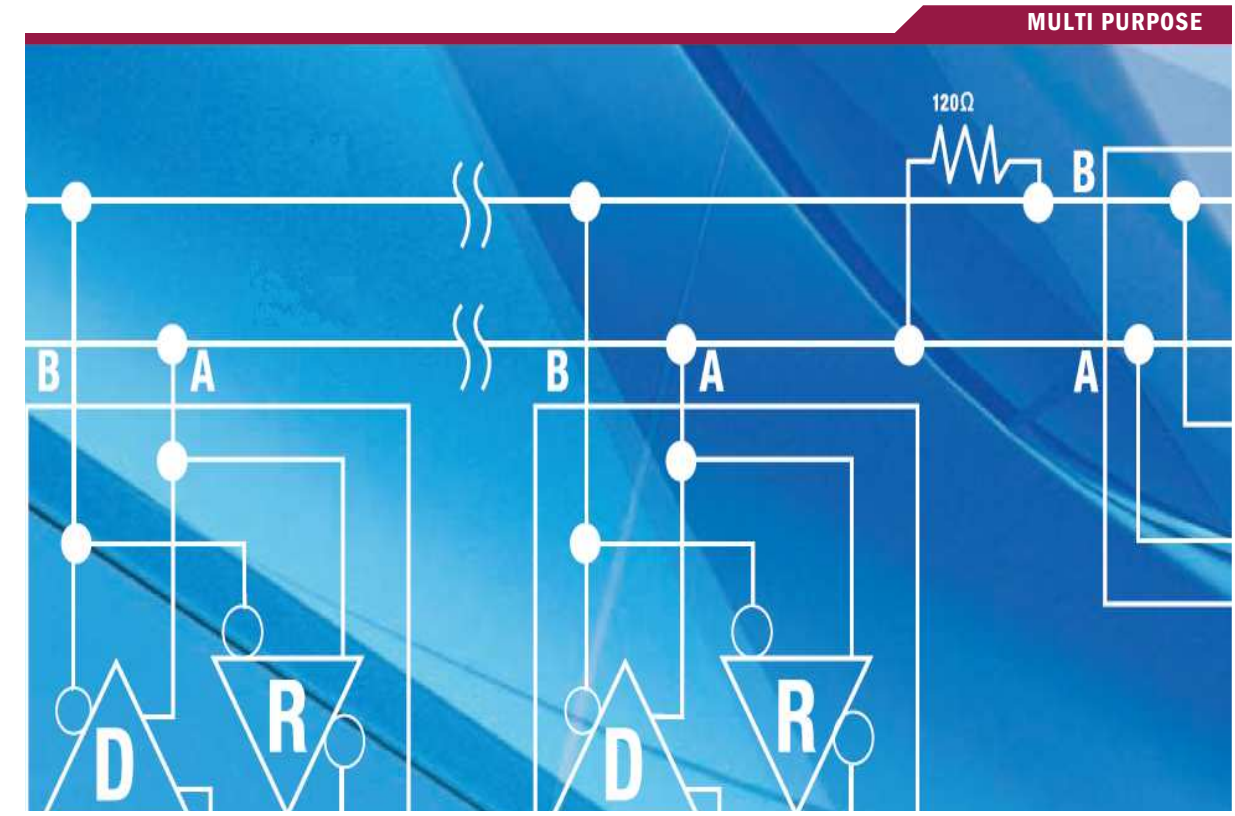
Operating Instructions

RS 485 Interface



Art. Nr.: 90003901GB
Stand: 10.12.2009
Version: 01

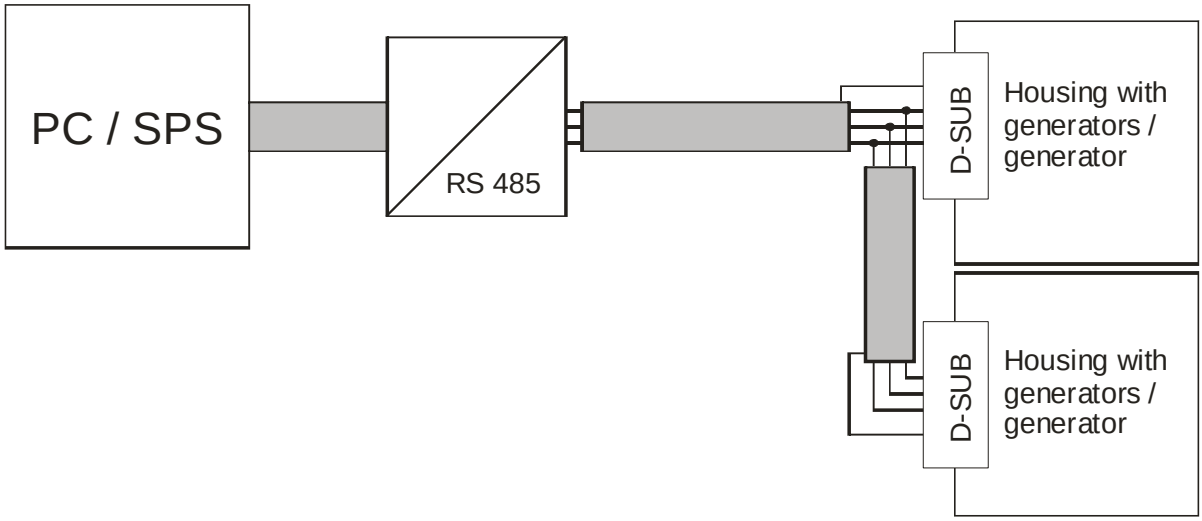
WEBER Ultrasonics

Connecting generators to a host control system via RS485

Contents

Weber Ultrasonics RS-485 interface

The RS485 interface is designed for serial data transfer over long distances (up to 500 metres) and is used in industrial applications. The Weber Ultrasonics RS485 interface was designed as a bidirectional system.



1 Interface Description

1.1 Configuration of the 15-pole d-sub socket (ULC, LC, HS)

Pin No.	Signal
8	TRD+
10	TRD-
15	GND

1.2 Configuration of the 9-pole d-sub socket (Module Generator)

Pin No.	Signal
2	GND
4	TRD-
9	TRD+

2 The master-slave principle

Communication via the RS485 between a host control system (PC or other controls) and components uses the master-slave principle. In this arrangement the host control system is the master, while the generators are the slave. This means that the generators only respond when they are sent an inquiry.
As a general rule, a generator always responds within approximately 300 ms. If no answer is issued within this time, the host control system must send a new inquiry.
There are 2 types of telegrams:

- Inquiry
  The host control system requests data from the generator via an inquiry telegram
- Statement
  The host control system sends data to the generators using statement telegrams.

If the host control system commences a new (<Start of Text>) communication during the response time (also with other generators), the generator discards its response and does not answer.
If the generator detects an error, it responds in a full telegram with the ~ character (0x7E) or 126 Dec.
