLin	Special linearization (not adjustable for all sensor types S.tYP). The linearization table can be created with the Engineering Tool.
						0	No special linearization.
						1	Special linearization. Definition of the linearization table is possible with the Engineering Tool.
Corr	r/w	Base 1dP 2dP 3dP	165 8357 16549 24741	33098	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) is 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 PAR-level). The input values for the upper (InL, OuL) and lower scaling point (InH, OuH) are visible at the parameter level.

PARA							
Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
InL.?	r/w	Base 1dP 2dP 3dP	1300 9492 17684 25876	35368	Float	-1999...9999 <input type="checkbox"/>	Input value of the lower scaling point. The display of the value is done using the corresponding measured electrical value.
OuL.?	r/w	Base 1dP 2dP 3dP	1301 9493 17685 25877	35370	Float	-1999...9999 <input type="checkbox"/>	Display value of the lower scaling point. This is the physical value, which is assigned to the measured lower input value.
InH.?	r/w	Base 1dP 2dP 3dP	1302 9494 17686 25878	35372	Float	-1999...9999 <input type="checkbox"/>	Input value of the upper scaling point. The display of the value is done using the corresponding measured electrical value.
OuH.?	r/w	Base 1dP 2dP 3dP	1303 9495 17687 25879	35374	Float	-1999...9999 <input type="checkbox"/>	Display value of the upper scaling point. This is the physical value, which is assigned to the measured upper input value.
t.F?	r/w	Base 1dP 2dP 3dP	1304 9496 17688 25880	35376	Float	0...999.9 <input type="checkbox"/>	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.?	r	Base 1dP 2dP 3dP	1370 9562 17754 25946	35508	Float	-1999...9999 <input type="checkbox"/>	Measurement value after the measurement value correction (e.g. with offset or 2-point correction, and scaling).
Fail	r	Base 1dP 2dP 3dP	1371 9563 17755 25947	35510	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
In.?r	r	Base 1dP 2dP 3dP	1372 9564 17756 25948	35512	Float	-1999...9999 <input type="checkbox"/>	Measurement value before the measurement value correction (unprocessed).
F.Inp	r/w	Base 1dP 2dP 3dP	1380 9572 17764 25956	35528	Float	-1999...9999 <input type="checkbox"/>	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.)

## 5 Lim

ConF							
Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
Fnc.?	r/w	Base 1dP 2dP 3dP	2150 10342 18534 26726	37068	Enum	Enum_Fnc	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 reset.
Src.?	r/w	Base 1dP 2dP 3dP	2151 10343 18535 26727	37070	Enum	Enum_Src	Source for limit value. Selection of which value is to be monitored.
						0	process value = absolute alarm
						1	Control deviation $x_w$ (process value - setpoint) = relative alarm Note: Monitoring with the effective setpoint $Weff$ . For example using a ramp it is the changing setpoint, not the target setpoint of the ramp.
						2	Control deviation $X_w$ (= 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 $10 * Ti1$ .
						6	effective setpoint $Weff$ . For example the ramp-function changes the effective setpoint until it matches the internal (target) setpoint.
						7	correcting variable $y$ (controller output)
						11	Control deviation $X_w$ (= 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.

HC.AL	r/w	Base 1dP 2dP 3dP	2050 10242 18434 26626	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. Sho circuit is monitored in both cases.
							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.
LP.AL	r/w	Base 1dP 2dP 3dP	5058 13250 21442 29634	42884	Enum	Enum_LPAL	Monitoring of control loop interruption (not possible with 3-point stepping controller, not possible with signaller)
							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.

### 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 <input type="checkbox"/>	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 <input type="checkbox"/>	Measured heating current [A]. Apart from the short circular 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 <input type="checkbox"/>	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.

## 6 Lim2

- ConF

Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
Fnc.?	r/w	Base 1dP 2dP 3dP	2250 10442 18634 26826	37268	Enum	Enum_Fnc	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 reset.
							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 reset.
Src.?	r/w	Base 1dP 2dP 3dP	2251 10443 18635 26827	37270	Enum	Enum_Src	Source for limit value. Selection of which value is to be monitored.
							0 process value = absolute alarm
							1 Control deviation $x_w$ (process value - setpoint) = relative alarm Note: Monitoring with the effective setpoint $Weff$ . For example using a ramp it is the changing setpoint, not the target setpoint of the ramp.
							2 Control deviation $X_w$ (= 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 $10 * Ti1$ .
							6 effective setpoint $Weff$ . For example the ramp-function changes the effective setpoint until it matches the internal (target) setpoint.
							7 correcting variable $y$ (controller output)
							11 Control deviation $X_w$ (= 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.?	r/w	Base 1dP 2dP 3dP	2200 10392 18584 26776	37618	Float	-1999...9999 <input checked="" type="checkbox"/>	Lower limit value. The alarm is triggered if the value falls below the limit, and is reset with lower limit value plus hysteresis.
H.?	r/w	Base 1dP 2dP 3dP	2201 10393 18585 26777	37170	Float	-1999...9999 <input checked="" type="checkbox"/>	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.?	r/w	Base 1dP 2dP 3dP	2202 10394 18586 26778	37172	Float	0...9999 <input type="checkbox"/>	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.
dEL.?	r/w	Base 1dP 2dP 3dP	2203 10395 18587 26779	37174	Float	0...9999 <input type="checkbox"/>	Delayed alarm of a limit value. The alarm is only triggered after the defined delay time. It is only indicated, and possibly stored, if it is still present after the delay time has elapsed.

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	Limit value status: No alarm present or stored.

## 7 Lim3

ConF							
Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
Fnc.?	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 reset.
						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 reset.
Src.?	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 $x_w$ (process value - setpoint) = relative alarm Note: Monitoring with the effective setpoint $Weff$ . For example using a ramp it is the changing setpoint, not the target setpoint of the ramp.
						2	Control deviation $X_w$ (= 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 $10 * Ti1$ .
						6	effective setpoint $Weff$ . For example the ramp-function changes the effective setpoint until it matches the internal (target) setpoint.
						7	correcting variable $y$ (controller output)
						11	Control deviation $X_w$ (= 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.?	r/w	Base 1dP 2dP 3dP	2200 10492 18684 26876	37368	Float	-1999...9999 <input checked="" type="checkbox"/>	Lower limit value. The alarm is triggered if the value falls below the limit, and is reset with lower limit value plus hysteresis.
H.?	r/w	Base 1dP 2dP 3dP	2301 10493 18685 26877	37370	Float	-1999...9999 <input checked="" type="checkbox"/>	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.?	r/w	Base 1dP 2dP 3dP	2302 10494 18686 26878	37372	Float	0...9999 <input type="checkbox"/>	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.
dEL.?	r/w	Base 1dP 2dP 3dP	2203 10495 18687 26879	37374	Float	0...9999 <input type="checkbox"/>	Delayed alarm of a limit value. The alarm is only triggered after the defined delay time. It is only indicated, and possibly stored, if it is still present after the delay time has elapsed.

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	Limit value status: No alarm present or stored.

## 8 LOGI

ConF							
Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
L_r	r/w	Base 1dP 2dP 3dP	1051 9243 17435 25627	34780	Enum	Enum_dlnP_ Ks2	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 on
						2	Digital Input di1 switches
						3	Process value = absolute alarm
						4	Digital Input di1 switches
						5	di3 switches (only visible with OPTION)
						6	F-key switches
						7	Limit 1 switches
						8	Limit 2 switches
						9	Limit 3 schaltet

## 8 LOGI

- ConF

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_dlnP_ Ks20	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)
						1	Always on
						2	Digital Input di1 switches
						3	Digital Input di1 switches
						4	di3 switches (only visible with OPTION)
						5	di4 switches (only visible with OPTION)
						6	F-key switches
						7	Limit 1 switches
						8	Limit 2 switches
						9	Limit 3 schaltet
SPE	r/w	Base 1dP 2dP 3dP	1053 9245 17437 25628	34874	Enum	Enum_dlnP_ Ks20	Switching between internal set-point an external setpoint S.P.E. The external S.P.E is either the absolute set-point $W_{ext}$ or the offset to the set-point (dependent on instrument and configuration).
						0	No function (switch-over via interface is possible)
						1	Always on
						2	Digital Input di1 switches
						3	Digital Input di1 switches
						4	di3 switches (only visible with OPTION)
						5	di4 switches (only visible with OPTION)
						6	F-key switches
						7	Limit 1 switches
						8	Limit 2 switches
						9	Limit 3 schaltet
Y2	r/w	Base 1dP 2dP 3dP	1054 9246 17438 25630	34876	Enum	Enum_dlnP_ Ks20	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)
						1	Always on
						2	Digital Input di1 switches
						3	Digital Input di1 switches
						4	di3 switches (only visible with OPTION)
						5	di4 switches (only visible with OPTION)
						6	F-key switches
						7	Limit 1 switches
						8	Limit 2 switches
						9	Limit 3 schaltet

**ConF**

Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
YE	r/w	Base 1dP 2dP 3dP	1055 9247 17439 25631	34878	Enum	Enum_dlnP_ Ks20	Signal for activating the external output value. The internal output value Ypid is the controllers reaction on the process, with external output value Y.E the controller output is controlled.
						0	No function (switch-over via interface is possible)
						1	Always on
						2	Digital Input di1 switches
						3	Digital Input di1 switches
						4	di3 switches (only visible with OPTION)
						5	di4 switches (only visible with OPTION)
						6	F-key switches
						7	Limit 1 switches
						8	Limit 2 switches
						9	Limit 3 schaltet
mAn	r/w	Base 1dP 2dP 3dP	1056 9248 17440 25632	34880	Enum	Enum_dlnP_ Ks20	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 on
						2	Digital Input di1 switches
						3	Digital Input di1 switches
						4	di3 switches (only visible with OPTION)
						5	di4 switches (only visible with OPTION)
						6	F-key switches
						7	Limit 1 switches
						8	Limit 2 switches
						9	Limit 3 schaltet
C.oFF	r/w	Base 1dP 2dP 3dP	1057 9249 17441 25633	34882	Enum	Enum_dlnP_ Ks20	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)
						1	Always on
						2	Digital Input di1 switches
						3	Digital Input di1 switches
						4	di3 switches (only visible with OPTION)
						5	di4 switches (only visible with OPTION)
						6	F-key switches
						7	Limit 1 switches
						8	Limit 2 switches
						9	Limit 3 schaltet

**ConF**

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_dlnP_ Ks20	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)
						1	Always on
						2	Digital Input di1 switches
						3	Digital Input di1 switches
						4	di3 switches (only visible with OPTION)
						5	di4 switches (only visible with OPTION)
						6	F-key switches
						7	Limit 1 switches
						8	Limit 2 switches
						9	Limit 3 schaltet
Err.r	r/w	Base 1dP 2dP 3dP	1059 9521 17443 25635	34886	Enum	Enum_dlnP_ Ks20	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).
						0	No function (switch-over via interface is possible)
						1	Always on
						2	Digital Input di1 switches
						3	Digital Input di1 switches
						4	di3 switches (only visible with OPTION)
						5	di4 switches (only visible with OPTION)
						6	F-key switches
						7	Limit 1 switches
						8	Limit 2 switches
						9	Limit 3 schaltet
booS	r/w	Base 1dP 2dP 3dP	1060 9252 17444 25636	34888	Enum	Enum_dlnP_ Ks20	Source of the control signal for activating the boost function: The setpoint is increased by the value SP.bo for the duration t.bo. The boost function causes a brief setpoint increase, which is used to clear blocked channels from 'frozen' material in a hot runner system.
						0	No function (switch-over via interface is possible)
						1	Always on
						2	Digital Input di1 switches
						3	Digital Input di1 switches
						4	di3 switches (only visible with OPTION)
						5	di4 switches (only visible with OPTION)
						6	F-key switches
						7	Limit 1 switches
						8	Limit 2 switches
						9	Limit 3 schaltet

**ConF**

Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description	
Pid.2	r/w	Base 1dP 2dP 3dP	1061 9253 17445 25637	34890	Enum	Enum_dlnP_ Ks20	Source of the control signal for switchover between the two parameter sets. The second parameter set is complete, and comprises Pb (= proportional band), ti (= integral action time), and td (= derivative action time) for heating and for cooling. All other control parameters, e.g. the switching duty cycles, are valid for both parameter sets.	
							0	No function (switch-over via interface is possible)
							1	Always on
							2	Digital Input di1 switches
							3	Digital Input di1 switches
							4	di3 switches (only visible with OPTION)
							5	di4 switches (only visible with OPTION)
							6	F-key switches
							7	Limit 1 switches
							8	Limit 2 switches
P.run	r/w	Base 1dP 2dP 3dP	1062 9254 17446 25638	34880	Enum	Enum_dlnP_ Ks20	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.	
							0	No function (switch-over via interface is possible)
							1	Always on
							2	Digital Input di1 switches
							3	Digital Input di1 switches
							4	di3 switches (only visible with OPTION)
							5	di4 switches (only visible with OPTION)
							6	F-key switches
							7	Limit 1 switches
							8	Limit 2 switches
P.oFF	r/w	Base 1dP 2dP 3dP	1063 9255 17447 25639	34894	Enum	Enum_dlnP_ Ks20	Source of the control signal for switching off the programmer (if the programmer is switched off, the internal setpoint becomes effective).	
							0	No function (switch-over via interface is possible)
							1	Always on
							2	Digital Input di1 switches
							3	Digital Input di1 switches
							4	di3 switches (only visible with OPTION)
							5	di4 switches (only visible with OPTION)
							6	F-key switches
							7	Limit 1 switches
							8	Limit 2 switches
9	Limit 3 schaltet							

**ConF**

Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
di.Fn	r/w	Base 1dP 2dP 3dP	1050 9242 17434 25626	34868	Enum	Enum_diFn	Function of digital inputs (valid for all inputs)
							<p>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.</p> <p>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.</p> <p>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'.</p>

**Signal**

Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
St.Di	r	Base 1dP 2dP 3dP	1070 9262 17454 25646	34908	Int	0...7 <input checked="" type="checkbox"/>	Status of the digital inputs or of push-buttons (binary coded).
							<p>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</p>
L-R	r/w	Base 1dP 2dP 3dP	1080 9272 17464 25656	34928	Int	0...1 <input type="checkbox"/>	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 <input type="checkbox"/>	Signal for activating the second (safety) setpoint (SP2= 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 <input type="checkbox"/>	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 <input type="checkbox"/>	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!
Y_Y.E	r/w	Base 1dP 2dP 3dP	1084 9276 17468 25660	34936	Int	0...1 <input type="checkbox"/>	Signal for activating the external positioning value. The controller is operated as positioner.
A-M	r/w	Base 1dP 2dP 3dP	1085 9277 17469 25661	34938	Int	0...1 <input type="checkbox"/>	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 <input type="checkbox"/>	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 <input type="checkbox"/>	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).

