ELOTECH
INDUSTRIEELEKTRONIK GMBH

Manual:
Data Transfer

Profibus DP

Single Basic Controller
SBC Type: R8300...

Content:

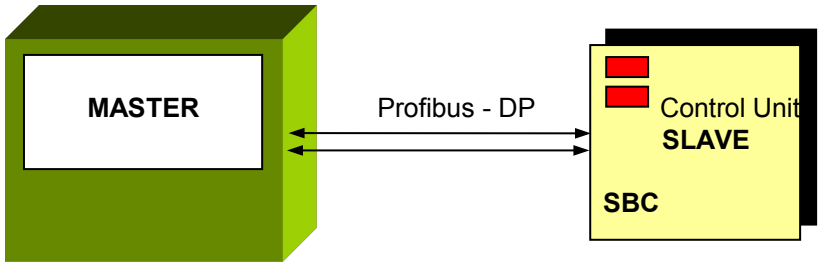
ELOTECH Industrieelektronik GmbH
Verbindungsstraße 27
D - 40723 HILDEN
FON +49 2103 / 255 97 0 FAX +49 2103 / 255 97 29
www.elotech.de Email: info@elotech.de

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.

1. Interface, general

The “basic control unit” **SBC** (slave) is equipped with a PROFIBUS DP interface.
The interface is electrically isolated by the rest of the device circuit.



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 address 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 data.
The old parameter values are still valid.

Slave adjustments:

Unit address (Adr):	The address of the slave 1...255 has to be adjusted into the configuration level. See parameter „Adr“.
Baud rate (bAUd):	9,6 kBaud ... 12 MBaud (with automatically detection)

Please take attention to the manual of the slave (SBC).

GDS - data file:
Will be delivered by Single Temperiertechnik, Hochdorf (Germany).
Internet: www.single-temp.de

1.1 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 .

1.2 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 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 connec-tors. Attach the shield of the data line to the plug or connector housing. Do not connect the shield to PIN 1 of the multi pole connector!

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!

1.3 Connection guide

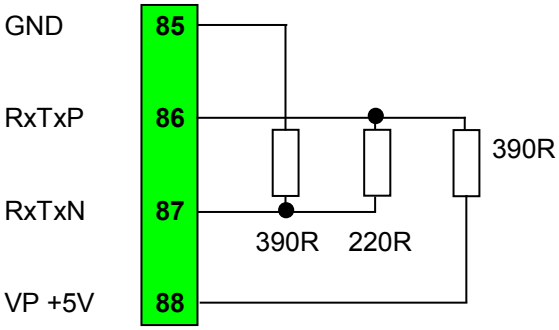
Note: Only in PROFIBUS- technology trained personnel following the safety regulations may do the PROFIBUS - connections.
It is essential, that one has well experience in installing a profibus device.
See also the FAQ – list.

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.

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



The terminals Vp and GND have to be used to connect the terminating-resistors.
There is no further load allowed.

PROFIBUS – Adjustments: Adjust the following parameters (slave):
Parameter „Address“, Adr: „Profibus- address“
Parameter „Baud rate“, b: No adjustment possible.
The baud rate will detected and monitored automatically.
Display: „ndt“ = no baud rate detected.

Diagnostic displays:
If the parameter „address“ (Adr) is selected on the slave, the following diagnostic informations will be displayed in display „SET“:

1 decimal point, permanent on:	The slave is in the data-exchange-modus. The communication is OK. The data-exchange with the master takes place.
1 decimal point, flashing:	The bus is detected. The slave is waiting until the master has programmed the slave. This happens automatically.
No decimal point flashing or permanent on:	The slave is not correct connected to the bus. E.g.:- Maybe there is a wiring error. - The master is not active. - The protocol isn't selected in the right way.
3 decimal points, flashing:	Hardware error of the slave. No communication possible. Please return the slave. The controller-function of the slave itself is further possible.