If the host control system detects an incorrect telegram, it has to then resend the inquiry or statement telegram. (the host control system must not request a new response with the ~ telegram)

3 The interface setting

The interface is set to 19200 bd, 8 data, no parity, 1 stop bit.

4 The Telegram

A telegram consists of the following components:

<Start of Text> <Generator Address> <Function ID> < Data > .... <Data> <End of Telegram>

4.1 Start of Text

The Start of Text character is:

Character	Hex	Decimal
$	0x24	36

When a generator receives this character, the telegram commences for that generator.
If another <Start of Text> is issued while the telegram is being transferred, the generator discards all characters received in the meantime and restarts the telegram from the beginning.

4.2 Generator Address

Every generator has its own address. This allows each generator to determine which telegrams to receive. The address can range from 1 to 255, although the addresses **36 Dec** (<Start of Text>) and **13 Dec** (<End of Telegram>) are exceptions. Example: ID 65 ☐ the generator is addressed with an ASCII A.
Since almost all characters range from 1 to 255, it is not possible to use visible characters as addresses, e.g. 1, 2, 3, etc.
Anyone wishing to use solely ASCII characters could, for example, also start with 48 Dec (0 ASCII) and then continue via upper case letters, right up to the lower case letters.

4.3 Function ID and Data

The function ID tells the generator what kind of action it has to trigger. It then reacts accordingly, processing the data in this.

4.4 End of Telegram

The End of Telegram character is:

Character	Hex	Decimal
<cr>	0x0D	13

Each telegram is completed with a <cr>. When an <End of Telegram> character is issued, the generator uses the address to check whether it is the intended recipient of the telegram.

