

**ELOTECH**

INDUSTRIELELEKTRONIK GMBH

## Manual: Data Transfer

**Profibus DP, EN 50170**



**Multizones Temperature Controller**

**Series: R2000, R2000B, R2100, R2200**

### Content:

Series: R2000, R2000B, R2100, R2200.....	1
1. Disclaimer of liability .....	2
2. Interface, general.....	2
2.1 Line routing, screening and measures to combat interference voltage .....	3
2.2 Shielding of lines .....	4
2.3 Connection guide .....	5
2.3.1 PROFIBUS-Connections: .....	5
2.3.2 PROFIBUS – Adjustments:.....	5
2.3.3 PROFIBUS - Diagnostic displays:.....	6
3. Data Transfer, general.....	6
3.1 The Communication:.....	6
3.2 Process reflection.....	7
3.2.1 From master to slave:.....	7
3.2.2 From slave to master:.....	8
3.2.3 Transmission example.....	9
3.3 Configuration channel .....	11
3.3.1 Data transmission, general .....	11
3.3.2 Terms.....	11
3.3.3 Parameter ranges .....	11
3.3.4 Configuration of the parameters via the configuration channel. ....	11
3.3.5 Parameter list.....	13
3.3.6 Transmission examples .....	15
<b>3.3.6.1 Configuration channel, Instruction code: 10 H .....</b>	<b>15</b>
<b>3.3.6.2 Configuration channel, Instruction code: 20 H .....</b>	<b>16</b>
<b>3.3.6.3 Configuration channel, Instruction code: 21 H .....</b>	<b>17</b>
3.4 Process reflection and Configuration channel.....	18

## 1. Disclaimer of liability

We have checked the contents of the document for conformity with the hardware and software described. Nevertheless, we are unable to preclude the possibility of deviations so that we are unable to assume warranty for full compliance. The information given in the publication is reviewed regularly. Necessary amendments are incorporated in the following editions.

We would be pleased to receive any improvement proposals which you may have.

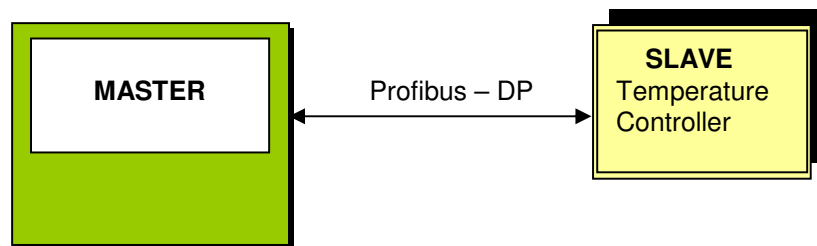
This document may not be passed on nor duplicated, nor may its contents be used or disclosed unless expressly permitted.

**Note: Only in PROFIBUS-technology trained personnel following the safety regulations may do the PROFIBUS-connections.**

**It is essential, that one has well experience in installing a Profibus device.**

## 2. Interface, general

The ELOTECH – multizones temperature controller is equipped with a PROFIBUS DP interface.



The PROFIBUS -interface allows the slave to be monitored and controlled by a PROFIBUS master. The data transfer between the slave and master takes place with the aid of the PROFIBUS-DP-protocol acc. to EN 50170.

The communication is always controlled by the PROFIBUS-DP master.

The temperature controller actuates as a slave.

Each slave has its own device address.

If there are transmission or other errors detected by the slave, it doesn't accept this data.

The old parameter values are still valid.

**Please take attention to the manual of the specific temperature controller.**

### GDS - data file:

Will be delivered with the slave on CD or download: [www.elotech.de](http://www.elotech.de)

### FAQ's:

see [www.elotech.de](http://www.elotech.de) (downloads)

## 2.1 Line routing, screening and measures to combat interference voltage

This chapter deals with line routing in the case of bus, signal and power supply lines, with the aim of ensuring an EMC-compliant design of your system.

### General information on line routing

- Inside and outside of cabinets

In order to achieve EMC-compliant routing of the lines, it is advisable to split the lines into the following line groups and to lay these groups separately.