2. Parameter transmission

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: „SBC Process Data“
2. Configuration channel:	Module: „Parameter Channel“
3. Process reflection and Configuration channel:	Module: „SBC Process Data+Parameter“

2.1 Process reflection

Parameter transfer according to the process reflection modul:

2.1.1 From master to slave: Transfer of Setpoint 1 and Control word

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

Setpoint: The parameter value consists out of 2 data bytes.

Example:	Dec.	Hex.	High-Byte	Low-Byte
Setpoint (°C):	230	00E6	00	E6

(Means 230°C or 230°F – see parameter list, parameter “CF”)

Transmission of the data value takes place without decimal point.
If measuring range is with decimal point: e.g. 150 means 15,0
If measuring range is without decimal point: e.g. 150 means 150

Control word:

Bit 0:	slave „on“ or „off“	1 = on
Bit 1:	slave „cool down“ and „off“	1 = on
Bit 2:	---	---
Bit 3:	suction mode	1 = on
Bit 4:	evacuating mode	1 = on
Bit 5:	2nd. setpoint	1 = on
Bit 6:	auto tuning *)	1 = on
Bit 7:	---	---

*) Bit 6 „Auto tuning“:
The changing from „0“ to „1“ forces one auto tuning action.
Before starting another auto tuning function, set bit 6 first to „0“ again.
If Bit 6 is set on „0“ the running auto tuning circle stops.
You can read the actual state of auto tuning in the process data state.

2.1.2 From slave to master: Transfer of the process data

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Status-Instruction	**Actual Process temperature** High Byte	**Actual Process temperature** Low Byte	**0x00**	**0x00**

Byte 6	Byte 7	Byte 8	Byte 9	Byte 10	Byte 11
0x00	**0x00**	**0x00**	**0x00**	**0x00**	**0x00**

Byte 12	Byte 13	Byte 14	Byte 15	Byte 16	Byte 17
0x00	**0x00**	**Controller output** 0x9C...0x64	**Alarms 1**	**Alarms 2**	**Status**

Definition „Status Instruction“ : Indicates, if a range error has been detected, when writing the setpoint.
0 = setpoint value OK.
1 = setpoint value faulty

Definition „Alarms 1“ :
Bit 0 = collecting alarm
Bit 1 = alarm 1
Bit 2 = ---
Bit 3 = alarm pump (motor protection)
Bit 4 = alarm filling level
Bit 5 = alarm flow transducer and through-flow
Bit 6 = system error
Bit 7 = auto-tune error

Definition „Alarms 2“ :
Bit 0 = alarm pre-flow
Bit 1 = ---
Bit 2 = ---
Bit 3 = alarm sensor breakage (act. sensor)
Bit 4 = ---
Bit 5 = ---
Bit 6 = running dry
Bit 7 = ---

Definition „Status“ :

Bit 0 = slave on / off	1 = on
Bit 1 = slave „cool down“ and „out“	1 = on
Bit 2 = ---	---
Bit 3 = suction mode	1 = on
Bit 4 = evacuating mode	1 = on
Bit 5 = 2nd. setpoint	1 = on
Bit 6 = auto tuning	1 = on
Bit 7 = hand- or remote-operation	1 = hand

2.1.3 From master to slave: Example; transfer of setpoint 1 and control word

Parameter “CF” should be set to “°C” (not “°F”, no decimal point).

Byte 1 + 2: The setpoint 50°C should be send to the slave.
Setpoint: 50 decimal = 0x0032 hexadecimal as a 16 Bit integer-value

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

Byte 1	Byte 2	Byte 3
Setpoint High Byte	**Setpoint Low-Byte**	**Control word**
0x00	0x32	0x01

Answer from slave to master: Transmission of the process reflection

The slave sends the following parameter-values:

Byte 1: status instruction the last instruction was ok.
Byte 2 + 3: pre-flow temperature 55 decimal = 0x0037 hexadecimal as a 16 Bit integer-value