5 Programming parameters

5.1.1 Writing amplitude

generator receives	$<ID>Axxxz<cr>	
generator answers	$<ID>!<c>	telegram understood
	$<ID>~<cr>	telegram not understood
xxx amplitude in % [050..100] z = 0 do not store amplitude in EEPROM z = 1 store amplitude in EEPROM		

5.1.2 Reading amplitude

generator receives	$<ID>A<cr>	
generator answers	$<ID>Axxx <cr>	telegram understood
	$<ID>~<cr>	telegram not understood
The actual output value is given in xxx% [050..100]%		

5.1.3 Writing trigger delay

generator receives	$<ID>Bxxx<cr>	
generator answers	$<ID>!<cr>	telegram understood
	$<ID>~<cr>	telegram not understood
xxx Time in ms [000...999] ☐ [0.00s..9.99 s]. If the value is 000, the function is switched off.		

5.1.4 Reading trigger delay

generator receives	$<ID>B<cr>	
generator answers	$<ID>Bxxx <cr>	telegram understood
	$<ID>~<cr>	telegram not understood
xxx Time in ms [000...999] ☐ [0.00s..9.99 s]. If the value is 000, the function is switched off.		

5.1.5 Writing welding mode

generator receives	$<IDCx<cr>	
generator answers	$<ID>!<cr>	telegram understood
	$<ID>~<cr>	telegram not understood
x = 0 remote mode x = 1 timer mode x = 2 energy mode x = 3 peak power mode x = 4 pulse mode		

5.1.6 Reading welding mode

generator receives	$<IDC<cr>	
generator answers	$<ID>Cx<cr>	telegram understood
	$<ID>~<cr>	telegram not understood
x = 0 remote mode x = 1 timer mode x = 2 energy mode x = 3 peak power mode x = 4 pulse mode		

5.1.7 Writing on / off time

generator receives	$<ID>Dxxxyyy<cr>	
generator answers	$<ID>!<cr>	telegram understood
	$<ID>~<cr>	telegram not understood

[001..9.99] => 0.01 s .. 9.99 s

xxx time [001...9.99]	weld time for timer mode	1
	monitoring time for energy mode	2
	monitoring time for peak power mode	3
	weld time for pulse mode	4
yyy off time [001..9.99]	ultrasound off time for pulse mode	4

5.1.8 Reading on/off time

generator receives	$<ID>D<cr>	
generator answers	$<ID>Dxxxyyy<cr>	telegram understood
	$<ID>~<cr>	telegram not understood

[001..9.99] => 0.01 s .. 9.99 s

xxx time [001...9.99]	weld time for timer mode	1
	monitoring time for energy mode	2
	monitoring time for peak power mode	3
	weld time for pulse mode	4
yyy off time [001..9.99]	ultrasound off time for pulse mode	4

5.1.9 Writing energy

generator receives	$<ID>Exxxxx<cr>	
generator answers	$<ID>!<cr>	telegram understood
	$<ID>~<cr>	telegram not understood

This parameter is only used in the energy welding mode (2).
xxxxx Energy in Ws [1...65000]

5.1.10 Reading energy

generator receives	$<ID>E<cr>	
generator answers	$<ID>Exxxxx<cr>	telegram understood
	$<ID>~<cr>	telegram not understood

This parameter is only used in the energy welding mode (2).
xxxxx Energy in Ws [1...65000]

5.1.11 Writing peak power reference

generator receives	$<ID>Fxxxx<cr>	
generator answers	$<ID>!<cr>	telegram understood
	$<ID>~<cr>	telegram not understood

This parameter is only used in the peak power welding mode as an output entry. [0001.Pmax]
Pmax is the maximum output of the generator

5.1.12 Reading peak power reference

generator receives	$<ID>F<cr>	
generator answers	$<ID>Fxxxx<cr>	telegram understood
	$<ID>~<cr>	telegram not understood

This parameter is only used in the peak power welding mode as an output entry. [0001.Pmax]
Pmax is the maximum output of the generator
Attention: The value is converted during writing and reading. Rounding errors may therefore occur as a result of division.
The read value is not always the same as the written value!