Signal							
Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
Err.r	r/w	Base 1dP 2dP 3dP	1088 9280 17472 25663	34944	Int	0...1 <input type="checkbox"/>	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 25664	34946	Int.	0...1 <input type="checkbox"/>	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.
Boost	r/w	Base 1dP 2dP 3dP	1090 9282 17475 25666	34948	Int	0...1 <input type="checkbox"/>	Signal for activating the boost function. The boost function causes a brief setpoint increase, which is used e.g. to clear blocked channels ('frozen' material) in a hot-runner system.
Set1.2	r/w	Base 1dP 2dP 3dP	1091 9283 17475 25667	34950	Int	0...1 <input type="checkbox"/>	Switch-over of parameter set. The 2nd parameter set contains one complete set each of Pb (= proportional band), ti (= integral action time), and td (= derivative action time) for heating and for cooling. All other control parameters, such as switching duty cycles, are valid for both parameter sets.
Prg.R.S	r/w	Base 1dP 2dP 3dP	1092 9284 17476 25668	34952	Int	0...1 <input type="checkbox"/>	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.
Prg.Res	r/w	Base 1dP 2dP 3dP	1093 9285 17477 25669	34954	Int	0...1 <input type="checkbox"/>	Programmer reset switches the programmer off, and sets it back to the starting condition. Reset stops the currently active program, and activates the internal setpoint. A newly selected program becomes the active program.
F.Di	r/w	Base 1dP 2dP 3dP	1094 9286 17478 25670	34956	Int	0...7 <input type="checkbox"/>	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.)
						Bit 0 Bit 1 Bit 2 Bit 3 Bit 4	Forcing of digital input 1 Forcing of digital input 2 Forcing of digital input 3 Forcing of digital input 4 Forcing of digital input 5

## 9 ohnE

PARA							
Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
Conf	r/w	Base 1dP 2dP 3dP	8193 16385 24577	32770	Int	0...2 <input type="checkbox"/>	Start/Stop and abortion of the configuration mode 0 = End of configuration 1 = Start of configuration 2 = Abort configuration

Signal							
Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
UPD	r/w	Base 1dP 2dP 3dP	95 8287 16479 24671	32958	Enum	Enum_Aenderungslag	Status message indicating that parameter / configuration have been changed via the front panel.
						0	No change via the front panel keys.
						1	A change has been made via the front panel keys, which must be processed.

Signal							
Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
Hw.Opt	r	Base 1dP 2dP 3dP	200 8392 16584 24778	33168	Int	0...65535 <input type="checkbox"/>	KSx-1-devices hardware option 0000 WXYZ 0000 00BA Z=1: Option Modbus + di2/di3 + TPS Y=1: Option INP3 (KS90-1, KS90-1P) X=1: Option 16 programs (KS90-1P) W=1: Option OUT5/OUT6 (KS50-1, KS90-1, KS90-1P) A=1: OUT3 is analogue output B=1: OUT4 is analogue output (KS90-1, KS90-1P)
Sw.Op	r	Base 1dP 2dP 3dP	201 8393 16585 24777	33170	Int	0...255 <input type="checkbox"/>	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.
Bed.V	r	Base 1dP 2dP 3dP	202 8394 16586 24778	33172	Int	0...255 <input type="checkbox"/>	Operating version (numeric value). For the correct interaction of E-Tool and device, the software version and operating version must match.
Unit	r	Base 1dP 2dP 3dP	203 8395 16587 24779	33173	Int	0...255 <input type="checkbox"/>	Identification of the device.
S.Vers	r	Base 1dP 2dP 3dP	204 8396 16588 24780	33176	Int	100...255 <input type="checkbox"/>	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	... <input type="checkbox"/>	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 <input type="checkbox"/>	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	33720	Int	0...31 <input type="checkbox"/>	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

Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
St.Ain	r	Base 1dP 2dP 3dP	252 8444 16636 24828	33272	Int	0...7 <input type="checkbox"/>	Bit-coded status of the analog input (fault, e.g. short circuit)
						Bit 0 Bit 1 Bit 2 Bit 3 Bit 4 Bit 5 Bit 6 Bit 7 Bit 8 Bit 9 Bit 10 Bit 11	Break at Input 1 Reversed polarity at Input 1 Short circuit at Input 1 Not used Break at Input 2 Reversed polarity at Input 2 Short-circuit at Input 2 Not used Break at Input 3 (only KS 90) Reversed polarity at Input 3 (only KS 90) Short-circuit at Input 3 (only KS 90) Not used
St.Di	r	Base 1dP 2dP 3dP	253 8444 16637 24829	33274	Int	0...7 <input type="checkbox"/>	Status of the digital inputs or of push-buttons (binary coded).
						Bit 0 Bit 1 Bit 2 Bit 8 Bit 9 Bit 10 Bit 11 Bit 12 Bit 13	Input 1 Input 2 Input 3 Status of 'F' key Status of 'A/M' key Status of 'Sel' key Status of 'Down' key Status of 'Up' key Status of 'Loc' key
F.Di	r/w	Base 1dP 2dP 3dP	303 8495 16687 24879	33374	Int	0...1 <input type="checkbox"/>	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.)
						Bit 0 Bit 1 Bit 2 Bit 3 Bit 4	Forcing of digital input 1 Forcing of digital input 2 Forcing of digital input 3 Forcing of digital input 4 Forcing of digital input 5
F.Do	r/w	Base 1dP 2dP 3dP	304 8496 16688 24880	33376	Int	0...15 <input type="checkbox"/>	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).
Cmodules	r	Base 1dP 2dP 3dP	911 9103 17295 25487	34590	Text	... <input type="checkbox"/>	In the device connected modules. If an error E4 occur this data should be compared with the code number of the Uident.

## 10 ohnE1

- Signal

Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
In.?	r	Base 1dP 2dP 3dP	232 8424 16616 24808	33232	Float	-1999...9999 <input type="checkbox"/>	Measurement value after the measurement value correction (e.g. with offset or 2-point correction, and scaling).
In.?r	r	Base 1dP 2dP 3dP	240 8432 16624 24816	33248	Float	-1999...9999 <input type="checkbox"/>	Measurement value before the measurement value correction (unprocessed).
F.Inp	r/w	Base 1dP 2dP 3dP	300 8492 16684 24876	33368	Float	-1999...9999 <input type="checkbox"/>	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.)
In.?	r	Base 1dP 2dP 3dP	233 8425 16617 24809	33234	Float	-1999...9999 <input type="checkbox"/>	Measurement value after the measurement value correction (e.g. with offset or 2-point correction, and scaling).

- Signal**

Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
In.?r	r	Base 1dP 2dP 3dP	241 8433 16625 24817	33250	Float	-1999...9999 <input type="checkbox"/>	Measurement value before the measurement value correction (unprocessed).
F.Inp	r/w	Base 1dP 2dP 3dP	301 8493 16685 24877	33370	Float	-1999...9999 <input type="checkbox"/>	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 ohnE2

- Signal**

Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
In.?	r	Base 1dP 2dP 3dP	233 8425 16617 24809	33234	Float	-1999...9999 <input type="checkbox"/>	Measurement value after the measurement value correction (e.g. with offset or 2-point correction, and scaling).
In.?r	r	Base 1dP 2dP 3dP	241 8433 16625 24817	33250	Float	-1999...9999 <input type="checkbox"/>	Measurement value before the measurement value correction (unprocessed).
F.Inp	r/w	Base 1dP 2dP 3dP	301 8493 16685 24877	33370	Float	-1999...9999 <input type="checkbox"/>	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.)

## 12 ohnE3

- Signal**

Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
F.Out.?	r/w	Base 1dP 2dP 3dP	305 8497 16689 24481	33378	Float	0...120 <input type="checkbox"/>	Forcing value of the analog output. Forcing involves the external operation of an output, i.e. the instrument has influence on this output. (Used for the operation of free outputs e.g. by a supervisory PLC.)

## 13 ohnE4

- Signal**

Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
F.Out.?	r/w	Base 1dP 2dP 3dP	306 8498 16690 24482	33380	Float	0...120 <input type="checkbox"/>	Forcing value of the analog output. Forcing involves the external operation of an output, i.e. the instrument has influence on this output. (Used for the operation of free outputs e.g. by a supervisory PLC.)

## 14 othr

- ConF

Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
D2.Err	r/w	Base	193	33154	Enum	Enum_Disp2E	Queued faults can be displayed directly in the 2nd line of the display. In case of a fault, the display then alternates between the value of the lower display line (standard = setpoint) and the error message for the fault with the highest priority (blinking display).
		1dP	8385				
		2dP	16577				
		3dP	24769				
						0	Display line 2 is not switched over in case of a fault. The fault is signalled via the LED, and the error message is shown in the error list.
						1	In case of a fault, display line 2 alternates between the error message and the value of the lower display line- The fault with the highest priority is displayed as long as it is present. Latched (stored) faults must be acknowledged in order to remove them from the display.
F.Coff	r/w	Base	192	33152	Enum	Enum_Coff	The standard disabling procedure only switches off the controller outputs, whereby the alarms, displays, and other functions remain active. Alternatively, all functions can be switched off (including alarms and displays).
		1dP	8384				
		2dP	16576				
		3dP	24768				
						0	Only the PID controller functions are disabled. The analog controller outputs have the value 0.0, and the switching outputs generate the logical state FALSE. All other functions, e.g. alarms and displays, continue operating in the normal manner.
						1	All the controller functions are disabled. The analog outputs have the value 0.0, and the switching outputs generate the logical state FALSE. If configured, an inversion is carried out.
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
Addr	r/w	Base	181	33130	Int	1...247 <input type="checkbox"/>	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, 2 stop bits
						1	Even parity
						2	Odd parity
						3	no parity with 1 stopbit
dELY	r/w	Base	183	33134	Int	0...200 <input type="checkbox"/>	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

**ConF**

Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description	
dP	r/w	Base 1dP 2dP 3dP	171 8363 16555 24747	33110	Enum	Enum_dP	Decimal point (max. no of decimals). Format of the measured value display.	
							0	No digit behind the decimal point
							1	One digit behind the decimal point
							2	Two digits behind the decimal point
							3	Three digits behind the decimal point
LEd	r/w	Base 1dP 2dP 3dP	190 8382 16574 24766	33148	Enum	Enum_Led	Meaning of the signalling LEDs. Selection of a combination of the displayable signals.	
							0	The digital outputs OUT1, OUT2, and OUT3 are displayed.
							1	Display of controller output y1 (heating / open), alarm2, and alarm3
							2	Display of controller output y1 (heating / open), controller output y2 (cooling / close), alarm3
							3	Display of controller output y2 (cooling / close), controller output y1 (heating / open), alarm3
C.dEL	r/w	Base 1dP 2dP 3dP	184 8376 16568 24760	33136	Int	0...200 <input type="checkbox"/>	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 continuously by the modem.	
FrEq	r/w	Base 1dP 2dP 3dP	150 8342 16534 24726	33068	Enum	Enum_FrEq	Switchover of the applied mains frequency 50 / 60 Hz, thereby better adaptation of the input filter for hum suppression.	
							0	mains frequency 50 Hz
							1	mains frequency 60 Hz
MASt	r/w	Base 1dP 2dP 3dP	185 8377 16569 24761	33138	Enum	Enum_MASt	Device works as Modbus master. The communication is executed according to the master/slave principle, whereby the device can be operated as master or as slave. Operation as master must be configured here.	
							0	The unit is operated as a Modbus slave.
							1	The unit is operated as a Modbus master.
Cycl	r/w	Base 1dP 2dP 3dP	186 8378 16570 24762	33140	Int	0...200 <input type="checkbox"/>	Cycle time (in seconds) during which the Modbus master transmits its message on the bus.	
AdrO	r/w	Base 1dP 2dP 3dP	187 8379 16571 24763	33142	Int	1...65535 <input type="checkbox"/>	Target address to which the data specified with AdrU are output on the bus.	
AdrU	r/w	Base 1dP 2dP 3dP	188 8380 16572 24764	33144	Int	1...65535 <input type="checkbox"/>	Modbus address of the data output on the bus by the Modbus master.	
NUMB	r/w	Base 1dP 2dP 3dP	189 8381 1673 24765	33146	Int	0...100 <input type="checkbox"/>	Quantity of data that are to be transmitted from the Modbus master.	

