ELOTECH

Manual:
Data Transfer
Profibus DP, EN 50170

Temperature Controller
Type: R1140

Content:

ELOTECH Industrieelektronik GmbH
Verbindungsstrasse 27
D – 40723 HILDEN
FON +49 2103 / 255 97 0 FAX +49 2103 / 255 97 29
www.elotech.de Email: info@elotech.de

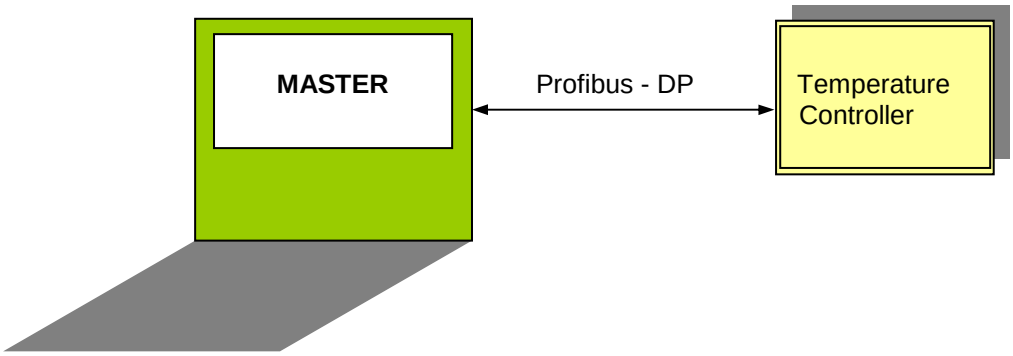
1. Disclaimer of liability

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, however, 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-technologie 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 – temperature controller is equipped with a PROFIBUS DP interface.
The temperature controller actuates as a slave.



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 adress of the slave has to be programmed in the configuration level of the slave.

If there are transmission or other errors detected by the slave, it doesn`t accept this datas.
The old parameter values are still valid.

Slave adjustments:

Unit adress (Adr): The adress of the slave 1...255 has to be adjusted into the configuration level. See parameter „Adr“.

Baudrate bAUd): 9,6 kBaud ... 12 MBaud (with automatically detected)

Please take attention to the manual of the specific temperature controller.
See: FAQ`s

GDS - data file:
Will be delivered with each slave on disk.

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 ≥ 60 V
- unshielded lines for AC voltage ≥ 25 V
- coaxial lines for monitors

Group B:
- unshielded lines for DC voltages ≥ 60 V and ≥ 400 V
- unshielded lines for AC voltage ≥ 24 V and ≥ 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:

	Group A	**Group B**	**Group C**
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 con-nected 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 shiel-ding effect.

In general, you should always connect the shields of cables at both ends. The only way of achie-ving 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-technologie trained personnel following the safety regulations may do the PROFIBUS - connections.
It is essential, that one has well experience in installing a profibus device.

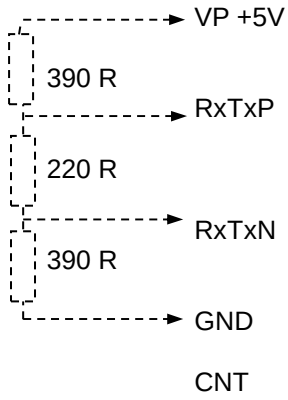
You will require the following components to connect the slave:
- Connector for Profibus connection to the slave
- PROFIBUS cable (this cable is generally already installed on site!)
- Diskette with type resp. GSD file
- Project planning tool for the PROFIBUS-Master

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

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



PROFIBUS – Adjustments:

Adjust the following parameters (slave):

Series R1140:

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

PROFIBUS - Diagnostic displays:
The diagnostic display works with the help of the decimal-point within parameter „Adr – Adress“.
The slaves of the series R11400 have to be set to „remote operation“ with the parameter „rEno“.
This makes sure, that the parameter values can be written via the profibus.

Dec.point, permanent on:	The slave is in the data-exchange-modus. The communication is ok. The data-exchange with the master takes place.
Dec.point, flashing:	The bus is detected. The slave is waiting until the master has programmed the slave. This happens automatically.
Dec.point off:	The slave is not correct connected to the bus. E.g.:- Maybe there is a wiring error. - The master is not active.