Byte 4 - 13: 0x00

Byte 14: Output ratio -33 decimal = 0xDF hexadecimal as a 8 Bit integer-value
Byte 15: alarms 1 no alarm
Byte 16: alarms 2 collecting-alarm is active
Byte 17: status the slave is switched „on“

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Status-Instruction	**Actual Process temperature** High Byte	**Actual Process temperature** Low Byte		
0x00	0x00	0x37	0x00	0x00

Byte 6	Byte 7	Byte 8	Byte 9	Byte 10	Byte 11
0x00	0x00	0x00	0x00	0x00	0x10

Byte 12	Byte 13	Byte 14	Byte 15	Byte 16	Byte 17
		Controller output -100…+100	**Alarms 1**	**Alarms 2**	**Status (read)**
0x00	0x2F	0xDF	0x00	0x04	0x01

2.2 Configuration channel

With the help of the configuration channel each parameter can be addressed individually.
The PROFIBUS DP- master is able to read all available data and to change all write- data.

The instruction- or parameter transfer is executed in both directions by means of defined data blocks.

Terms

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

Parameter values

Instruction-code	[**BC**]:	0x10, 0x20, 0x21
Parameter-code	[**PC**]:	0x00...0xFF
Parameter-value	[**PW**]:	16 bit integer, real numerical value **PWH** and **PWL** and comma **PWK**

Parameter-value High-Byte	[**PWH**]
Parameter-value Low- Byte	[**PWL**]
Comma, decimal point)	[**PWK**]

2.2.1 Configuration 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	**Always:**	**Instruction code**	**Always:**	**Parameter-code PC**	**Parameter-value PWH**	**Parameter-value PWL**	**decimal point Comma PWK**
0x00 ... 0xFF	0x01	**BC** 0x10, 0x20 od. 0x21	0x00	0x00 ... 0xFF	High-Byte	Low-Byte	0x00 ... 0xFF

Byte 1
Current Number: For every new task the master should pre-set a current number.
This number will be repeated from the slave with each answer.
So it is possible to find out which instructions and which answers are belonging together.

Byte 2: Always 0x01

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)
Unit is not into remote operation.
04 H - Non-compliance with specified range (value to low orto high)
05 H - Byte 2 ≠ 0
06 H - The addressed parameter is a read-only parameter
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, writing into the power fail storage not possible
FFH - General error

Byte 6, 7 and 8
Parameter value: The parameter value comprises three data bytes:
2 data byte (PWH and PWL), 1 data byte (PWK, decimal point).

Byte 6: Parameter value **PWH**
Byte 7: Parameter value **PWL**
Byte 8: Comma (decimal point) **PWK**

Examples:	Dec.	Hex.	PWH	PWL	PWK
Process value (°C):	215	00D7	00	D7	00
Setpoint (°C):	230	00E6	00	E6	00
Output ratio, cooling (%)	-16	FFF0	FF	F0	00
Setpoint ramp (°C/min):	2,2	0016	00	16	FF

The parameter value is calculated as follows:
Dec.: 2,2 = 22 with one decimal point
Hex.: = 0016 (PWH PWL)
= 01 (comma / decimal point)

Negative data values:
Built binary two´s complement.