Group A:      •shielded bus and data lines (e.g. for PROFIBUS-DP, RS232C and printers etc.)  
                  •shielded analogue lines  
                  •unshielded lines for DC voltages  $\geq 60$  V  
                  •unshielded lines for AC voltage  $\geq 25$  V  
                  •coaxial lines for monitors

Group B:      •unshielded lines for DC voltages  $\geq 60$  V and  $\geq 400$  V  
                  •unshielded lines for AC voltage  $\geq 24$  V and  $\geq 400$  V

Group C:      •unshielded lines for DC voltages  $> 400$  V

The table below allows you to read off the conditions for laying the line groups on the basis of the combination of the individual groups.

Line laying instructions as a function of the combination of line groups:

	<b>Group A</b>	<b>Group B</b>	<b>Group C</b>
Group A	1	2	3
Group B	2	1	3
Group C	3	3	1

- 1) Lines may be laid in common bunches or cable ducts.
- 2) Lines must be laid in separate bunches or cable ducts (without minimum clearance).
- 3) Lines must be laid in separate bunches or cable ducts inside cabinets but on separate cable racks with at least 10 cm clearance outside of cabinets but inside buildings .

## 2.2 Shielding of lines

Shielding is intended to weaken (attenuate) magnetic, electrical or electromagnetic interference fields.

Interference currents on cable shields are discharged to earth via the shielding bus which is connected conductively to the chassis or housing. A low-impedance connection to the PE wire is particularly important in order to prevent these interference currents themselves becoming an interference source.

Wherever possible, use only lines with braided shield. The coverage density of the shield should exceed 80 %. Avoid lines with foil shield since the foil can be damaged very easily as the result of tensile and compressive stress on attachment. The consequence is a reduction in the shielding effect.

In general, you should always connect the shields of cables at both ends. The only way of achieving good interference suppression in the higher frequency band is by connecting the shields at both ends.

The shield may also be connected at one end only in exceptional cases. However, this then achieves only an attenuation of the lower frequencies. Connecting the shield at one end may be more favourable if

- it is not possible to lay an equipotential bonding line
- analogue signals (a few mV resp. mA) are to be transmitted
- foil shields (static shields) are used.

In the case of data lines for serial couplings, always use metallic or metallised plugs and connectors. Attach the shield of the data line to the plug or connector housing. Do not connect the shield on the connector of the slave (controller).

If there are potential differences between the earthing points, a compensating current may flow via the shield connected at both ends. In this case, you should lay an additional equipotential bonding line.

Please note the following points when shielding:

- Use metal cable clips to secure the shield braiding. The clips must surround the shield over a large area and must have good contact.
- Downstream of the entry point of the line into the cabinet, connect the shield to a shielding bus. Continue the shield as far as the module, but do not connect it again at this point!

## 2.3 Connection guide

**Note:** Only in PROFIBUS-technology trained personnel following the safety regulations may do the PROFIBUS - connections.

**It is essential, that one has well experience in installing a Profibus device.**

You will require the following components to connect the slave:

- Connector for Profibus connection to the slave
- PROFIBUS cable (this cable is generally already installed on site!)
- GSD file
- Project planning tool for the PROFIBUS-Master

**It is essential, that you perform the following during connecting in order to ensure that the slave operates correctly:**

### 2.3.1 PROFIBUS-Connections:

Connect the slave with the PROFIBUS. Take care to the terminals.

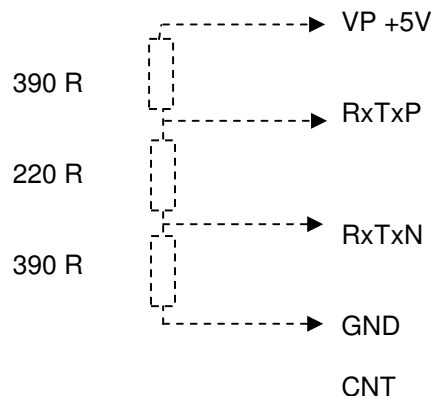
**See: connection diagram of the specific controller type.**

**The terminals VP and GND have to be used to connect the terminating-resistors.**

**There is no further load allowed.**

**The terminating resistors have to be connected on the first and the last device of the Profibus line.**

Terminating-Resistors (Tol. +/-2%):



### 2.3.2 PROFIBUS – Adjustments:

Adjust the following parameters (slave):

#### Series R2000, R2100:

Parameter „Address“:	Adr	Adjustment of the Profibus-Address.
Parameter „Baudrate“:	baud	No adjustment possible. The baudrate will detected and monitored automatically. Display: „ndEt“ = no baudrate detected.