**Signal**

Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
E.1	r/w	Base	210	33146	Enum	Defect	Err 1 (internal error) Contact Service.
		1dP	8402				
		2dP	16594				
		3dP	24765				
							0 No fault exists
							2 The device is defective
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
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.
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
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.
FbF.2	r/w	Base	215	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!)
		1dP	8407				
		2dP	16599				
		3dP	24791				
							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.

**Signal**

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.
HCA	r/w	Base 1dP 2dP 3dP	218 8410 16602 24794	33204	Enum	HeatCurr	Heating current alarm. possible faults are an open heating current circuit with current $I <$ heating current lim 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.
							2 A short-circuit fault has occurred.
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 sensor break alarm possible (Reset).
							1 A short-circuit fault has occurred and has been stored.
							2 A short-circuit fault has occurred.
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 short-circuit 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 setpoint range, change setpoint, 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.
AdA.C	r/w	Base 1dP 2dP 3dP	222 8414 16606 24798	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!)
							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 setpoint range, change setpoint, 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.

**Signal**

Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
Lim.1	r/w	Base 1dP 2dP 3dP	223 8415 16607 24799	33214	Enum	Limit	Limit value 1 exceeded. Hint for trouble-shooting: check the process. (As a process value via fieldbus interface not writable!)
							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.
Lim.2	r/w	Base 1dP 2dP 3dP	224 8416 16608 24800	33216	Enum	Limit	Limit value 2 exceeded. Hint for trouble-shooting: check the process. (As a process value via fieldbus interface not writable!)
							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.
Lim.3	r/w	Base 1dP 2dP 3dP	225 8417 16609 24801	33218	Enum	Limit	Limit value 3 exceeded. Hint for trouble-shooting: check the process. (As a process value via fieldbus interface not writable!)
							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.
Inf.1	r/w	Base 1dP 2dP 3dP	226 8418 16610 24802	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!)
							0 No signal, resetting of the time limit signal possible (Reset).
							1 Operating hours - limit value (maintenance period) reached: please acknowledge.
Inf.2	r/w	Base 1dP 2dP 3dP	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!)
							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.
E.4	r/w	Base 1dP 2dP 3dP	228 8430 16612 24804	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!)
							0 No fault, resetting possible (Reset).
							1 A fault has occurred and has been stored.

## 15 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 de-energized mode
						1	Inverse / normally closed
O.tYp	r/w	Base	4170	41108	Enum	Enum_OtYp	Signal type selection OUT
		1dP	12362				
		2dP	20554				
		3dP	28746				
						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
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.
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	This 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	This 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	This 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	This loop alarm (= open loop alarm) is assigned to this output.

**ConF**

Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
HC.AL	r/w	Base	4158	41804	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_HSCS	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	10543				
		3dP	28735				
							0 Not active
							1 This output is activated by an SSR fault.
timE	r/w	Base	4160	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.
		1dP	12352				
		2dP	20544				
		3dP	28736				
							0 Not active
							1 This output is activated by the timer status
t.End	r/w	Base	4176	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).
		1dP	12368				
		2dP	20560				
		3dP	28752				
							0 Not active
							1 This output is activated by the message 'Timer end'.
P.End	r/w	Base	4161	41120	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	12353				
		2dP	20545				
		3dP	28737				
							0 Not active.
							1 This output is activated by the message 'Program end'.
FAi.1	r/w	Base	4162	41092	Enum	Enum_FAI1	Output function: Signal INP1 fault. The fail signal is generated, if a fault occurs at the analog Input INP1.
		1dP	12354				
		2dP	20546				
		3dP	28738				
							0 Not active.
							1 This output sends the error message 'INP1 fault'.
FAi.2	r/w	Base	4163	41094	Enum	Enum_FAI2	Output function: Signal INP2 fault. The fail signal is generated, if a fault occurs at the analog Input INP2.
		1dP	12355				
		2dP	20546				
		3dP	28738				
							0 Not active.
							1 This output sends the error message 'INP2 fault'.
PrG.1	r/w	Base	4165	41098	Enum	Enum_Prg1	Output function: Signal programmer's control output no. 1. A control output is one of the four digital signals that can be operated segment-wise by a program.
		1dP	12357				
		2dP	20549				
		3dP	28741				
							0 Not active.
							1 Control output 1 is assigned to this output.

**ConF**

Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
PrG.2	r/w	Base	4166	41100	Enum	Enum_PrG2	Output function: Signal programmer's control output no 2. A control output is one of the four digital signals that can be operated segment-wise by a program.
		1dP	12358				
		2dP	20550				
		3dP	28742				
							0 Not active
							1 Control output 2 is assigned to this output.
PrG.3	r/w	Base	4167	41102	Enum	Enum_PrG3	Output function: Signal programmer's control output no. 3. T A control output is one of the four digital signals that can be operated segment-wise by a program.
		1dP	12359				
		2dP	20551				
		3dP	28743				
							0 Not active
							1 Control output 3 is assigned to this output.
PrG.4	r/w	Base	4168	41104	Enum	Enum_PrG4	Output function: Signal programmer's control output no 4. A control output is one of the four digital signals that can be operated segment-wise by a program.
		1dP	12360				
		2dP	20552				
		3dP	28744				
							0 Not active
							1 Control output 4 is assigned to this output.
CALL	r/w	Base	4169	41106	Enum	Enum_CALL	Output: Operator call. At the end of a program segment, a contact is set, e.g. for an acoustic signal. This indicates to the operator that a certain program status has been reached, and operator action is required. Operator calling is used, if the program may only be continued after a check or some kind of operator action.
		1dP	12361				
		2dP	20553				
		3dP	28745				
							0 Not active
							1 This output is switched by an operator call.
Out.0	r/w	Base	4171	41110	Float	-1999...9999 <input type="checkbox"/>	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).
		1dP	12363				
		2dP	20555				
		3dP	28747				
Out.1	r/w	Base	4172	41112	Float	-1999...9999 <input type="checkbox"/>	Upper scaling limit of the analog output (corresponds to 100%). If current or voltage signals are used as output values, scaling of the display can be applied to the output value by means of the Parameter Level. Definition of the upper output limit is done using the corresponding electrical value (mA / V).
		1dP	12364				
		2dP	20556				
		3dP	28748				
O.Src	r/w	Base	4173	41114	Enum	Enum_Osrc	Signal source of the analog output.
		1dP	12365				
		2dP	20557				
		3dP	28749				
							0 Not active
							1 Controller output Y1 (cont.)
							2 Controller output Y2 (cont.)
							3 Process value
							4 Effective setpoint Weff
							5 Control deviation xw (process value - setpoint)= relative alarm Note: Monitoring with the effective setpoint Weff. For example using a ramp it is the changing setpoint, not the target setpoint of the ramp.

Signal							
Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
Out?	r	Base 1dP 2dP 3dP	4180 12372 20564 28756	41128	Enum	Enum_Ausgang	Status of the digital output
							0 Off
							1 On
F.Do?	r/w	Base 1dP 2dP 3dP	4181 12373 20565 28757	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).
							0 Off
							1 On
Out.1	r/w	Base 1dP 2dP 3dP	4182 12374 20566 28758	41132	Float	0..120 <input type="checkbox"/>	Forcing value of the analog output. Forcing involves the external operation of an output, i.e. the instrument has an influence on this output. (Used for the operation of free outputs e.g. by a supervisory PLC.)
							0 Off
							1 On

## 16 Out.2

ConF							
Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
O.Act	r/w	Base 1dP 2dP 3dP	4250 12442 20634 28826	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.
							0 Direct / Normally de-energized mode
							1 Inverse / normally closed
Y.1	r/w	Base 1dP 2dP 3dP	4251 12443 20635 28827	41270	Enum	Enum_Y1	Output function: Controller output Y1
							0 Not active
							1 This output provides the controller output Y1.
Y.2	r/w	Base 1dP 2dP 3dP	4252 12444 20636 28828	41272	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.
Lim.1	r/w	Base 1dP 2dP 3dP	4253 12445 20637 28829	41272	Enum	Enum_Lim1	Output function: Signal limit 1
							0 Not active.
							1 This output is activated by an alarm from limit value 1.
Lim.2	r/w	Base 1dP 2dP 3dP	4254 12446 20638 28830	41076	Enum	Enum_Lim2	Output function: Signal limit 2
							0 Not active.
							1 This 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	4255 12447 20639 28831	41278	Enum	Enum_Lim3	Output function: Signal limit 3
							0 Not active
							1 This output is activated by an alarm from limit value 3
LP.AL	r/w	Base 1dP 2dP 3dP	4257 12449 20641 28833	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.
							0 Not active.
							1 This loop alarm (= open loop alarm) is assigned to this output.
HP.AL	r/w	Base 1dP 2dP 3dP	4258 12450 20642 28834	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.
							0 Not active.
							1 The heating current alarm is assigned to this output
HC.SC	r/w	Base 1dP 2dP 3dP	4259 12451 20643 28835	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.
							0 Not active.
							1 This output is activated by an SSR fault.
timE	r/w	Base 1dP 2dP 3dP	4260 12452 20644 28836	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.
							0 Not active.
							1 This output is activated by the timer status
t.End	r/w	Base 1dP 2dP 3dP	4276 12468 20660 28852	41329	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	4261 12453 20645 28837	51290	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	4262 12454 20646 28838	51292	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 This output sends the error message 'INP1 fault'.
FAi.2	r/w	Base 1dP 2dP 3dP	4263 12455 20647 28839	51294	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 This output sends the error message 'INP2 fault'.

Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
PrG.1	r/w	Base 1dP 2dP 3dP	4265 12457 20649 28841	41298	Enum	Enum_PrG1	Output function: Signal programmer's control output no. 1. A control output is one of the four digital signals that can be operated segment-wise by a program.
							0 Not active.
							1 Control output 1 is assigned to this output.
PrG.2	r/w	Base 1dP 2dP 3dP	4266 12458 20650 28842	41300	Enum	Enum_PrG2	Output function: Signal programmer's control output no. 2. A control output is one of the four digital signals that can be operated segment-wise by a program.
							0 Not active.
							1 Control output 2 is assigned to this output
PrG.3	r/w	Base 1dP 2dP 3dP	4267 12459 20651 28844	41302	Enum	Enum_PrG3	Output function: Signal programmer's control output no. 3. T A control output is one of the four digital signals that can be operated segment-wise by a program.
							0 Not active.
							1 Control output 3 is assigned to this output.
PrG.4	r/w	Base 1dP 2dP 3dP	4268 12460 20652 28844	41304	Enum	Enum_PrG4	Output function: Signal programmer's control output no. t. A control output is one of the four digital signals that can be operated segment-wise by a program.
							0 Not active.
							1 Control output 4 is assigned to this output.
CALL	r/w	Base 1dP 2dP 3dP	4269 12461 20653 28845	41306	Enum	Enum_CALL	Output: Operator call. At the end of a program segment, a contact is set, e.g. for an acoustic signal. This indicates to the operator that a certain program status has been reached, and operator action is required. Operator calling is used, if the program may only be continued after a check or some kind of operator action.
							0 Not active.
							1 This output is switched by an operator call.

- | Signal |     |                           |                                 |       |      |              |                              |
|--------|-----|---------------------------|---------------------------------|-------|------|--------------|------------------------------|
| Name   | r/w | Adr.                      | Integer                         | Real  | Typ  | Value/Off    | Description                  |
| Out?   | r   | Base<br>1dP<br>2dP<br>3dP | 4280<br>12472<br>20664<br>28856 | 41328 | Enum | Enum_Ausgang | Status of the digital output |
|        |     |                           |                                 |       |      |              | 0 Off                        |
|        |     |                           |                                 |       |      |              | 1 On                         |
| F.Do?  | r/w | Base<br>1dP<br>2dP<br>3dP | 4281<br>12473<br>20665<br>28857 | 41330 | Enum | Enum_Y1      | Status of the digital output |
|        |     |                           |                                 |       |      |              | 0 Off                        |
|        |     |                           |                                 |       |      |              | 1 On                         |

## 17 Out.3

- ConF

Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
O.Act	r/w	Base	4350	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.
		1dP	12542				
		2dP	20734				
		3dP	28926				
						0	Direct / Normally de-energized mode
						1	Inverse / normally closed
O.tYP	r/w	Base	4370	41508	Enum	Enum_OtYP	Signal type selection OUT
		1dP	12562				
		2dP	20754				
		3dP	28946				
						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
Y.1	r/w	Base	4351	41470	Enum	Enum_Y1	Output function: Controller output Y1
		1dP	12543				
		2dP	20735				
		3dP	28927				
						0	Not active
						1	This output provides the controller output Y1.
Y.2	r/w	Base	4352	41472	Enum	Enum_Y2	Output function: Controller output Y2. Caution: Do not confuse the controller output Y2 with the parameter 'Fixed output Y2' !
		1dP	12544				
		2dP	20736				
		3dP	28928				
						0	Not active
						1	This output provides the controller output Y2.
Lim.1	r/w	Base	4353	41474	Enum	Enum_Lim1	Output function: Signal limit 1
		1dP	12545				
		2dP	20737				
		3dP	28929				
						0	Not active.
						1	This output is activated by an alarm from limit value 1.
Lim.2	r/w	Base	4354	41476	Enum	Enum_Lim2	Output function: Signal limit 2
		1dP	12546				
		2dP	20738				
		3dP	2930				
						0	Not active.
						1	This output is activated by an alarm from limit value 2
Lim.3	r/w	Base	4355	41478	Enum	Enum_Lim3	Output function: Signal limit 3
		1dP	12547				
		2dP	20739				
		3dP	28931				
						0	Not active
						1	This output is activated by an alarm from limit value 3
LP.AL	r/w	Base	4357	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.
		1dP	12549				
		2dP	20741				
		3dP	28933				
						0	Not active.
						1	This 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 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 This output is 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 This output is activated by the timer status
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'.
P.End	r/w	Base 1dP 2dP 3dP	4261 12453 20645 28837	51290	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 This output sends the error message 'INP1 fault'.
FAi.2	r/w	Base 1dP 2dP 3dP	4263 12455 20647 28839	51294	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 This output sends the error message 'INP2 fault'.
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'.
PrG.1	r/w	Base 1dP 2dP 3dP	4365 12557 20749 28941	41498	Enum	Enum_PrG1	Output function: Signal programmer's control output no. 1. A control output is one of the four digital signals that can be operated segment-wise by a program.
							0 Not active.
							1 Control output 1 is assigned to this output.