2.2.2 Parameter codes (Tab 1)

Parameter	Display	Parametercode	Attribut	Others
Working level				
1. Setpoint		0x21	RW	
Alarm value	AL	0x38	RW	
2. Setpoint	SP2	0x22	RW	
Pre-flow alarm value (internal controller)	AP.I	0x3a	RW	
Aqua timer	Ati	0xa0	RW	
Change time	Cti	0xa1	RW	
Leakage stop action	LS	0xa7	RW	
Alarm 2 limit	AL2	0x3D	RW	only, if: ConF=2Pc
Pre-flow temperature value	t.out	0x13	RO	
Parameter level				
Actual output ratio	Y%	0x60	RO	
Output ratio limitation (heating)	hL%	0x64	RW	
Output ratio limitation (cooling)	cL%	0x69	RW	
Prop. range P (heating)	h P	0x40	RW	
Rate time I (heating)	h d	0x41	RW	
Reset time D (heating)	h I	0x42	RW	
Prop. range P (cooling)	c P	0x50	RW	
Rate time I (cooling)	c d	0x51	RW	
Reset time D (cooling)	c I	0x52	RW	
Dead band	db	0x46	RW	
Cycle time (heating)	h C	0x43	RW	
Cycle time (cooling)	c C	0x53	RW	
Setpoint limitation, high	SP.Hi	0x2c	RW	
Setpoint limitation, low	SP.Lo	0x2b	RW	
System lock, temperature value	SCL	0xa2	RW	
Selection: degree C or degree F	C F	0x1b	RW	
Self tuning (Auto tune)	OPt	0x88	RW	
Setpoint ramp function, up	SP./	0x2f	RW	
Setpoint ramp function, down	SP.\	0x2e	RW	
Hysteresis, to switch-on Cooling	HY.Hi	0x5A	RW	only, if: ConF=2Pc
Hysteresis, to switch off Cooling	HY.Lo	0x59	RW	only, if: ConF=2Pc

Parameter code (Tab 2a)

Parameter	Display	Parameter code	Attribute	Others
Configuration level				
Key look	LOC	0x85	RW	
Lock: key NIVEAU (level)	niv	0x92	RW	
Cool down temperature	c60	0x93	RW	
Direct cooling	cdi	0x94	RW	
Configuration alarm1	C.AL	0x34	RW	
Configuration collecting alarm	C.SA	0x36	RW	
Change logic	ChL	0xa8	RW	
Permission: Leakage stop operation	E.LS	0x9c	RW	
Aquatimer, start time	Ast	0xa9	RW	
Emergency Off	EMO	0x90	RW	In again-switch-lock
Offset: controller temp. sensor	OF1	0xab	RW	
Offset: pre-flow temperature	OF4	0xae	RW	
Offset: output ratio cooling	OF6	0x8a	RW	
Time constant factor: X-filter	P.Fi	0x8b	RW	
Device No. 1	dn1	0x9e	RW	
Device No. 2	dn2	0x9f	RW	
Configuration „OFF“	C.OFF	0x8c	RW	
Configuration Alarm 2	C.A2	0x36	RW	only, if: ConF=2Pc
Controller function, configuration	ConF	0x80	RW	

Parameter code (Tab 2b)

Other parameters:				
Parameter	**Display**	**Parameter code**	**Attribut**	**Others**
Act. process temperature value		0x10	RO	
Act. pre-flow temperature		0x13	RO	
Act. setpoint		0x20	RO	
Device on / off		0x8f	RW	
Device type		0x01	RO	

2.2.3 Transmission example: 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
Comma / decimal point:	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
Comma / decimal point:	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.

2.2.4 Transmission example: 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
Comma / decimal point:	1	0x01

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

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
Comma / decimal 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.

2.2.5 Transmission example: Configuration channel, Instruction code: 21 H

The slave gets the instruction:
"Overtake parameter setpoint 1 / SP1 = 200°C (parameter code: 21H) and store power fail safe 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
Comma / decimal point:	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
Comma / decimal point:	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.

2.3 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 word**

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	**Comma / decimal point: PWK**

Slave to master:

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Status-Instruction	**Process temperature** High Byte	**Process temperature** Low Byte	0x00	0x00

Byte 6	Byte 7	Byte 8	Byte 9	Byte 10	Byte 11
0x00	0x00	0x00	0x00	0x00	0x00

Byte 12	Byte 13	Byte 14	Byte 15	Byte 16	Byte 17
0x00	0x00	**Controller output**	**Alarms 1**	**Alarms 2**	**Status (read)**

Byte 18	Byte 19	Byte 20	Byte 21	Byte 22	Byte 23	Byte 24	Byte 25
Current number	**always:** 0x01	**Instruction code BC**	**always:** 0x00	**Parameter-code PC**	**Parameter-value PWH** High Byte	**Parameter-value PWL** Low Byte	**Comma / decimal point: PWK**