#### Series R2200:

Parameter „Address“:	see DIP switches in front of the instrument.
Parameter „Baudrate“:	baud No adjustment possible. The baudrate will detected and monitored automatically.

### 2.3.3 PROFIBUS - Diagnostic displays:

Series R2000 and R2100: See the lightened decimal point in the left display of the zones indication display (Zone).

Series R2200: See the green "BUS-LED"

#### Function:

Dec.-point or LED are "on": Data exchange:  
The device is into the "data-exchange-modus".  
The communication is OK..  
The data exchange takes place.

Dec.-point or LED "flashing": Wait:  
The bus connection has been detected.  
The device is waiting to be configured or programmed by the master.

Dec.-point or LED are "off": No connection:  
Please check the wiring and the bus connections.  
Check if the master works OK..  
There is no communication.

## 3. Data Transfer, general

### 3.1 The Communication:

The master sends it's data to the slave.  
The data will be overtaken only if the instrument is programmed to "remote"-action.

After this the slave sends an answer to the PROFIBUS DP - master.  
This takes place cyclic and will be controlled by the master.  
The configuration of the slave takes place with the help of the GSD-file.

The following modules are available for the slave:

- |  |   |
|--|---|
| 1. Process reflection:                           | Module: „x – channel process data“  |
| 2. Configuration channel:                        | Module: „parameter channel“   |
| 3. Process reflection and Configuration channel: | Module: „x – channel process + parameter“<br>x = 2, 4, 6, 8, 10, 12 or 16 (zones) |

Example for 8-zones controllers: x = 8 channel module, also if only 1 or 6 zones are in action.

If the device is an 8-zones controller and if there are only 4 or 6 zones in operation, an 8-channel module has to be used.

## 3.2 Process reflection

Parameter transfer according to the process reflection module:

### 3.2.1 From master to slave: Transfer of Setpoint 1 and Status word 1

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	
Zone 1: <b>Setpoint 1</b> High Byte	Zone 1: <b>Setpoint 1</b> Low Byte	Zone 1: <b>Control byte</b>	Zone 2: <b>Setpoint 1</b> High Byte	Zone 2: <b>Setpoint 1</b> Low Byte	Zone 2: <b>Control -byte</b>	...

aso.

	Byte 46	Byte 47	Byte 48
...	Zone 16: <b>Setpoint 1</b> High Byte	Zone 16: <b>Setpoint 1</b> Low Byte	Zone 16: <b>Control -byte</b>

If there are instruments with controller zones lower than 16 (e.g. 8 or 12 zones) the number of the transmitted data bytes is lower too.

**Setpoint / process value:** The parameter value consists out of 2 data bytes within the process reflection.  
First the high- and then the low-byte will be transmitted.  
Setpoint and actual process values will be transmitted always with a decimal digit, although the measuring range has no decimal digit.

Example:	°C	Dec.	Hex.	High-Byte	Low-Byte
Measuring range with dec. point: act. value	23,0	230	00E6	00	E6
Measuring range with dec. point: Setpoint	170,0	1700	06A4	06	A4
Measuring range without dec. point: act. value	23	230	00E6	00	E6
Measuring range without dec. point: Setpoint	170	1700	06A4	06	A4

**Control byte:** The parameter consists out of one data byte:

- Bit 0: controller zone            0=on, 1=off
- Bit 1: self tuning                0=off, 1=on  
Changing this bit from „0“ to „1“ will force the controller to do one self tuning action.  
Set this bit to „0“, before starting a new self tuning action.
- Bit 2: 0
- Bit 3: actual setpoint            0=setpoint SP1, 1=setpoint SP2
- Bit 4: 1 = delete warning „self tuning error“ into the status byte
- Bit 5: 0
- Bit 6: 0
- Bit 7: 1 = delete warning „system error“ into the status byte

### 3.2.2 From slave to master: Transfer of the process data

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
<b>Status Setpoint</b> High Byte	<b>Status Setpoint</b> Low Byte	Zone 1 <b>Process value</b> High Byte	Zone 1 <b>Process value</b> Low Byte	Zone 1 <b>Controller Status</b>	Zone 1 <b>Alarm Status</b>

Byte 7	Byte 8	Byte 9	Byte 10	
Zone 2 <b>Process value</b> High Byte	Zone 2 <b>Process value</b> Low Byte	Zone 2 <b>Controller Status</b>	Zone 2 <b>Alarm Status</b>	. . .