PrG.2	r/w	Base 1dP 2dP 3dP	4366 12558 20750 28942	41500	Enum	Enum_PrG2	Output function: Signal programmer's control output no 2. A control output is one of the four digital signals that can be operated segment-wise by a program.
							0 Not active.
							1 Control output 2 is assigned to this output
PrG.3	r/w	Base 1dP 2dP 3dP	4367 12559 20751 28944	41502	Enum	Enum_PrG3	Output function: Signal programmer's control output no. 3. T A control output is one of the four digital signals that can be operated segment-wise by a program.
							0 Not active.
							1 Control output 3 is assigned to this output.
PrG.4	r/w	Base 1dP 2dP 3dP	4368 12560 20752 28944	41504	Enum	Enum_PrG4	Output function: Signal programmer's control output no. 4. A control output is one of the four digital signals that can be operated segment-wise by a program.
							0 Not active.
							1 Control output 4 is assigned to this output.
CALL	r/w	Base 1dP 2dP 3dP	4369 12561 20753 28945	41506	Enum	Enum_CALL	Output: Operator call. At the end of a program segment, a contact is set, e.g. for an acoustic signal. This indicates to the operator that a certain program status has been reached, and operator action is required. Operator calling is used, if the program may only be continued after a check or some kind of operator action.
							0 Not active.
							1 This output is switched by an operator call.
Out.0	r/w	Base 1dP 2dP 3dP	4371 12563 20755 28947	41510	Float	-1999...9999 <input type="checkbox"/>	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
O.Src	r/w	Base 1dP 2dP 3dP	4373 12565 20757 28949	41514	Enum	Enum_OSrc	Signal source of the analog output.
							0 not active
							1 controller output y1 (cont.)
							2 controller output y2 (cont.)
							3 process value
							4 effective setpoint Weff
							5 Control deviation xw (process value - setpoint)= relative alarm Note: Monitoring with the effective setpoint Weff. For example using a ramp it is the changing setpoint, not the target setpoint of the ramp.

• **Signal**

Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
Out?	r	Base	4280	41328	Enum	Enum_Ausgang	Status of the digital output
		1dP	12472				
		2dP	20664				
		3dP	28856				
						0	Off
						1	On
F.Do?	r/w	Base	4381	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).
		1dP	12573				
		2dP	20765				
		3dP	28957				
						0	Off
						1	On
F.Out?	r/w	Base	4382	41532	Float	-1999...9999	Forcing value of the analog output. Forcing involves the external operation of an output, i.e. the instrument has an influence on this output. (Used for the operation of free outputs e.g. by a supervisory PLC.)
		1dP	12574				
		2dP	20766				
		3dP	28958				

## 18 Out.4

• **ConF**

Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
O.Act	r/w	Base	4450	41668	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	12642				
		2dP	29834				
		3dP	29026				
						0	Direct / Normally de-energized mode
						1	Inverse / normally closed
Y.1	r/w	Base	4451	41670	Enum	Enum_Y1	Output function: Controller output Y1
		1dP	12643				
		2dP	20835				
		3dP	29027				
						0	Not active
						1	This output provides the controller output Y1.
Y.2	r/w	Base	4452	41672	Enum	Enum_Y2	Output function: Controller output Y2. Caution: Do not confuse the controller output Y2 with the parameter 'Fixed output Y2' !
		1dP	12644				
		2dP	20836				
		3dP	29028				
						0	Not active
						1	This output provides the controller output Y2.
Lim.1	r/w	Base	4453	41674	Enum	Enum_Lim1	Output function: Signal limit 1
		1dP	12645				
		2dP	20837				
		3dP	29029				
						0	Not active
						1	This output is activated by an alarm from limit value 1.
Lim.2	r/w	Base	4454	41676	Enum	Enum_Lim2	Output function: Signal limit 2
		1dP	12646				
		2dP	20838				
		3dP	29030				
						0	Not active.
						1	This 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	4455 12647 20839 29031	41678	Enum	Enum_Lim3	Output function: Signal limit 3
							0 Not active
							1 This output is activated by an alarm from limit value 3
LPAL	r/w	Base 1dP 2dP 3dP	4457 12649 20841 29033	41682	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 This loop alarm (= open loop alarm) is assigned to this output.
HC.AL	r/w	Base 1dP 2dP 3dP	4458 2650 0842 9034	1684	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	4459 2651 0843 9035	1686	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 This output is activated by an SSR fault.
timE	r/w	Base 1dP 2dP 3dP	4460 2652 0844 9036	1688	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 This output is activated by the timer status
P.End	r/w	Base 1dP 2dP 3dP	4461 2653 0845 9037	1690	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'.
t.End	r/w	Base 1dP 2dP 3dP	4476 2668 0860 9052	1720	Enum	Enum_tEnd	Output function: Signal Program end. This message is available when the program has been completed (only when configured as a timer controller).
							0 Not active.
							1 This output is activated by the message 'Timer end'.
FAi.1	r/w	Base 1dP 2dP 3dP	4462 2654 0846 9038	1692	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 This output sends the error message 'INP1 fault'.
FAi.2	r/w	Base 1dP 2dP 3dP	4463 2655 0847 9039	1694	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 This output sends the error message 'INP2 fault'.

Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
PrG.1	r/w	Base 1dP 2dP 3dP	4465 12657 20849 29041	41698	Enum	Enum_PrG1	Output function: Signal programmer's control output no. 1. A control output is one of the four digital signals that can be operated segment-wise by a program.
							0 Not active.
							1 Control output 1 is assigned to this output.
PrG.2	r/w	Base 1dP 2dP 3dP	4366 12558 20750 28942	41500	Enum	Enum_PrG2	Output function: Signal programmer's control output no. 2. A control output is one of the four digital signals that can be operated segment-wise by a program.
							0 Not active.
							1 Control output 2 is assigned to this output
PrG.3	r/w	Base 1dP 2dP 3dP	4467 12659 20851 29043	41702	Enum	Enum_PrG3	Output function: Signal programmer's control output no. 3. T A control output is one of the four digital signals that can be operated segment-wise by a program.
							0 Not active.
							1 Control output 3 is assigned to this output.
PrG.4	r/w	Base 1dP 2dP 3dP	4468 12660 20852 29044	41704	Enum	Enum_PrG4	Output function: Signal programmer's control output no. 4. A control output is one of the four digital signals that can be operated segment-wise by a program.
							0 Not active.
							1 Control output 4 is assigned to this output.
CALL	r/w	Base 1dP 2dP 3dP	4469 12661 20853 29045	41706	Enum	Enum_CALL	Output: Operator call. At the end of a program segment, a contact is set, e.g. for an acoustic signal. This indicates to the operator that a certain program status has been reached, and operator action is required. Operator calling is used, if the program may only be continued after a check or some kind of operator action.
							0 Not active.
							1 This output is switched by an operator call.

### Signal

Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
Out?	r	Base 1dP 2dP 3dP	4480 12672 20864 29056	41728	Enum	Enum_Ausgang	Status of the digital output
							0 Off
							1 On
F.Do?	r/w	Base 1dP 2dP 3dP	4481 12673 20865 29057	41730	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

## 19 Out.5

### ConF

Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
O.Act	r/w	Base	4550	41868	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	12742				
		2dP	20934				
		3dP	29126				
							0 Direct / Normally de-energized mode
							1 Inverse / normally closed
O.TyP	r/w	Base	4570	41908	Enum	Enum_OtYp	Signal type selection OUT
		1dP	12762				
		2dP	20954				
		3dP	29146				
							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
Y.1	r/w	Base	4551	41870	Enum	Enum_Y1	Output function: Controller output Y1
		1dP	12743				
		2dP	20935				
		3dP	29127				
							0 Not active
							1 This output provides the controller output Y1.
Y.2	r/w	Base	4552	41872	Enum	Enum_Y2	Output function: Controller output Y2. Caution: Do not confuse the controller output Y2 with the parameter 'Fixed output Y2' !
		1dP	12744				
		2dP	20936				
		3dP	29128				
							0 Not active
							1 This output provides the controller output Y2.
Lim.1	r/w	Base	4553	41874	Enum	Enum_Lim1	Output function: Signal limit 1
		1dP	12745				
		2dP	20937				
		3dP	29129				
							0 Not active
							1 This output is activated by an alarm from limit value 1.
Lim.2	r/w	Base	4554	41876	Enum	Enum_Lim2	Output function: Signal limit 2
		1dP	12746				
		2dP	20938				
		3dP	29130				
							0 Not active.
							1 This output is activated by an alarm from limit value 2
Lim.3	r/w	Base	4555	41878	Enum	Enum_Lim3	Output function: Signal limit 3
		1dP	12747				
		2dP	20939				
		3dP	29131				
							0 Not active
							1 This output is activated by an alarm from limit value 3
LPAL	r/w	Base	4557	41882	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	12749				
		2dP	20941				
		3dP	29133				
							0 Not active.
							1 This 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 1dP 2dP 3dP	4558 12750 20942 29134	41884	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	4559 12751 20943 29135	41886	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 This output is activated by an SSR fault.
timE	r/w	Base 1dP 2dP 3dP	4560 12752 20944 29136	41888	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 This output is activated by the timer status
P.End	r/w	Base 1dP 2dP 3dP	4561 12753 20945 29137	41890	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'.
t.End	r/w	Base 1dP 2dP 3dP	4576 12768 20960 29152	41920	Enum	Enum_TEnd	Output function: Signal Timer end. This message is available when the program has been completed (only when configured as a timer controller).
							0 Not active.
							1 This output is activated by the message 'Timer end'.
FAi.1	r/w	Base 1dP 2dP 3dP	4562 12754 20946 29138	41692	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 This output sends the error message 'INP1 fault'.
FAi.2	r/w	Base 1dP 2dP 3dP	4563 12755 20947 29139	61894	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 This output sends the error message 'INP2 fault'.
PrG.1	r/w	Base 1dP 2dP 3dP	4565 12757 20949 29141	41898	Enum	Enum_PrG1	Output function: Signal programmer's control output no. 1. A control output is one of the four digital signals that can be operated segment-wise by a program.
							0 Not active.
							1 Control output 1 is assigned to this output.
PrG.2	r/w	Base 1dP 2dP 3dP	4566 12758 20950 29142	41900	Enum	Enum_PrG2	Output function: Signal programmer's control output no 2. A control output is one of the four digital signals that can be operated segment-wise by a program.
							0 Not active.
							1 Control output 2 is assigned to this output

Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
PrG.3	r/w	Base 1dP 2dP 3dP	4567 12759 20951 29143	41902	Enum	Enum_PrG3	Output function: Signal programmer's control output no. 3. T A control output is one of the four digital signals that can be operated segment-wise by a program.
							0 Not active.
							1 Control output 3 is assigned to this output.
PrG.4	r/w	Base 1dP 2dP 3dP	4568 12760 20952 29144	41904	Enum	Enum_PrG4	Output function: Signal programmer's control output no. 4. A control output is one of the four digital signals that can be operated segment-wise by a program.
							0 Not active.
							1 Control output 4 is assigned to this output.
CALL	r/w	Base 1dP 2dP 3dP	4569 12761 20953 29145	41906	Enum	Enum_CALL	Output: Operator call. At the end of a program segment, a contact is set, e.g. for an acoustic signal. This indicates to the operator that a certain program status has been reached, and operator action is required. Operator calling is used, if the program may only be continued after a check or some kind of operator action.
							0 Not active.
							1 This output is switched by an operator call.
Out.0	r/w	Base 1dP 2dP 3dP	4571 12763 20955 29147	41910	Float	-1999...9999 <input type="checkbox"/>	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	4572 12764 20956 29148	41912	Float	-1999...9999 <input type="checkbox"/>	Upper scaling limit of the analog output (corresponds to 100%). If current or voltage signals are used as output values, scaling of the display can be applied to the output value by means of the Parameter Level. Definition of the upper output limit is done using the corresponding electrical value (mA / V).
O.Src	r/w	Base 1dP 2dP 3dP	4573 12765 20957 29149	41914	Enum	Enum_OSrc	Signal source of the analog output.
							0 not active
							1 controller output y1 (cont.)
							2 controller output y2 (cont.)
							3 process value
							4 effective setpoint Weff
							5 Control deviation xw (process value - setpoint)= relative alarm Note: Monitoring with the effective setpoint Weff. For example using a ramp it is the changing setpoint, not the target setpoint of the ramp.