3. Data transfer, general

The Communication:

The master sends it´s data to the slave.
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 moduls are available for the slave:

1. Process reflection: Module: „1 – channel process data“

2. Configuration channel: Module: „parameter channel“

3. Process reflection and Configuration channel: Module: „1 – channel process + parameter“

3.1 Process reflection

Parameter transfer according to the process reflection modul:

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

Byte 1	Byte 2	Byte 3
Setpoint 1 High Byte	**Setpoint 1** Low Byte	**Control byte**

WARNING: **Every change of the setpoint is stored in the internal nonvolatile memory. It permits max. 1.000.000 write cycles!**

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

Example:	℃	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 consist out of one data byte:

Bit 0: controller / slave 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 selftuning action.
Set this bit to „0“, before starting a new selftuning action.
Bit 2: 0
Bit 3: actual setpoint 0 = setpoint SP1, 1 = setpoint SP2
Bit 4: 1 = delete warning „selftuning error“ into the status byte
Bit 5: 0
Bit 6: 0
Bit 7: 1 = delete warning „system error“ into the status byte

3.1.2 From slave to master:
Transfer of the process data

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
Status Setpoint High Byte	**Status Setpoint** Low Byte	**Actual Process temperature** High Byte	**Actual Process temperature** Low Byte	**Status byte**	**Alarm status**

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

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

Bit 1–15: no function

Status byte: The parameter consist out of one data byte:

Bit 0: controller / slave 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 = selftuning error

Bit 5: 1 = setpoint ramp function active

Bit 6: 1 = sensor error

Bit 7: 1 = system error

Alarm status: Bit 0 1 = alarm 1 active

Bit 1 1 = alarm 2 active

Bit 2–7: no function

3.1.3 Transmission example
From master to slave: transfer of setpoint 1 and control byte

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

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

Byte 1	Byte 2	Byte 3
Setpoint High Byte 0x01	**Setpoint Low Byte** 0xF4	**Control byte** 0x00

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: act.process temp. value: 55,0℃ 550 dec. = 0x226hex. as a 16 bit integer-value

Byte 5: Controller status: controller = on

Byte 6: Alarm status: alarm = no alarm

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
Status Setpoint High Byte 0x00	**Status Setpoint** Low Byte 0x00	**Actual Process temperature** High Byte 0x02	**Actual Process temperature** Low Byte 0x26	**Status byte** 0x00	**Alarm status** 0x00

3.2 Configuration channel

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

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

3.2.1 Data transmission, general

The PROFIBUS – master is allowed to monitore and controll all parameters of the slave.
The transfer of instructions and parameter values takes place with the aid defined data blocks.

3.2.2 Terms

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

3.2.3 Parameter values

Instruction-code [**BC**]: 0x10, 0x20, 0x21
Parameter-code [**PC**]: 0x00...0xFF
Parameter-value [**PW**]: 16 bit integer, mantissa **PWH** and **PWL** and exponent **PWE** base 10

Parameter-value High-Byte [**PWH**]
Parameter-value Low- Byte [**PWL**]
Parameter-exponent [**PWE**]

3.2.4 Configurating of the parameters via the configuration channel.

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Current number 0x00 ... 0xFF	**Always:** 0x01	**Instruction code** **BC** 0x10, 0x20 od. 0x21	**Always:** 0x00	**Parameter-code** **PC** 0x00 ... 0xFF	**Parameter-value** **PWH** High-Byte	**Parameter-value** **PWL** Low-Byte	**Exponent** **PWE** 0x00 ... 0xFF

Byte 1
Current Number: For every new task the master should preset a current number.
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: always 0x01

Byte 3
Instruction code, BC: 0x10 : Read parameter
0x20 : Write parameter
0x21 : Write parameter and store with powerfail 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:**
Adresses 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 orto high)
05 H - Byte 2 ≠ 0
06 H - The adressed parameter is a read-only parameter
07 H - Writing of datas 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 autotuning can´t be started)
FEH - Error during writing into the powerfail storage
FFH - General error

Byte 6, 7 und 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 (℃):	215	00D7	00D7	00
Setpoint (℃):	230	00E6	00E6	00
Output ratio, cooling (%)	-16	FFF0	FFF0	00
Setpoint ramp (℃/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.

3.2.5 Parameter list

Parameter	Mnemonics	Parameter-Code	R1140
Actual process values:			
Act. temperature value		0x10	**RO**
Temperature offset value	OFSt	0x18	RW
Sensor configuration	SEn	0x1a	RW
Linear input; decimal points	r. dP	0x1d	RW
Linear input; bottom end value	r. Lo	0x1e	RW
Linear input; top end value	r. Hi	0x1f	RW
Setpoints:			
Act. setpoint	SP, act.	0x20	**RO**
Setpoint 1	SP1	0x21	RW
Setpoint 2	SP2	0x22	RW
Setpoint limitation, low range	SP.Lo	0x2b	RW
Setpoint limitation, high range	SP.Hi	0x2c	RW
Setpoint ramp, rising	SP ↑	0x2f	RW
Setpoint range, falling	SP ↓	0x2d	RW
Alarms:			
Alarm 3, Configuration	Co.A3	0x34	RW
Alarm 2, Configuration	Co.A2	0x35	RW
Alarm value 3	AL3	0x38	RW
Alarm value 2	AL2	0x39	RW
Switching behaviour A3	rE.A3	0x3c	RW
Switching behaviour A2	rE.A2	0x3d	RW
PID parameters „heating“:			
Proportional range (P)	1 P	0x40	RW
Rate time (D)	1 d	0x41	RW
Reset time (I)	1 I	0x42	RW
Cycle time	1 CY	0x43	RW
Control sensitivity	1 Sd	0x47	RW
Dead band / switch-point difference (only for 3-point-contr.)	Sh	0x46	RW
PID parameters „cooling“:			
Proportional range (P)	2 P	0x50	RW
Rate time (D)	2 d	0x51	RW
Reset time (I)	2 I	0x52	RW
Cycle time	2 CY	0x53	RW
Control sensitivity	2 Sd	0x57	RW