	Byte 63	Byte 64	Byte 65	Byte 66
. . .	Zone 16 <b>Process value</b> High Byte	Zone 16 <b>Process value</b> Low Byte	Zone 16 <b>Controller Status</b>	Zone 16 <b>Alarm Status</b>

If there are instruments with controller zones lower than 16 (e.g. 8 or 12 zones) the number of the transmitted data bytes is lower too.

**Status setpoint:** Indicates, if a range error has been detected, when writing the setpoint:

Bit 0:           0 = Zone 1:    setpoint value OK.  
                  1 =            setpoint value not OK. (out of range ?)

Bit 1:           0 = Zone 2:    setpoint value OK.  
                  1 =            setpoint value not OK. (out of range ?)

Bit 2:           0 = Zone 3:    setpoint value OK.  
                  1 =            setpoint value not OK. (out of range ?)

Bit 15:          0 = Zone 16: setpoint value OK.  
                  1 =            setpoint value not OK. (out of range ?)

**Alarm Status:** Bit 0: 1 = alarm 1 active  
Bit 1: 1 = alarm 2 active  
Bit 2...: 0 (no function)

**Controller Status:** The parameter consists out of one data byte:

Bit 0:           controller zone 0=on, 1=off

Bit 1:           self tuning     0=off, 1=on

Bit 2:           remote action 0=on, 1=off, operation via keyboard

Bit 3:           actual setpoint 0=setpoint SP1, 1=setpoint SP2

Bit 4:           1 = self tuning error

Bit 5:           1 = setpoint ramp function active

Bit 6:           1 = sensor error

Bit 7:           1 = system error

### 3.2.3 Transmission example

#### From master to slave: transfer of setpoint 1 and control byte

Byte 1 + 2: Zone 1 The setpoint 50,0°C should be send to the slave.  
 Setpoint: 500 decimal = 0x01F4 hexadecimal as a 16 bit integer-value

Byte 3: Zone 1 The slave should be switched „on“ (Bit 0 = 1).

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	
Zone 1: <b>Setpoint 1</b> High Byte 0x01	Zone 1: <b>Setpoint 1</b> Low Byte 0xF4	Zone 1: <b>Control Byte</b> 0x01	Zone 2: <b>Setpoint 1</b> High Byte 0x..	Zone 2: <b>Setpoint 1</b> Low Byte 0x..	Zone 2: <b>Control Byte</b> 0x..	...

	Byte 46	Byte 47	Byte 48
...	Zone 16: <b>Setpoint 1</b> High Byte 0x..	Zone 16: <b>Setpoint 1</b> Low Byte 0x..	Zone 16: <b>Control Byte</b> 0x..

If there are instruments with controller zones lower than 16 (e.g. 8 or 12 zones) the number of the transmitted data bytes is lower too.

**Answer from slave to master:  
Transmission of the process reflection**

The slave sends the following parameter-values:

Byte 1 + 2: status instruction setpoint transmission: the last instruction was OK.

Byte 3 + 4: Zone 1 Act. process temp. value:55,0°C 550dec.=0x0226hex., 16 bit integer-value

Byte 5: Zone 1 Controller status: controller = on

Byte 6: Zone 1 Alarm status: alarm = no alarm

Byte 7 + 8: Zone 2 Act. process temp. value:56,0°C 560dec.=0x0230hex, 16 bit integer-value

Byte 9: Zone 2 Controller status: controller = on

Byte 10: Zone 2 Alarm status: alarm = no alarm

.

.

.

Byte63 + 64: Zone 16 Act. process temp. value

Byte 65: Zone 16 Controller status

Byte 66: Zone 16 Alarm status

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
<b>Status Setpoint</b> High Byte 0x00	<b>Status Setpoint</b> Low Byte 0x00	Zone 1 <b>Process Value</b> High Byte 0x02	Zone 1 <b>Process Value</b> Low Byte 0x26	Zone 1 <b>Controller Status</b> 0x01	Zone 1 <b>Alarm Status</b> 0x00

Byte 7	Byte 8	Byte 9	Byte 10	
Zone 2 <b>Process Value</b> High Byte 0x02	Zone 2 <b>Process Value</b> Low Byte 0x3A	Zone 2 <b>Controller Status</b> 0x01	Zone 2 <b>Alarm Status</b> 0x02	. . .