Signal							
Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
Out?	r	Base	4580	41928	Enum	Enum_Ausgang	Status of the digital output
		1dP	12772				
		2dP	20964				
		3dP	29156				
						0	Off
						1	On
F.Do?	r/w	Base	4581	41730	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	12773				
		2dP	20965				
		3dP	29157				
						0	Off
						1	On
F.Out?	r/w	Base	4582	41732	Float	0...120 <input type="checkbox"/>	Forcing value of the analog output. Forcing involves the external operation of an output, i.e. the instrument has an influence on this output. (Used for the operation of free outputs e.g. by a supervisory PLC.)
		1dP	12774				
		2dP	20966				
		3dP	29158				
						0	Off
						1	On

## 20 Out.6

ConF							
Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
O.Act	r/w	Base	4650	42068	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	12842				
		2dP	21034				
		3dP	29226				
						0	Direct / Normally de-energized mode
						1	Inverse / normally closed
Y.1	r/w	Base	4651	42079	Enum	Enum_Y1	Output function: Controller output Y1
		1dP	12843				
		2dP	21035				
		3dP	29227				
						0	Not active
						1	This output provides the controller output Y1.
Y.2	r/w	Base	4652	42072	Enum	Enum_Y2	Output function: Controller output Y2. Caution: Do not confuse the controller output Y2 with the parameter 'Fixed output Y2' !
		1dP	12844				
		2dP	21036				
		3dP	29228				
						0	Not active
						1	This output provides the controller output Y2.
Lim.1	r/w	Base	4653	42074	Enum	Enum_Lim1	Output function: Signal limit 1
		1dP	12845				
		2dP	21037				
		3dP	29229				
						0	Not active
						1	This output is activated by an alarm from limit value 1.
Lim.2	r/w	Base	4654	42076	Enum	Enum_Lim2	Output function: Signal limit 2
		1dP	12846				
		2dP	21038				
		3dP	29230				
						0	Not active.
						1	This 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	4655 12847 21039 29231	42078	Enum	Enum_Lim3	Output function: Signal limit 3
							0 Not active
							1 This output is activated by an alarm from limit value 3
LPAL	r/w	Base 1dP 2dP 3dP	4657 12849 21041 29233	42082	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 This loop alarm (= open loop alarm) is assigned to this output.
HC.AL	r/w	Base 1dP 2dP 3dP	4658 12850 21042 29234	42084	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	4659 12851 21043 29235	42086	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 This output is activated by an SSR fault.
timE	r/w	Base 1dP 2dP 3dP	4660 12852 21044 29236	42088	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 This output is activated by the timer status
P.End	r/w	Base 1dP 2dP 3dP	4661 12853 21045 29237	42090	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'.
t.End	r/w	Base 1dP 2dP 3dP	4676 12868 21060 29252	42120	Enum	Enum_TEnd	Output function: Signal Timer end. This message is available when the program has been completed (only when configured as a timer controller).
							0 Not active.
							1 This output is activated by the message 'Timer end'.
FAi.1	r/w	Base 1dP 2dP 3dP	4662 12854 21046 29238	42092	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 This output sends the error message 'INP1 fault'.
FAi.2	r/w	Base 1dP 2dP 3dP	4663 12855 21047 29239	42094	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 This output sends the error message 'INP2 fault'.

Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
PrG.1	r/w	Base 1dP 2dP 3dP	4665 12857 21049 29241	42098	Enum	Enum_PrG1	Output function: Signal programmer's control output no. 1. A control output is one of the four digital signals that can be operated segment-wise by a program.
							0 Not active.
							1 Control output 1 is assigned to this output.
PrG.2	r/w	Base 1dP 2dP 3dP	4666 12858 21050 29242	42100	Enum	Enum_PrG2	Output function: Signal programmer's control output no. 2. A control output is one of the four digital signals that can be operated segment-wise by a program.
							0 Not active.
							1 Control output 2 is assigned to this output
PrG.3	r/w	Base 1dP 2dP 3dP	4667 12859 21051 29243	42102	Enum	Enum_PrG3	Output function: Signal programmer's control output no. 3. T A control output is one of the four digital signals that can be operated segment-wise by a program.
							0 Not active.
							1 Control output 3 is assigned to this output.
PrG.4	r/w	Base 1dP 2dP 3dP	4668 12860 21052 29244	42104	Enum	Enum_PrG4	Output function: Signal programmer's control output no. 4. A control output is one of the four digital signals that can be operated segment-wise by a program.
							0 Not active.
							1 Control output 4 is assigned to this output.
CALL	r/w	Base 1dP 2dP 3dP	4669 12861 21053 29245	42106	Enum	Enum_CALL	Output: Operator call. At the end of a program segment, a contact is set, e.g. for an acoustic signal. This indicates to the operator that a certain program status has been reached, and operator action is required. Operator calling is used, if the program may only be continued after a check or some kind of operator action.
							0 Not active.
							1 This output is switched by an operator call.

### Signal

Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
Out?	r	Base 1dP 2dP 3dP	4680 12872 21064 29256	41928	Enum	Enum_Ausgang	Status of the digital output
							0 Off
							1 On
F.Do?	r/w	Base 1dP 2dP 3dP	4681 12873 21065 29257	42130	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

## 21 PAr.2

- PArA

Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
Pb12	r/w	Base 1dP 2dP 3dP	5030 13222 21414 29606	42828	Float	0.1...9999 <input type="checkbox"/>	Proportional band 1 (heating) in engineering unit (e.g. °C) of the 2nd parameter set. The Pb defines the ratio between output value and control deviation. The smaller the value of Pb is, the stronger is the control response for a specific control deviation. Too large and too small values for Pb lead to process oscillations (hunting).
Pb22	r/w	Base 1dP 2dP 3dP	5031 13223 21415 29607	42830	Float	0.1...9999 <input type="checkbox"/>	Proportional band 2 (cooling) in engineering unit (e.g. °C) of the 2nd parameter set. The Pb defines the ratio between output value and control deviation. The smaller the value of Pb is, the stronger is the control response for a specific control deviation. Too large and too small values for Pb lead to process oscillations (hunting).
ti22	r/w	Base 1dP 2dP 3dP	5033 13225 21417 29609	42834	Float	0...9999 <input checked="" type="checkbox"/>	Integral action time 2 (cooling) [s]. 2nd parameter set. T 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.
ti12	r/w	Base 1dP 2dP 3dP	5032 13224 21416 29608	42832	Float	0...9999 <input checked="" type="checkbox"/>	Integral action time 1 (heating) [s]. 2nd parameter set. T 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.
td12	r/w	Base 1dP 2dP 3dP	5034 13226 21418 29610	42836	Float	0...9999 <input checked="" type="checkbox"/>	Derivative action time 1 (heating) [s], 2nd 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.
td22	r/w	Base 1dP 2dP 3dP	5035 13227 21419 29611	42838	Float	0...9999 <input checked="" type="checkbox"/>	Derivative action time 2 (cooling) [s], 2nd 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.

## 22 ProG

- ConF

Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description		
t.bAS	r/w	Base	6030	44828	Enum	Enum_tbAS	Definition of the programmer's time base in hours using minutes, or in minutes using seconds.		
		1dP	14222						
		2dP	22414						
		3dP	30606						
						0	Hours [hh] : Minutes [mm]		
						2	Minutes [mm] : Seconds [ss]		
Pr.no	r/w	Base	6000	44768	Enum	Enum_PrgNoPar	Program number (nominal). The program number (nominal) determines which program is to be started next. Running programs are not affected. The selected program is only started after a reset or restart.		
		1dP	14192						
		2dP	22384						
		3dP	30576						
								1	prog. 01
								2	prog. 02
								3	prog. 03
								4	prog. 04
								5	prog. 05
								6	prog. 06
								7	prog. 07
								8	prog. 08
								9	prog. 09
								10	prog. 10
								11	prog. 11
								12	prog. 12
								13	prog. 13
		14	prog. 14						
		15	prog. 15						
		16	prog. 16						
b.Lo	r/w	Base 1dP 2dP 3dP	6100 14292 22484 30676	44968	Float	0...9999 <input type="checkbox"/>	Lower bandwidth limit. The bandwidth monitor is valid for all segments of an individual program. If the bandwidth is exceeded, the programmer is stopped. The program continues, if the process value returns within the defined monitoring limits.		
b.Hi	r/w	Base 1dP 2dP 3dP	6101 14293 22485 30677	44970	Float	0...9999 <input type="checkbox"/>	Upper bandwidth limit. The bandwidth monitor is valid for all segments of an individual program. If the bandwidth is exceeded, the programmer is stopped. The program continues, if the process value returns within the defined monitoring limits.		

Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
d.00	r/w	Base 1dP 2dP 3dP	6134 14326 22518 30710	45036	Enum	Enum_Spuren	Reset value for control outputs 1...4. A program can control up to four digital signals: the control outputs 1...4 The reset value of the control output contains the combination of these signals, which are output together with the controller's internal setpoint, if the programmer is not active.
						0	0-0-0-0
						1	1-0-0-0
						2	0-1-0-0
						3	1-1-0-0
						4	0-0-1-0
						5	1-0-1-0
						6	0-1-1-0
						7	1-1-1-0
						8	0-0-0-1
						9	1-0-0-1
						10	0-1-0-1
						11	1-1-0-1
						12	0-0-1-1
						13	1-0-1-1
						14	0-1-1-1
						15	1-1-1-1
tYPE	r/w	Base 1dP 2dP 3dP	6135 14327 22519 30711	45038	Enum	Enum_SegTyp	Type of segment 1. The segment type defines the setpoint behavior for this segment. The setpoint can be held constant or be changed with a ramp or a step function. Continuation to next segment is automatic or manual (define a hold time).Note: The 1st segment cannot be configured as the end segment.
						0	time to setpoint
						1	rate to setpoint
						2	The final setpoint of the previous segment is kept constant for the duration 'Pt'.
						3	step to setpoint
						4	time to setpoint and wait
						5	rate to setpoint and wait
						6	The final setpoint of the previous segment is kept constant for the duration 'Pt'. At the end of a segment, the programmer enters the Stop mode (Run LED is off), and can be restarted by pressing the Start/Stop key (more than 3 s), via the interface, or a digital input.
						7	step to setpoint and wait
						8	The last segment in a program is the end segment. When the end segment has been reached, the last setpoint is maintained.
						9	timer
						10	timer and hold period.
SP	r/w	Base 1dP 2dP 3dP	6102 14294 22486 60678	44972	Float	-1999...9999 <input type="checkbox"/>	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	r/w	Base 1dP 2dP 3dP	6103 14295 22487 30679	44974	Float	0...9999 <input type="checkbox"/>	Segment time/gradient 1. The duration of a segment can be defined directly, or by using the segment time and the setpoint difference (SP – segment starting setpoint). Whether the setting is for segment time or the gradient, is defined by means of the segment type parameter (tYPE).

Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
d.Out	r/w	Base 1dP 2dP 3dP	6136 14328 22520 30712	45040	Enum	Enum_Spuren	Control outputs 1...4 - 2. A program can control up to four digital signals: the control outputs 1...4. A combination of these signals can be assigned to every segment, whereby the signals are operated while the segment is running. For access to the controller's outputs, the signals must be assigned accordingly.
						0	0-0-0-0
						1	1-0-0-0
						2	0-1-0-0
						3	1-1-0-0
						4	0-0-1-0
						5	1-0-1-0
						6	0-1-1-0
						7	1-1-1-0
						8	0-0-0-1
						9	1-0-0-1
						10	0-1-0-1
						11	1-1-0-1
						12	0-0-1-1
						13	1-0-1-1
						14	0-1-1-1
						15	1-1-1-1
t.YPE	r/w	Base 1dP 2dP 3dP	6137 14329 22521 30713	45042	Enum	Enum_SegTyp	Segment type of segment 2. The segment type defines the setpoint behavior for this segment. The setpoint can be held constant or be changed with a ramp or a step function. Continuation to next segment is automatic or manual (define a hold time).
						0	time to setpoint
						1	rate to setpoint
						2	The final setpoint of the previous segment is kept constant for the duration 'Pt'.
						3	step to setpoint
						4	time to setpoint and wait
						5	rate to setpoint and wait
						6	The final setpoint of the previous segment is kept constant for the duration 'Pt'. At the end of a segment, the programmer enters the Stop mode (Run LED is off), and can be restarted by pressing the Start/Stop key (more than 3 s), via the interface, or a digital input.
						7	step to setpoint and wait
						8	The last segment in a program is the end segment. When the end segment has been reached, the last setpoint is maintained.
						9	timer
						10	timer and hold period.
SP	r/w	Base 1dP 2dP 3dP	6104 14296 22488 30680	44976	Float	-1999...9999 <input type="checkbox"/>	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	r/w	Base 1dP 2dP 3dP	6105 14297 22489 30681	44978	Float	0...9999 <input type="checkbox"/>	Segment time/gradient 2. The duration of a segment can be defined directly, or by using the segment time and the setpoint difference (SP – segment starting setpoint). Whether the setting is for segment time or the gradient, is defined by means of the segment type parameter (tYPE).

Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
d.Out	r/w	Base 1dP 2dP 3dP	6138 14330 22522 30714	45044	Enum	Enum_Spuren	Control outputs 1...4 - 2. A program can control up to four digital signals: the control outputs 1...4. A combination of these signals can be assigned to every segment, whereby the signals are operated while the segment is running. For access to the controller's outputs, the signals must be assigned accordingly.
						0	0-0-0-0
						1	1-0-0-0
						2	0-1-0-0
						3	1-1-0-0
						4	0-0-1-0
						5	1-0-1-0
						6	0-1-1-0
						7	1-1-1-0
						8	0-0-0-1
						9	1-0-0-1
						10	0-1-0-1
						11	1-1-0-1
						12	0-0-1-1
						13	1-0-1-1
						14	0-1-1-1
						15	1-1-1-1
t.YPE	r/w	Base 1dP 2dP 3dP	6139 14331 22523 30715	45046	Enum	Enum_SegTyp	Segment type of segment 3. The segment type defines the setpoint behavior for this segment. The setpoint can be held constant or be changed with a ramp or a step function. Continuation to next segment is automatic or manual (define a hold time).
						0	time to setpoint
						1	rate to setpoint
						2	The final setpoint of the previous segment is kept constant for the duration 'Pt'.
						3	step to setpoint
						4	time to setpoint and wait
						5	rate to setpoint and wait
						6	The final setpoint of the previous segment is kept constant for the duration 'Pt'. At the end of a segment, the programmer enters the Stop mode (Run LED is off), and can be restarted by pressing the Start/Stop key (more than 3 s), via the interface, or a digital input.
						7	step to setpoint and wait
						8	The last segment in a program is the end segment. When the end segment has been reached, the last setpoint is maintained.
						9	timer
						10	timer and hold period.
SP	r/w	Base 1dP 2dP 3dP	6106 14298 22490 30682	44980	Float	-1999...9999 <input type="checkbox"/>	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	r/w	Base 1dP 2dP 3dP	6107 14299 22491 30683	44982	Float	0...9999 <input type="checkbox"/>	Segment time/gradient 3. The duration of a segment can be defined directly, or by using the segment time and the setpoint difference (SP – segment starting setpoint). Whether the setting is for segment time or the gradient, is defined by means of the segment type parameter (tYPE).

Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
d.Out	r/w	Base 1dP 2dP 3dP	6140 14332 22524 30716	45048	Enum	Enum_Spuren	Control outputs 1...4 - 3. A program can control up to four digital signals: the control outputs 1...4. A combination of these signals can be assigned to every segment, whereby the signals are operated while the segment is running. For access to the controller's outputs, the signals must be assigned accordingly
						0	0-0-0-0
						1	1-0-0-0
						2	0-1-0-0
						3	1-1-0-0
						4	0-0-1-0
						5	1-0-1-0
						6	0-1-1-0
						7	1-1-1-0
						8	0-0-0-1
						9	1-0-0-1
						10	0-1-0-1
						11	1-1-0-1
						12	0-0-1-1
						13	1-0-1-1
						14	0-1-1-1
						15	1-1-1-1
t.YPE	r/w	Base 1dP 2dP 3dP	6141 14333 22525 30717	45050	Enum	Enum_SegTyp	Segment type of segment 4. The segment type defines the setpoint behavior for this segment. The setpoint can be held constant or be changed with a ramp or a step function. Continuation to next segment is automatic or manual (define a hold time).
						0	time to setpoint
						1	rate to setpoint
						2	The final setpoint of the previous segment is kept constant for the duration 'Pt'.
						3	step to setpoint
						4	time to setpoint and wait
						5	rate to setpoint and wait
						6	The final setpoint of the previous segment is kept constant for the duration 'Pt'. At the end of a segment, the programmer enters the Stop mode (Run LED is off), and can be restarted by pressing the Start/Stop key (more than 3 s), via the interface, or a digital input.
						7	step to setpoint and wait
						8	The last segment in a program is the end segment. When the end segment has been reached, the last setpoint is maintained.
						9	timer
						10	timer and hold period.
SP	r/w	Base 1dP 2dP 3dP	6108 14300 22492 30684	44984	Float	-1999...9999 <input type="checkbox"/>	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	r/w	Base 1dP 2dP 3dP	6109 14301 22493 30685	44986	Float	0...9999 <input type="checkbox"/>	Segment time/gradient 4. The duration of a segment can be defined directly, or by using the segment time and the setpoint difference (SP – segment starting setpoint). Whether the setting is for segment time or the gradient, is defined by means of the segment type parameter (TYPE).

Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
d.Out	r/w	Base 1dP 2dP 3dP	6142 14334 22526 30718	45052	Enum	Enum_Spuren	Control outputs 1...4 - 4. A program can control up to four digital signals: the control outputs 1...4. A combination of these signals can be assigned to every segment, whereby the signals are operated while the segment is running. For access to the controller's outputs, the signals must be assigned accordingly.
							0 0-0-0-0
							1 1-0-0-0
							2 0-1-0-0
							3 1-1-0-0
							4 0-0-1-0
							5 1-0-1-0
							6 0-1-1-0
							7 1-1-1-0
							8 0-0-0-1
							9 1-0-0-1
							10 0-1-0-1
							11 1-1-0-1
							12 0-0-1-1
							13 1-0-1-1
							14 0-1-1-1
							15 1-1-1-1
t.YPE	r/w	Base 1dP 2dP 3dP	6143 14335 22527 30719	45054	Enum	Enum_SegTyp	Segment type of segment 5. The segment type defines the setpoint behavior for this segment. The setpoint can be held constant or be changed with a ramp or a step function. Continuation to next segment is automatic or manual (define a hold time).
							0 time to setpoint
							1 rate to setpoint
							2 The final setpoint of the previous segment is kept constant for the duration 'Pt'.
							3 step to setpoint
							4 time to setpoint and wait
							5 rate to setpoint and wait
							6 The final setpoint of the previous segment is kept constant for the duration 'Pt'. At the end of a segment, the programmer enters the Stop mode (Run LED is off), and can be restarted by pressing the Start/Stop key (more than 3 s), via the interface, or a digital input.
							7 step to setpoint and wait
							8 The last segment in a program is the end segment. When the end segment has been reached, the last setpoint is maintained.
							9 timer
							10 timer and hold period.
SP	r/w	Base 1dP 2dP 3dP	6110 14302 22494 30686	44988	Float	-1999...9999 <input type="checkbox"/>	End setpoint of segment 5. This is the target setpoint that is reached at the end of the fifth 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	r/w	Base 1dP 2dP 3dP	6111 14303 22495 30687	44990	Float	0...9999 <input type="checkbox"/>	Segment time/gradient 5. The duration of a segment can be defined directly, or by using the segment time and the setpoint difference (SP – segment starting setpoint). Whether the setting is for segment time or the gradient, is defined by means of the segment type parameter (TYPE).

Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
d.Out	r/w	Base 1dP 2dP 3dP	6144 14336 22528 30720	45056	Enum	Enum_Spuren	Control outputs 1...4 - 5. A program can control up to four digital signals: the control outputs 1...4. A combination of these signals can be assigned to every segment, whereby the signals are operated while the segment is running. For access to the controller's outputs, the signals must be assigned accordingly.
							0 0-0-0-0
							1 1-0-0-0
							2 0-1-0-0
							3 1-1-0-0
							4 0-0-1-0
							5 1-0-1-0
							6 0-1-1-0
							7 1-1-1-0
							8 0-0-0-1
							9 1-0-0-1
							10 0-1-0-1
							11 1-1-0-1
							12 0-0-1-1
							13 1-0-1-1
							14 0-1-1-1
							15 1-1-1-1
t.YPE	r/w	Base 1dP 2dP 3dP	6145 14337 22529 30721	45058	Enum	Enum_SegTyp	Segment type of segment 6. The segment type defines the setpoint behavior for this segment. The setpoint can be held constant or be changed with a ramp or a step function. Continuation to next segment is automatic or manual (define a hold time).
							0 time to setpoint
							1 rate to setpoint
							2 The final setpoint of the previous segment is kept constant for the duration 'Pt'.
							3 step to setpoint
							4 time to setpoint and wait
							5 rate to setpoint and wait
							6 The final setpoint of the previous segment is kept constant for the duration 'Pt'. At the end of a segment, the programmer enters the Stop mode (Run LED is off), and can be restarted by pressing the Start/Stop key (more than 3 s), via the interface, or a digital input.
							7 step to setpoint and wait
							8 The last segment in a program is the end segment. When the end segment has been reached, the last setpoint is maintained.
							9 timer
							10 timer and hold period.
SP	r/w	Base 1dP 2dP 3dP	6112 14304 22496 30688	44992	Float	-1999...9999 <input type="checkbox"/>	End setpoint of segment 6. This is the target setpoint that is reached at the end of the sixth 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	r/w	Base 1dP 2dP 3dP	6113 14305 22497 30689	44994	Float	0...9999 <input type="checkbox"/>	Segment time/gradient 6. The duration of a segment can be defined directly, or by using the segment time and the setpoint difference (SP – segment starting setpoint). Whether the setting is for segment time or the gradient, is defined by means of the segment type parameter (TYPE).

Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
d.Out	r/w	Base 1dP 2dP 3dP	6146 14338 22530 30722	45060	Enum	Enum_Spuren	Control outputs 1...4 - 6. A program can control up to four digital signals: the control outputs 1...4. A combination of these signals can be assigned to every segment, whereby the signals are operated while the segment is running. For access to the controller's outputs, the signals must be assigned accordingly.
						0	0-0-0-0
						1	1-0-0-0
						2	0-1-0-0
						3	1-1-0-0
						4	0-0-1-0
						5	1-0-1-0
						6	0-1-1-0
						7	1-1-1-0
						8	0-0-0-1
						9	1-0-0-1
						10	0-1-0-1
						11	1-1-0-1
						12	0-0-1-1
						13	1-0-1-1
						14	0-1-1-1
						15	1-1-1-1
t.YPE	r/w	Base 1dP 2dP 3dP	6147 14339 22531 30723	45062	Enum	Enum_SegTyp	Segment type of segment 7. The segment type defines the setpoint behavior for this segment. The setpoint can be held constant or be changed with a ramp or a step function. Continuation to next segment is automatic or manual (define a hold time).
						0	time to setpoint
						1	rate to setpoint
						2	The final setpoint of the previous segment is kept constant for the duration 'Pt'.
						3	step to setpoint
						4	time to setpoint and wait
						5	rate to setpoint and wait
						6	The final setpoint of the previous segment is kept constant for the duration 'Pt'. At the end of a segment, the programmer enters the Stop mode (Run LED is off), and can be restarted by pressing the Start/Stop key (more than 3 s), via the interface, or a digital input.
						7	step to setpoint and wait
						8	The last segment in a program is the end segment. When the end segment has been reached, the last setpoint is maintained.
						9	timer
						10	timer and hold period.
SP	r/w	Base 1dP 2dP 3dP	6114 14306 22498 30690	44996	Float	-1999...9999 <input type="checkbox"/>	End setpoint of segment 7. This is the target setpoint that is reached at the end of the seventh 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	r/w	Base 1dP 2dP 3dP	6115 14307 22499 30691	44998	Float	0...9999 <input type="checkbox"/>	Segment time/gradient 7. The duration of a segment can be defined directly, or by using the segment time and the setpoint difference (SP – segment starting setpoint). Whether the setting is for segment time or the gradient, is defined by means of the segment type parameter (TYPE).

Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
d.Out	r/w	Base 1dP 2dP 3dP	6148 14340 22532 30724	45064	Enum	Enum_Spuren	Control outputs 1...4 - 7. A program can control up to four digital signals: the control outputs 1...4. A combination of these signals can be assigned to every segment, whereby the signals are operated while the segment is running. For access to the controller's outputs, the signals must be assigned accordingly.
						0	0-0-0-0
						1	1-0-0-0
						2	0-1-0-0
						3	1-1-0-0
						4	0-0-1-0
						5	1-0-1-0
						6	0-1-1-0
						7	1-1-1-0
						8	0-0-0-1
						9	1-0-0-1
						10	0-1-0-1
						11	1-1-0-1
						12	0-0-1-1
						13	1-0-1-1
						14	0-1-1-1
						15	1-1-1-1
t.YPE	r/w	Base 1dP 2dP 3dP	6149 14341 22533 30725	45066	Enum	Enum_SegTyp	Segment type of segment 8. The segment type defines the setpoint behavior for this segment. The setpoint can be held constant or be changed with a ramp or a step function. Continuation to next segment is automatic or manual (define a hold time).
						0	time to setpoint
						1	rate to setpoint
						2	The final setpoint of the previous segment is kept constant for the duration 'Pt'.
						3	step to setpoint
						4	time to setpoint and wait
						5	rate to setpoint and wait
						6	The final setpoint of the previous segment is kept constant for the duration 'Pt'. At the end of a segment, the programmer enters the Stop mode (Run LED is off), and can be restarted by pressing the Start/Stop key (more than 3 s), via the interface, or a digital input.
						7	step to setpoint and wait
						8	The last segment in a program is the end segment. When the end segment has been reached, the last setpoint is maintained.
						9	timer
						10	timer and hold period.
SP	r/w	Base 1dP 2dP 3dP	6116 14308 22500 30692	45000	Float	-1999...9999 <input type="checkbox"/>	End setpoint of segment 8. This is the target setpoint that is reached at the end of the eighth 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	r/w	Base 1dP 2dP 3dP	6117 14309 22501 30693	45002	Float	0...9999 <input type="checkbox"/>	Segment time/gradient 8. The duration of a segment can be defined directly, or by using the segment time and the setpoint difference (SP – segment starting setpoint). Whether the setting is for segment time or the gradient, is defined by means of the segment type parameter (YPE).

Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
d.Out	r/w	Base 1dP 2dP 3dP	6150 14342 22534 30726	45068	Enum	Enum_Spuren	Control outputs 1...4 - 8. A program can control up to four digital signals: the control outputs 1...4. A combination of these signals can be assigned to every segment, whereby the signals are operated while the segment is running. For access to the controller's outputs, the signals must be assigned accordingly.
							0 0-0-0-0
							1 1-0-0-0
							2 0-1-0-0
							3 1-1-0-0
							4 0-0-1-0
							5 1-0-1-0
							6 0-1-1-0
							7 1-1-1-0
							8 0-0-0-1
							9 1-0-0-1
							10 0-1-0-1
							11 1-1-0-1
							12 0-0-1-1
							13 1-0-1-1
							14 0-1-1-1
							15 1-1-1-1
t.YPE	r/w	Base 1dP 2dP 3dP	6151 14343 22535 30727	45070	Enum	Enum_SegTyp	Segment type of segment 9. The segment type defines the setpoint behavior for this segment. The setpoint can be held constant or be changed with a ramp or a step function. Continuation to next segment is automatic or manual (define a hold time).
							0 time to setpoint
							1 rate to setpoint
							2 The final setpoint of the previous segment is kept constant for the duration 'Pt'.
							3 step to setpoint
							4 time to setpoint and wait
							5 rate to setpoint and wait
							6 The final setpoint of the previous segment is kept constant for the duration 'Pt'. At the end of a segment, the programmer enters the Stop mode (Run LED is off), and can be restarted by pressing the Start/Stop key (more than 3 s), via the interface, or a digital input.
							7 step to setpoint and wait
							8 The last segment in a program is the end segment. When the end segment has been reached, the last setpoint is maintained.
							9 timer
							10 timer and hold period.
SP	r/w	Base 1dP 2dP 3dP	6118 14310 22502 30694	45004	Float	-1999...9999 <input type="checkbox"/>	End setpoint of segment 9. This is the target setpoint that is reached at the end of the ninth 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	r/w	Base 1dP 2dP 3dP	6119 14311 22503 30695	45006	Float	0...9999 <input type="checkbox"/>	Segment time/gradient 9. The duration of a segment can be defined directly, or by using the segment time and the setpoint difference (SP – segment starting setpoint). Whether the setting is for segment time or the gradient, is defined by means of the segment type parameter (TYPE).

Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
d.Out	r/w	Base 1dP 2dP 3dP	6152 14344 22536 30728	45072	Enum	Enum_Spuren	Control outputs 1...4 - 9. A program can control up to four digital signals: the control outputs 1...4. A combination of these signals can be assigned to every segment, whereby the signals are operated while the segment is running. For access to the controller's outputs, the signals must be assigned accordingly.
						0	0-0-0-0
						1	1-0-0-0
						2	0-1-0-0
						3	1-1-0-0
						4	0-0-1-0
						5	1-0-1-0
						6	0-1-1-0
						7	1-1-1-0
						8	0-0-0-1
						9	1-0-0-1
						10	0-1-0-1
						11	1-1-0-1
						12	0-0-1-1
						13	1-0-1-1
						14	0-1-1-1
						15	1-1-1-1
t.YPE	r/w	Base 1dP 2dP 3dP	6153 14345 22537 30729	45074	Enum	Enum_SegTyp	Segment type of segment 10. The segment type defines the setpoint behavior for this segment. The setpoint can be held constant or be changed with a ramp or a step function. Continuation to next segment is automatic or manual (define a hold time).
						0	time to setpoint
						1	rate to setpoint
						2	The final setpoint of the previous segment is kept constant for the duration 'Pt'.
						3	step to setpoint
						4	time to setpoint and wait
						5	rate to setpoint and wait
						6	The final setpoint of the previous segment is kept constant for the duration 'Pt'. At the end of a segment, the programmer enters the Stop mode (Run LED is off), and can be restarted by pressing the Start/Stop key (more than 3 s), via the interface, or a digital input.
						7	step to setpoint and wait
						8	The last segment in a program is the end segment. When the end segment has been reached, the last setpoint is maintained.
						9	timer
						10	timer and hold period.
SP	r/w	Base 1dP 2dP 3dP	6120 14312 22504 30696	45008	Float	-1999...9999 <input type="checkbox"/>	End setpoint of segment 10. This is the target setpoint that is reached at the end of the tenth 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	r/w	Base 1dP 2dP 3dP	6121 14313 22505 30697	45010	Float	0...9999 <input type="checkbox"/>	Segment time/gradient 10. The duration of a segment can be defined directly, or by using the segment time and the setpoint difference (SP – segment starting setpoint). Whether the setting is for segment time or the gradient, is defined by means of the segment type parameter (TYPE).

Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
d.Out	r/w	Base 1dP 2dP 3dP	6153 14345 22537 30729	45076	Enum	Enum_Spuren	Control outputs 1...4 - 10. A program can control up to four digital signals: the control outputs 1...4. A combination of these signals can be assigned to every segment, whereby the signals are operated while the segment is running. For access to the controller's outputs, the signals must be assigned accordingly.
						0	0-0-0-0
						1	1-0-0-0
						2	0-1-0-0
						3	1-1-0-0
						4	0-0-1-0
						5	1-0-1-0
						6	0-1-1-0
						7	1-1-1-0
						8	0-0-0-1
						9	1-0-0-1
						10	0-1-0-1
						11	1-1-0-1
						12	0-0-1-1
						13	1-0-1-1
						14	0-1-1-1
						15	1-1-1-1
t.YPE	r/w	Base 1dP 2dP 3dP	6155 14347 22539 30731	45078	Enum	Enum_SegTyp	Segment type of segment 11. The segment type defines the setpoint behavior for this segment. The setpoint can be held constant or be changed with a ramp or a step function. Continuation to next segment is automatic or manual (define a hold time).
						0	time to setpoint
						1	rate to setpoint
						2	The final setpoint of the previous segment is kept constant for the duration 'Pt'.
						3	step to setpoint
						4	time to setpoint and wait
						5	rate to setpoint and wait
						6	The final setpoint of the previous segment is kept constant for the duration 'Pt'. At the end of a segment, the programmer enters the Stop mode (Run LED is off), and can be restarted by pressing the Start/Stop key (more than 3 s), via the interface, or a digital input.
						7	step to setpoint and wait
						8	The last segment in a program is the end segment. When the end segment has been reached, the last setpoint is maintained.
						9	timer
						10	timer and hold period.
SP	r/w	Base 1dP 2dP 3dP	6122 14314 22506 30698	45012	Float	-1999...9999 <input type="checkbox"/>	End setpoint of segment 11. This is the target setpoint that is reached at the end of the eleventh 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	r/w	Base 1dP 2dP 3dP	6123 14315 22507 30699	45014	Float	0...9999 <input type="checkbox"/>	Segment time/gradient 11. The duration of a segment can be defined directly, or by using the segment time and the setpoint difference (SP – segment starting setpoint). Whether the setting is for segment time or the gradient, is defined by means of the segment type parameter (tYPE).

Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
d.Out	r/w	Base 1dP 2dP 3dP	6156 14348 22540 30732	45080	Enum	Enum_Spuren	Control outputs 1...4 - 11. A program can control up to four digital signals: the control outputs 1...4. A combination of these signals can be assigned to every segment, whereby the signals are operated while the segment is running. For access to the controller's outputs, the signals must be assigned accordingly.
						0	0-0-0-0
						1	1-0-0-0
						2	0-1-0-0
						3	1-1-0-0
						4	0-0-1-0
						5	1-0-1-0
						6	0-1-1-0
						7	1-1-1-0
						8	0-0-0-1
						9	1-0-0-1
						10	0-1-0-1
						11	1-1-0-1
						12	0-0-1-1
						13	1-0-1-1
						14	0-1-1-1
						15	1-1-1-1
t.YPE	r/w	Base 1dP 2dP 3dP	6157 14349 22541 30733	45082	Enum	Enum_SegTyp	Segment type of segment 12. The segment type defines the setpoint behavior for this segment. The setpoint can be held constant or be changed with a ramp or a step function. Continuation to next segment is automatic or manual (define a hold time).
						0	time to setpoint
						1	rate to setpoint
						2	The final setpoint of the previous segment is kept constant for the duration 'Pt'.
						3	step to setpoint
						4	time to setpoint and wait
						5	rate to setpoint and wait
						6	The final setpoint of the previous segment is kept constant for the duration 'Pt'. At the end of a segment, the programmer enters the Stop mode (Run LED is off), and can be restarted by pressing the Start/Stop key (more than 3 s), via the interface, or a digital input.
						7	step to setpoint and wait
						8	The last segment in a program is the end segment. When the end segment has been reached, the last setpoint is maintained.
						9	timer
						10	timer and hold period.
SP	r/w	Base 1dP 2dP 3dP	6124 14316 22508 30700	45016	Float	-1999...9999 <input type="checkbox"/>	End setpoint of segment 12. This is the target setpoint that is reached at the end of the twelfth 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	r/w	Base 1dP 2dP 3dP	6125 14317 22509 30701	45018	Float	0...9999 <input type="checkbox"/>	Segment time/gradient 12. The duration of a segment can be defined directly, or by using the segment time and the setpoint difference (SP – segment starting setpoint). Whether the setting is for segment time or the gradient, is defined by means of the segment type parameter (tYPE).

Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
d.Out	r/w	Base 1dP 2dP 3dP	6158 14350 22542 30734	45084	Enum	Enum_Spuren	Control outputs 1...4 - 12. A program can control up to four digital signals: the control outputs 1...4. A combination of these signals can be assigned to every segment, whereby the signals are operated while the segment is running. For access to the controller's outputs, the signals must be assigned accordingly.
						0	0-0-0-0
						1	1-0-0-0
						2	0-1-0-0
						3	1-1-0-0
						4	0-0-1-0
						5	1-0-1-0
						6	0-1-1-0
						7	1-1-1-0
						8	0-0-0-1
						9	1-0-0-1
						10	0-1-0-1
						11	1-1-0-1
						12	0-0-1-1
						13	1-0-1-1
						14	0-1-1-1
						15	1-1-1-1
t.YPE	r/w	Base 1dP 2dP 3dP	6159 14351 22543 30735	45086	Enum	Enum_SegTyp	Segment type of segment 13. The segment type defines the setpoint behavior for this segment. The setpoint can be held constant or be changed with a ramp or a step function. Continuation to next segment is automatic or manual (define a hold time).
						0	time to setpoint
						1	rate to setpoint
						2	The final setpoint of the previous segment is kept constant for the duration 'Pt'.
						3	step to setpoint
						4	time to setpoint and wait
						5	rate to setpoint and wait
						6	The final setpoint of the previous segment is kept constant for the duration 'Pt'. At the end of a segment, the programmer enters the Stop mode (Run LED is off), and can be restarted by pressing the Start/Stop key (more than 3 s), via the interface, or a digital input.
						7	step to setpoint and wait
						8	The last segment in a program is the end segment. When the end segment has been reached, the last setpoint is maintained.
						9	timer
						10	timer and hold period.
SP	r/w	Base 1dP 2dP 3dP	6126 14318 22510 30702	45020	Float	-1999...9999 <input type="checkbox"/>	End setpoint of segment 13. This is the target setpoint that is reached at the end of the 13th 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	r/w	Base 1dP 2dP 3dP	6127 14319 22511 30703	45022	Float	0...9999 <input type="checkbox"/>	Segment time/gradient 13. The duration of a segment can be defined directly, or by using the segment time and the setpoint difference (SP – segment starting setpoint). Whether the setting is for segment time or the gradient, is defined by means of the segment type parameter (tYPE).

Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
d.Out	r/w	Base 1dP 2dP 3dP	6160 14352 22544 30736	45088	Enum	Enum_Spuren	Control outputs 1...4 - 13. A program can control up to four digital signals: the control outputs 1...4. A combination of these signals can be assigned to every segment, whereby the signals are operated while the segment is running. For access to the controller's outputs, the signals must be assigned accordingly.
						0	0-0-0-0
						1	1-0-0-0
						2	0-1-0-0
						3	1-1-0-0
						4	0-0-1-0
						5	1-0-1-0
						6	0-1-1-0
						7	1-1-1-0
						8	0-0-0-1
						9	1-0-0-1
						10	0-1-0-1
						11	1-1-0-1
						12	0-0-1-1
						13	1-0-1-1
						14	0-1-1-1
						15	1-1-1-1
t.YPE	r/w	Base 1dP 2dP 3dP	6161 14353 22545 30737	45090	Enum	Enum_SegTyp	Segment type of segment 14. The segment type defines the setpoint behavior for this segment. The setpoint can be held constant or be changed with a ramp or a step function. Continuation to next segment is automatic or manual (define a hold time).
						0	time to setpoint
						1	rate to setpoint
						2	The final setpoint of the previous segment is kept constant for the duration 'Pt'.
						3	step to setpoint
						4	time to setpoint and wait
						5	rate to setpoint and wait
						6	The final setpoint of the previous segment is kept constant for the duration 'Pt'. At the end of a segment, the programmer enters the Stop mode (Run LED is off), and can be restarted by pressing the Start/Stop key (more than 3 s), via the interface, or a digital input.
						7	step to setpoint and wait
						8	The last segment in a program is the end segment. When the end segment has been reached, the last setpoint is maintained.
						9	timer
						10	timer and hold period.
SP	r/w	Base 1dP 2dP 3dP	6128 14320 22512 30704	45024	Float	-1999...9999 <input type="checkbox"/>	End setpoint of segment 14. This is the target setpoint that is reached at the end of the 14th 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	r/w	Base 1dP 2dP 3dP	6129 14321 22513 30705	45026	Float	0...9999 <input type="checkbox"/>	Segment time/gradient 14. The duration of a segment can be defined directly, or by using the segment time and the setpoint difference (SP – segment starting setpoint). Whether the setting is for segment time or the gradient, is defined by means of the segment type parameter (tYPE).

Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
d.Out	r/w	Base 1dP 2dP 3dP	6162 14354 22546 30738	45092	Enum	Enum_Spuren	Control outputs 1...4 - 14. A program can control up to four digital signals: the control outputs 1...4. A combination of these signals can be assigned to every segment, whereby the signals are operated while the segment is running. For access to the controller's outputs, the signals must be assigned accordingly.
						0	0-0-0-0
						1	1-0-0-0
						2	0-1-0-0
						3	1-1-0-0
						4	0-0-1-0
						5	1-0-1-0
						6	0-1-1-0
						7	1-1-1-0
						8	0-0-0-1
						9	1-0-0-1
						10	0-1-0-1
						11	1-1-0-1
						12	0-0-1-1
						13	1-0-1-1
						14	0-1-1-1
						15	1-1-1-1
t.YPE	r/w	Base 1dP 2dP 3dP	6163 14355 22547 30739	45094	Enum	Enum_SegTyp	Segment type of segment 15. The segment type defines the setpoint behavior for this segment. The setpoint can be held constant or be changed with a ramp or a step function. Continuation to next segment is automatic or manual (define a hold time).
						0	time to setpoint
						1	rate to setpoint
						2	The final setpoint of the previous segment is kept constant for the duration 'Pt'.
						3	step to setpoint
						4	time to setpoint and wait
						5	rate to setpoint and wait
						6	The final setpoint of the previous segment is kept constant for the duration 'Pt'. At the end of a segment, the programmer enters the Stop mode (Run LED is off), and can be restarted by pressing the Start/Stop key (more than 3 s), via the interface, or a digital input.
						7	step to setpoint and wait
						8	The last segment in a program is the end segment. When the end segment has been reached, the last setpoint is maintained.
						9	timer
						10	timer and hold period.
SP	r/w	Base 1dP 2dP 3dP	6130 14322 22514 30706	45028	Float	-1999...9999 <input type="checkbox"/>	End setpoint of segment 15. This is the target setpoint that is reached at the end of the 15th 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	r/w	Base 1dP 2dP 3dP	6131 14323 22515 30707	45030	Float	0...9999 <input type="checkbox"/>	Segment time/gradient 15. The duration of a segment can be defined directly, or by using the segment time and the setpoint difference (SP – segment starting setpoint). Whether the setting is for segment time or the gradient, is defined by means of the segment type parameter (tYPE).

Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
d.Out	r/w	Base 1dP 2dP 3dP	6164 14356 22548 30740	45096	Enum	Enum_Spuren	Control outputs 1...4 - 15. A program can control up to four digital signals: the control outputs 1...4. A combination of these signals can be assigned to every segment, whereby the signals are operated while the segment is running. For access to the controller's outputs, the signals must be assigned accordingly.
						0	0-0-0-0
						1	1-0-0-0
						2	0-1-0-0
						3	1-1-0-0
						4	0-0-1-0
						5	1-0-1-0
						6	0-1-1-0
						7	1-1-1-0
						8	0-0-0-1
						9	1-0-0-1
						10	0-1-0-1
						11	1-1-0-1
						12	0-0-1-1
						13	1-0-1-1
						14	0-1-1-1
						15	1-1-1-1
t.YPE	r/w	Base 1dP 2dP 3dP	6165 14357 22549 30741	45098	Enum	Enum_SegTyp	Segment type of segment 16. The segment type defines the setpoint behavior for this segment. The setpoint can be held constant or be changed with a ramp or a step function. Continuation to next segment is automatic or manual (define a hold time).
						0	time to setpoint
						1	rate to setpoint
						2	The final setpoint of the previous segment is kept constant for the duration 'Pt'.
						3	step to setpoint
						4	time to setpoint and wait
						5	rate to setpoint and wait
						6	The final setpoint of the previous segment is kept constant for the duration 'Pt'. At the end of a segment, the programmer enters the Stop mode (Run LED is off), and can be restarted by pressing the Start/Stop key (more than 3 s), via the interface, or a digital input.
						7	step to setpoint and wait
						8	The last segment in a program is the end segment. When the end segment has been reached, the last setpoint is maintained.
						9	timer
						10	timer and hold period.
SP	r/w	Base 1dP 2dP 3dP	6132 14324 22516 30708	45031	Float	-1999...9999 <input type="checkbox"/>	End setpoint of segment 16. This is the target setpoint that is reached at the end of the 16th 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	r/w	Base 1dP 2dP 3dP	6133 14325 22517 30709	45034	Float	0...9999 <input type="checkbox"/>	Segment time/gradient 16. The duration of a segment can be defined directly, or by using the segment time and the setpoint difference (SP – segment starting setpoint). Whether the setting is for segment time or the gradient, is defined by means of the segment type parameter (YPE).

Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
d.Out	r/w	Base 1dP 2dP 3dP	6166 14358 22550 30742	45100	Enum	Enum_Spuren	Control outputs 1...4 - 16. A program can control up to four digital signals: the control outputs 1...4. A combination of these signals can be assigned to every segment, whereby the signals are operated while the segment is running. For access to the controller's outputs, the signals must be assigned accordingly.
						0	0-0-0-0
						1	1-0-0-0
						2	0-1-0-0
						3	1-1-0-0
						4	0-0-1-0
						5	1-0-1-0
						6	0-1-1-0
						7	1-1-1-0
						8	0-0-0-1
						9	1-0-0-1
						10	0-1-0-1
						11	1-1-0-1
						12	0-0-1-1
						13	1-0-1-1
						14	0-1-1-1
						15	1-1-1-1

Signal							
Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
St.Prog	r	Base 1dP 2dP 3dP	6050 14242 22434 30626	44868	Enum	0...255 <input type="checkbox"/>	Segment type of segment 16. The segment type defines the setpoint behavior for this segment. The setpoint can be held constant or be changed with a ramp or a step function. Continuation to next segment is automatic or manual (define a hold time).
						Bit 0,1,2 0: 1: 2:	Type of segment rising falling hold (dwell)
						Bit 3 Bit 4 Bit 5 Bit 6 Bit 7 Bit 8	Program 'Run' Program 'End' Program 'Reset' Program 'StartFlankMissing' Program 'BandHold + FailHold' Program active
SP.Pr	r	Base 1dP 2dP 3dP	6051 14253 22435 30627	44870	Float	-1999...9999 <input type="checkbox"/>	The programmer's setpoint is displayed as the effective setpoint while the program is running.
T1.Pr.	r	Base 1dP 2dP 3dP	6052 14244 22436 30627	44870	Float	0...9999 <input type="checkbox"/>	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	6053 14245 22437 30629	44874	Float	0...9999 <input type="checkbox"/>	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	6054 14246 22438 30630	44876	Float	0...9999 <input type="checkbox"/>	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.

Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
T4.Pr	r	Base 1dP 2dP 3dP	6055 14247 22439 30631	44878	Enum	0...9999 <input type="checkbox"/>	Only with running program. The remaining time of the running program segment (without hold times).
SG.Pr	r	Base 1dP 2dP 3dP	6056 14248 22440 30632	44880	Int	0...16 <input type="checkbox"/>	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.
Pr.SG	r/w	Base 1dP 2dP 3dP	6060 14252 22444 30636	44888	Int	0...16 <input checked="" type="checkbox"/>	Segment number for Preset. Preset involves starting the selected program with a different segment than the normal (1st) start segment. The starting setpoint of the preset segment becomes effective immediately, i.e. the program is not started. To use the Preset function, the programmer must be in the Stop or Reset state.
Pr.EF	r	Base 1dP 2dP 3dP	6057 14249 22441 30633	44882	Int	0...16 <input type="checkbox"/>	Number of the active program. The program remains active until a reset or a new start is triggered.
SP.En	r	Base 1dP 2dP 3dP	6058 14250 22442 30634	44884	Float	-1999...9999 <input type="checkbox"/>	The segment end set-point of the active segment is displayed. If programmer is in reset, the internal set-point is displayed.

## 23 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 <input type="checkbox"/>	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 <input type="checkbox"/>	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 <input type="checkbox"/>	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	3013 11295 19487 27679	38974	Float	0.1...9999 <input checked="" type="checkbox"/>	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	3014 11296 19488 27680	38976	Float	0...9999 <input type="checkbox"/>	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.
SP.bo	r/w	Base 1dP 2dP 3dP	3105 11297 19490 27682	38978	Float	0...9999 <input type="checkbox"/>	Boost increase. Increases the setpoint SP for the duration t.bo by the amount SP.bo. The boost function causes a brief setpoint increase, which is used e.g. to clear blocked channels ('frozen' material) in a hot-runner system.
t.bo	r/w	Base 1dP 2dP 3dP	3106 11298 19491 27683	38980	Float	0...9999 <input type="checkbox"/>	Duration of the boost increase in minutes. When the boost time t.bo has elapsed, the controller switches back to the standard setpoint SP. The boost function causes a brief setpoint increase, which is used e.g. to clear blocked channels ('frozen' material) in a hot-runner system.

Name	r/w	Adr.	Integer	Real	Typ	Value/Off	Description
Y.St	r/w	Base 1dP 2dP 3dP	5023 13215 21407 29599	42814	Float	-120...120 <input type="checkbox"/>	Reduced output value for start-up [%]. The start-up function is a protective function, e.g. with hot runner control. To prevent destruction of high-performance heating elements, they must be heated slowly to remove any humidity. With activated start-up function, the controller maintains the reduced starting temperature for a defined dwell period. Subsequently, the controller switches over to the main setpoint.
SP.St	r/w	Base 1dP 2dP 3dP	3107 11299 19491 27683	38982	Float	-1999...9999 <input type="checkbox"/>	Setpoint for start-up function. The start-up function is a protective function, e.g. with hot runner control. To prevent destruction of high-performance heating elements, they must be heated slowly to remove any humidity. With activated start-up function, the controller maintains the reduced starting temperature for a defined dwell period. Subsequently, the controller switches over to the main setpoint.
t.St	r/w	Base 1dP 2dP 3dP	3108 11300 19492 27684	38984	Float	0...9999 <input type="checkbox"/>	Start-up dwell period [min]. The start-up function is a protective function, e.g. with hot runner control. To prevent destruction of high-performance heating elements, they must be heated slowly to remove any humidity. With activated start-up function, the controller maintains the reduced starting temperature for a defined dwell period. Subsequently, the controller switches over to the main setpoint.

- | Signal |     |                           |                                 |       |       |                                       |   |
|--------|-----|---------------------------|---------------------------------|-------|-------|---------------------------------------|---|
| Name   | r/w | Adr.                      | Integer                         | Real  | Typ   | Value/Off                             | Description   |
| SP.EF  | r   | Base<br>1dP<br>2dP<br>3dP | 3170<br>11362<br>19554<br>27746 | 39108 | Float | -1999...9999 <input type="checkbox"/> | 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. |
| Diff   | r   | Base<br>1dP<br>2dP<br>3dP | 3171<br>11363<br>19555<br>27747 | 39110 | Float | -1999...9999 <input type="checkbox"/> | Difference between the effective setpoint and setpoint 2  |
| SP     | r/w | Base<br>1dP<br>2dP<br>3dP | 3180<br>11372<br>19564<br>27756 | 39128 | Float | -1999...9999 <input type="checkbox"/> | 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<br>1dP<br>2dP<br>3dP | 3181<br>11373<br>19565<br>27757 | 39130 | Float | -1999...9999 <input type="checkbox"/> | 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<br>1dP<br>2dP<br>3dP | 3182<br>11374<br>19566<br>27758 | 39132 | Float | 0...9999 <input type="checkbox"/>     | Current timer count in minutes. Count-down timer. The run time is only visible, if the timer is active.<br>Configuration in the extended Operating Level.   |

## 24 Tool

- | ConF   |     |                           |                               |       |      |           |  |
|--------|-----|---------------------------|-------------------------------|-------|------|-----------|--|
| Name   | r/w | Adr.                      | Integer                       | Real  | Typ  | Value/Off | Description  |
| U.LinT | r/w | Base<br>1dP<br>2dP<br>3dP | 634<br>8826<br>17018<br>25210 | 34036 | Enum | Enum_Unit | Engineering unit of linearization table (temperature). |
|        |     |                           |                               |       |      | 0         | Without unit   |
|        |     |                           |                               |       |      | 1         | °C   |
|        |     |                           |                               |       |      | 2         | °F   |

**Limited Warranty:**

Please refer to the Chromalox limited warranty applicable to this product at <http://www.chromalox.com/customer-service/policies/termsofsale.aspx>.

**Chromalox, Inc.  
1347 Heil Quaker Boulevard  
Lavergne, TN 37086  
(615) 793-3900  
[www.chromalox.com](http://www.chromalox.com)**