Parameter	Mnemonics	Parameter-Code	R1140
Parameters, 3-point-stepping controller:			
Proportional range (P)	P	0x40	RW
Motor actuating time	tS	0x41	RW
Reset time (I)	tn	0x42	RW
Dead band / switch-point difference	Sh	0x46	RW
Control sensitivity	Sd	0x47	RW
Output ratio:			
Actual output ratio	Y	0x60	**RO**
Manual output ratio	HAnd	0x62	RW
Output ratio limit (heating)	1LY	0x64	RW
Output ratio limit (cooling)	2LY	0x69	RW
Softstart output ratio	So. Y	0x6a	RW
Softstart setpoint	So.Sp	0x6b	RW
Softstart duration time	So.ti	0x6c	RW
Softstart function on/off	So.St	0x6d	RW
Status words:			
Controller status / status byte		0x78	RW
Controller configuration:			
Control action	ConF	0x80	RW
Out1 or Out2 – configuration: bist. voltage output	Out4	0x83	RW
Adjustment lock	LOC	0x85	RW
Self tuning	OPt	0x88	RW
Manual output configuration (PID)	Hand	0x8b	RW
Controller off/on	Cont	0x8f	RW

3.2.6 Transmission examples

3.2.6.1 Configuration channel, Instruction code: 10 H

The slave is asked, to send the parameter „Process value, 10 H“ to the master.
The process value is 225 °C. 225 (Decimal) = 0xE1 (Hex)

Master to slave:	**Dec.**	**Hex**
Current number:	1	0x01
Always:	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

Slave to master:	**Dec.**	**Hex**
Current number of instruction:	1	0x01
Always:	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.

3.2.6.2 Configuration channel, Instruction code: 20 H

The slave gets the instruction:
"Overtake parameter „prop.-band heating“ (parameter code: 40H, parameter value: 5,0 %)
and store into the RAM".

Master to slave:	**Dec.**	**Hex**
Current number:	2	0x02
Always:	1	0x01
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, 0x01, 0x20, 0x00, 0x40, 0x00, 0x32, 0x01

Slave to master:	**Dec.**	**Hex**
Current number of instruction:	2	0x02
Always:	1	0x01
Instruction code:	32	0x20
Always:	0	0x00
Parameter code (Prop-band, heating):	0 *)	0x00
Parameter value (High-Byte):	0	0x00
Parameter valuewert (Low -Byte):	0	0x00
Dec.-point	0	0x00

Transmission to master: 0x02, 0x01, 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.

3.2.6.3 Configuration channel, Instruction code: 21 H

The slave gets the instruction:
"Overtake parameter setpoint 1 / SP1 = 200℃ (param eter code: 21H) and store powerfailsafe into the EEPROM“.

Master to slave:	**Dec.**	**Hex**
Current number:	3	0x03
Always:	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

Slave to master:	**Dec.**	**Hex**
Current number of instruction:	3	0x03
Always:	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.

3.3 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
Setpoint High Byte	**Setpoint** Low Byte	**Control byte**

Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9	Byte 10	Byte 11
Current number	**always:** 0x01	**Instruction code BC**	**always:** 0x00	**Parameter-code PC**	**Parameter-value PWH** High Byte	**Parameter-value PWL** Low Byte	**Dec.-point PWK**

Slave to master:

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
Status Setpoint High Byte	**Status Setpoint** Low Byte	**Actual Process temperature** High Byte	**Actual Process temperature** Low Byte	**Status byte**	**Alarm status**

Byte 7	Byte 8	Byte 9	Byte 10	Byte 11	Byte 12	Byte 13	Byte 14
Current number	**always:** 0x01	**Instruction code BC**	**always:** 0x00	**Parameter-code PC**	**Parameter-value PWH** High Byte	**Parameter-value PWL** Low Byte	**Dec.-point PWK**

FAQ: www.elotech.de / Products / Technical Data