	Byte 63	Byte 64	Byte 65	Byte 66
. . .	Zone 16 <b>Process Value</b> High Byte 0x..	Zone 16 <b>Process Value</b> Low Byte 0x..	Zone 16 <b>Controller Status</b> 0x..	Zone 16 <b>Alarm Status</b> 0x..

If there are instruments with controller zones lower than 16 (e.g. 8 or 12 zones) the number of the transmitted data bytes is lower too.

### 3.3 Configuration channel

With the help of the configuration channel each parameter can be addressed individually.

The sequence of the described bytes is valid for „question“ and „answer“.

#### 3.3.1 Data transmission, general

The PROFIBUS – master is allowed to monitor and control all parameters of the slave.

The transfer of instructions and parameter values takes place with the aid defined data blocks.

#### 3.3.2 Terms

Instruction-code	<b>[BC]:</b>	"tells" the device/slave, what to do	(1 byte)
Parameter-code	<b>[PC]:</b>	designates each individual parameter of the device	(1 byte)
Parameter-value	<b>[PW]:</b>	shows the value of a parameter	(3 bytes)

#### 3.3.3 Parameter ranges

Instruction-code	<b>[BC]:</b>	0x10, 0x20, 0x21
Parameter-code	<b>[PC]:</b>	0x00...0xFF
Parameter-value	<b>[PW]:</b>	16 bit integer, mantissa <b>PWH</b> and <b>PWL</b> and exponent <b>PWE</b> base 10
Parameter-value High-Byte	<b>[PWH]</b>	
Parameter-value Low- Byte	<b>[PWL]</b>	
Parameter-exponent	<b>[PWE]</b>	

#### 3.3.4 Configuration of the parameters via the configuration channel.

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
<b>Current number</b> 0x00 ... 0xFF	<b>Number of Zone</b> max. 16 0x01...0xFF	<b>Instruction code</b> <b>BC</b> 0x10, 0x20 or 0x21	<b>Always:</b> 0x00	<b>Parameter-code</b> <b>PC</b> 0x00 ... 0xFF	<b>Parameter-value</b> <b>PWH</b> High-Byte	<b>Parameter-value</b> <b>PWL</b> Low-Byte	<b>Exponent</b> <b>PWE</b> 0x00 ... 0xFF

#### Byte 1

**Current Number:** The master should pre-set a current number before every new task. This number will be repeated from the slave with every answer. So it is possible to find out, which instruction and which answer belong together.

**Byte 2:** Number of controller zone of the addressed instrument (slave).  
All parameters, which are necessary to configure the instrument, will be transmitted and addressed via zone no. 1.

**Byte 3**

**Instruction code, BC:** 0x10 : Read parameter  
0x20 : Write parameter  
0x21 : Write parameter and store with power fail protection  
Take care: The EAROM or EEPROM of the slave permits max. 1.000.000 write cycles.

**Byte 4:** always 0x00

**Byte 5**

**Parameter code, PC: Enquiry:** Addresses the parameter which should be configured.  
**Answer:**  
If the read-proceeding to the slave was OK., than, in the answer of the slave, byte 5 shows the parameter-code PC.  
If the write-proceeding to the slave was OK., than, in the answer of the slave, byte 5 shows the value 00H (acknowledge).  
If the communication was not OK., the following error-warnings are shown in byte 5:  
03 H - Procedure error (instruction code not valid)  
04 H - Non-compliance with specified range (value to low or to high)  
05 H - Byte 2 ≠ 0  
06 H - The addressed parameter is a read-only parameter  
07 H - Writing of data not possible. Slave status is not „remote“.  
08 H - Parameter-code not valid  
09 H - It is not possible, to execute the instruction (e.g., the auto tuning can't be started)  
FEH - Error during writing into the power fail storage  
FFH - General error

**Bytes 6, 7 and 8**

**Parameter value:** The parameter value comprises three data bytes:  
2 data byte (mantissa), 1 data byte (exponent).

Byte 6: Parameter value **PWH**  
Byte 7: Parameter value **PWL**  
Byte 8: Parameter value **PWE**

Examples:	Dec.	Hex.	Mantissa	Dec. point
Process value (°C):	215	00D7	00D7	00
Setpoint (°C):		230	00E6	00E6 00
Output ratio, cooling (%)	-16	FFF0	FFF0	00
Setpoint ramp (°C/min.):	2,2	0016	0016	01

The parameter value is calculated as follows: Dec.: 2,2 = 22 + 1 dec. point  
Hex.: = 0016 (mantissa)  
= 01 (1 dec. point)

Negative mantissa / negative Exponent: Built binary two's complement.

### 3.3.5 Parameter list

Parameter	Mnemonics	Parameter-Code	R2000	R2000B	R2100	R2200
<b>Actual process values:</b>						
Act. temperature value		0x10	RO	RO	RO	RO
Act. heater current value	cur	0x11	RO	-	RO	RO
Act. leakage current value	c.	0x12	RO	-	RO	RO
Temperature offset value	OFSt	0x18	RW	RW	RW	RW
Sensor configuration	SEn	0x1a	RW	RW	RW	RW
Measuring range, dec. point	r. dP	0x1d	RO	-	RO	RO
Linear input; decimal points	r. dP	0x1d	RW	-	RW	RW
Linear input; bottom end value	r. Lo	0x1e	RW	-	RW	RW
Linear input; top end value	r. Hi	0x1f	RW	-	RW	RW
<b>Setpoints:</b>						
Actual setpoint	SP, act.	0x20	RO	RO	RO	RO
Setpoint 1	SP1	0x21	RW	RW	RW	RW
Setpoint 2	SP2	0x22	RW	RW	RW	RW
Setpoint limitation, low range	SP.Lo	0x2b	RW	RW	RW	RW
Setpoint limitation, high range	SP.Hi	0x2c	RW	RW	RW	RW
Setpoint ramp, rising	SP ↑	0x2f	RW	RW	RW	RW
Setpoint range, falling	SP ↓	0x2d	RW	RW	RW	RW
<b>Alarms:</b>						
Heater current, detect. interval	Cu.CY	0x31	RW	-	RW	RW
Min. leakage current value	Cu.SP	0x32	RW	-	RW	RW
Alarm 1, Configuration	Co.A1	0x34	RW	RW	RW	RW
Alarm 2, Configuration	Co.A2	0x35	RW	RW	RW	RW
Alarm value 1	A1	0x38	RW	RW	RW	RW
Alarm value 2	A2	0x39	RW	RW	RW	RW
Switching behaviour A1	rE.A1	0x3c	RW	RW	RW	RW
Switching behaviour A2	rE.A2	0x3d	RW	RW	RW	RW
Delay time A1	dL.A1	0x3e	RW	-	RW	RW
Delay time A2	dL.A2	0x3f	RW	-	RW	RW
<b>PID parameters „heating“:</b>						
Proportional range (P)	1 P	0x40	RW	RW	RW	RW
Rate time (D)	1 d	0x41	RW	RW	RW	RW
Reset time (I)	1 I	0x42	RW	RW	RW	RW
Cycle time	1 C	0x43	RW	RW	RW	RW
Control sensitivity	1 Sd	0x47	RW	RW	RW	RW
Dead band / switch-point difference (only for 3-point-controller)	SH	0x46	RW	RW	RW	RW
<b>PID parameters „cooling“:</b>						
Proportional range (P)	2 P	0x50	RW	RW	RW	RW
Rate time (D)	2 d	0x51	RW	RW	RW	RW
Reset time (I)	2 I	0x52	RW	RW	RW	RW
Cycle time	2 C	0x53	RW	RW	RW	RW
Control sensitivity	2 Sd	0x57	RW	RW	RW	RW

Parameter	Mnemonics	Parameter-Code	R2000	R2000B	R2100	R2200
<b>Output ratio:</b>						
Actual output ratio	Y	0x60	RO	RO	RO	RO
Manual output ratio	HAnd	0x62	RW	RW	RW	RW
Output ratio limit (heating)	1LY	0x64	RW	RW	RW	RW
Output ratio limit (cooling)	2LY	0x69	RW	RW	RW	RW
Soft start output ratio	So. Y	0x6a	RW	RW	RW	RW
Soft start setpoint	So.Sp	0x6b	RW	RW	RW	RW
Soft start duration time	So.ti	0x6c	RW	RW	RW	RW
Soft start function on/off	So.St	0x6d	RW	RW	RW	RW
<b>Status words:</b>						
Controller status / status byte		0x78	RW	RW	RW	RW
<b>Controller configuration:</b>						
Control action: heat-only heating-off-cooling	ConF	0x80	RW	RW	RW	RW
Adjustment lock	LOC	0x85	RW	RW	RW	-
Configuration of key "F1"	Co.F1	0x86	RW	RW	RW	-
Self tuning	OPt	0x88 0 = off 1 = on	RW	RW	RW	RW
Zone offset	Zo.OF	0x89	RW	RW	RW	RW
Manual output ratio: configuration	Hand	0x8b 0 = control action 1 = auto 2 = manual operation	RW	RW	RW	RW
Unit of measuring zone	Unit	0x8d	RW	-	RW	-
Sensor configuration	P tc	0x8e	RW	RW	RW	RW
Controller zone off/on	ZonE	0x8f 0 = off 1 = on	RW	RW	RW	RW

### 3.3.6 Transmission examples

#### 3.3.6.1 Configuration channel, Instruction code: 10 H

The slave is asked to send the parameter „Process value, 10 H“ of zone no.1 to the master.  
The process value is 225 °C. 225 (Decimal) = 0xE1 (Hex)

<b>Master to slave:</b>	<b>Dec.</b>	<b>Hex</b>
Current number:	1	0x01
Zone no.:	1	0x01
Send parameter:	16	0x10
Always:	0	0x00
Parameter code (process value):	16	0x10
Parameter value (High-Byte):	0	0x00
Parameter value (Low -Byte):	0	0x00
Exponent:	0	0x00

Transmission to slave: 0x01, 0x01 0x10, 0x00, 0x10, 0x00, 0x00, 0x00

<b>Slave to master:</b>	<b>Dec.</b>	<b>Hex</b>
Current number of instruction:	1	0x01
Zone no.:	1	0x01
Send parameter:	16	0x10
Always:	0	0x00
Parameter code (process value):	16 *)	0x10
Parameter value (High-Byte):	0	0x00
Parameter value (Low -Byte):	225	0xE1
Exponent: 10°	0	0x00

Transmission to master: 0x01, 0x01 0x10, 0x00, 0x10, 0x00, 0xE1, 0x00

\*) Repetition of the parameter code (PC = 16), because the read-process was OK.

### 3.3.6.2 Configuration channel, Instruction code: 20 H

The slave gets the instruction:

"Overtake parameter „prop.-band heating“ (parameter code: 40H, parameter value: 5,0 %) of zone no. 2 and store into the RAM".

<b>Master to slave:</b>	<b>Dec.</b>	<b>Hex</b>
Current number:	2	0x02
Zone no.:	2	0x02
Instruction code:	32	0x20
Always:	0	0x00
Parameter code:	64	0x40
Parameter value (High-Byte):	0	0x00
Parameter value (Low -Byte):	50	0x32
Dec.-point	1	0x01

Transmission to slave: 0x02, 0x02, 0x20, 0x00, 0x40, 0x00, 0x32, 0x01

<b>Slave to master:</b>	<b>Dec.</b>	<b>Hex</b>
Current number of instruction:	2	0x02
Zone no.:	2	0x02
Instruction code:	32	0x20
Always:	0	0x00
Parameter code (Prop-band, heating): 0 *)	0	0x00
Parameter value (High-Byte):	0	0x00
Parameter value (Low -Byte):	0	0x00
Dec.-point	0	0x00

Transmission to master: 0x02, 0x02, 0x20, 0x00, 0x00, 0x00, 0x00, 0x00

- \*) If the slave has understood the instruction of the master, it answers always with the parameter code (PC) = 00, because the writing-process was OK.  
If there are transmission or other errors the slave answers with the corresponding error code.

### 3.3.6.3 Configuration channel, Instruction code: 21 H

The slave gets the instruction:

"Overtake parameter setpoint 1 / SP1 = 200 °C (parameter code: 21H) of zone no. 1 and store power fail safe into the EEPROM".

<b>Master to slave:</b>	<b>Dec.</b>	<b>Hex</b>
Current number:	3	0x03
Zone no.:	1	0x01
Instruction code:	33	0x21
Always:	0	0x00
Parameter code (SP1):	33	0x21
Parameter value (High-Byte):	0	0x00
Parameter value (Low -Byte):	200	0xC8
Exponent: $10^0$	0	0x00

Transmission to slave: 0x03, 0x01, 0x21, 0x00, 0x21, 0x00, 0xC8, 0x00

<b>Slave to master:</b>	<b>Dec.</b>	<b>Hex</b>
Current number of instruction:	3	0x03
Zone no.:	1	0x01
Instruction code:	33	0x21
Always:	0	0x00
Parameter code:	0 *)	0x00
Parameter value (High-Byte):	0	0x00
Parameter value (Low -Byte):	0	0x00
Exponent: $10^0$	0	0x00

Transmission to master: 0x03, 0x01, 0x21, 0x00, 0x00, 0x00, 0x00, 0x00

- \*) If the slave has understood the instruction of the master, it answers always with the parameter code (PC) = 00, because the writing-process was OK.  
If there are transmission or other errors the slave answers with the corresponding error code.

### 3.4 Process reflection and Configuration channel

It is possible, to transmit process reflection and configuration channel simultaneously  
 In this case the bytes of the configuration channel have to be fit together with the process reflection.

**Master to slave:**

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	
Zone 1: <b>Setpoint 1</b> High Byte	Zone 1: <b>Setpoint 1</b> Low Byte	Zone 1: <b>Control Byte</b>	Zone 2: <b>Setpoint 1</b> High Byte	Zone 2: <b>Setpoint 1</b> Low Byte	Zone 2: <b>Control Byte</b>	...

	Byte 46	Byte 47	Byte 48
...	Zone 16: <b>Setpoint 1</b> High Byte	Zone 16: <b>Setpoint 1</b> Low Byte	Zone 16: <b>Control Byte</b>

Byte 49	Byte 50	Byte 51	Byte 52	Byte 53	Byte 54	Byte 55	Byte 56
<b>Current Number</b>	<b>Controller Zone</b>	<b>Instruction Code</b>  <b>BC</b>	<b>Always:</b>  0x00	<b>Parameter Code</b> <b>PC</b>	<b>Parameter Value</b> <b>PWH</b> High Byte	<b>Parameter Value</b> <b>PWL</b> Low Byte	<b>Dec. Point</b>  <b>PWK</b>

If there are instruments with controller zones lower than 16 (e.g. 8 or 12 zones) the number of the transmitted data bytes is lower too.

**Slave to Master:**

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
<b>Status Setpoint</b> High Byte	<b>Status Setpoint</b> Low Byte	Zone 1 <b>Process Value</b> High Byte	Zone 1 <b>Process Value</b> Low Byte	Zone 1 <b>Controller Status</b>	Zone 1 <b>Alarm Status</b>

Byte 7	Byte 8	Byte 9	Byte 10
Zone 2 <b>Process Value</b> High Byte	Zone 2 <b>Process Value</b> Low Byte	Zone 2 <b>Controller Status</b>	Zone 2 <b>Alarm Status</b>

Byte 11	Byte 12	Byte 13	Byte 14	
Zone 3 <b>Process Value</b> High Byte	Zone 3 <b>Process Value</b> Low Byte	Zone 3 <b>Controller Status</b>	Zone 3 <b>Alarm Status</b>	. . .

	Byte 63	Byte 64	Byte 65	Byte 66
. . .	Zone 16 <b>Process Value</b> High Byte	Zone 16 <b>Process Value</b> Low Byte	Zone 16 <b>Controller Status</b>	Zone 16 <b>Alarm Status</b>

Byte 67	Byte 68	Byte 69	Byte 70	Byte 71	Byte 72	Byte 73	Byte 74
<b>Current Number</b>	<b>Controller Zone</b>	<b>Instruction Code</b>  <b>BC</b>	<b>Always:</b>  0x00	<b>Parameter Code</b> <b>PC</b>	<b>Parameter Value</b> <b>PWH</b> High Byte	<b>Parameter Value</b> <b>PWL</b> Low Byte	<b>Dec. Point</b>  <b>PWK</b>

---

**ELOTECH** Industrieelektronik GmbH  
 Verbindungsstr. 27  
 D - 40723 HILDEN  
 Tel.: 0049 / 2103 / 23055      Fax: 0049 / 2103 / 23057  
 www.elotech.de