5.1.13 Writing T-window

generator receives	$<ID>Gxxxyyy<cr>	
generator answers	$<ID>!<cr>	telegram understood
	$<ID>~<cr>	telegram not understood
xxx T-window low [000...999] ☐ [0.00s..9.99s]. If the value is 000, the T-window is switched off yyy T-window high [T-window_low...999] must be greater than T-window low		

5.1.14 Reading T-window

generator receives	$<ID>G<cr>	
generator answers	$<ID>Gxxxyyy<cr>	telegram understood
	$<ID>~<cr>	telegram not understood
xxx T-window low [000...999] ☐ [0.00s..9.99s]. If the value is 000, the T-window is switched off yyy T-window high [T-window_low...999] must be greater than T-window low		

5.1.15 Writing P-window

generator receives	$<ID>Hxxxx yyyy<cr>	
generator answers	$<ID>!<cr>	telegram understood
	$<ID>~<cr>	telegram not understood
Output entry in W [0001..Pmax] Pmax is the maximum output of the generator		

5.1.16 Reading P-window

generator receives	$<ID>H<cr>	
generator answers	$<ID>Hxxxx yyyy<cr>	telegram understood
	$<ID>~<cr>	telegram not understood
Output entry in W [0001..Pmax] Pmax is the maximum output of the generator Attention: the value is converted when writing and reading. Rounding errors may therefore occur as a result of division. **The read value is not always the same as the written value!**		

.5.1.17 Writing holding time

generator receives	$<ID>Ixxx<cr>	
generator answers	$<ID>!<cr>	telegram understood
	$<ID>~<cr>	telegram not understood
holding time [000..999] ☐[0.00s..9.99s]. If the value is 000, the holding function is switched off		

5.1.18 Reading holding time

generator receives	$<ID>I<cr>	
generator answers	$<ID>Ixxx<cr>	telegram understood
	$<ID>~<cr>	telegram not understood
holding time [000..999] ☐[0.00s..9.99s]. If the value is 000, the holding function is switched off		

5.1.19 Writing afterburst time

generator receives	$<ID>Jxxx yyyy<cr>	
generator answers	$<ID>!<cr>	telegram understood
	$<ID>~<cr>	telegram not understood
xxx afterburst delay [000..999] ☐ [0.00s..9.99s]. If 000 is set, the function is switched off yyyy afterburst time [0000..3000] ☐ [00.00s..30.00s]. If 000 is set, the function is switched off		

5.1.20 Reading afterburst time

generator receives	$<ID>J<cr>	
generator answers	$<ID>Jxxx yyyy<cr>	telegram understood
	$<ID>~<cr>	telegram not understood
xxx afterburst delay [000..999] ☐ [0.00s..9.99s]. If 000 is set, the function is switched off yyyy afterburst time [0000..3000] ☐ [00.00s..30.00s]. If 000 is set, the function is switched off		

5.1.21 Writing counter

generator receives	$<ID>Kxxxxx<cr>	
generator answers	$<ID>!<cr>	telegram understood
	$<ID>~<cr>	telegram not understood
counter value [00000..65535]		

5.1.22 Reading counter

generator receives	$<ID>K<cr>	
generator answers	$<ID>Kxxxxx<cr>	telegram understood
	$<ID>~<cr>	telegram not understood
counter value [00000..65535]		

Parameters used in welding mode

Code C	Welding mode	D On/off time	E Energy	F Peak power	G T-Window	H P-Window
00	Remote				X	X
01	Timer	X1			X	X
02	Energy	X1	X		X	X
03	Peak power	X1		X	X	X
04	Pulse	X			X	X

X = parameter required
X1 = only on-time is required

6 Parameters for running

6.1.1 Generator running

generator receives	$<ID>P<cr>			
generator answers	$<ID>P	telegram understood		
	<a>	a = 0 a = 1	Generator is switched off Generator is switched on	[0..1]
	<bb>	bb	For error code, see parameter R	[00..07]
	<ccccc>	ccccc	Frequency	[00000..65535] Hz
	<ccc>	ddd	Temperature	[000..100] °C
	<eeee>	eeee	Power output	[0001..9999] W
	<ffff>	ffff	Max. power output	[0001..9999] W
	<ggg>	ggg	Power output measuring time	[000.. 999] ➔ [0,00s 9,99s]
	<hhhhh>	hhhhh	Energy in Ws	[00001..65535]
	<iii>	iii	Energy measuring time	[000.. 999] ➔ [0,00s 9,99s]
	<jjj>	jjj	External amplitude in %	[000..100]
	<cr>	End of telegram		

6.1.2 Setting start / stop generator

generator receives	$<ID>Qx<cr>	
generator answers	$<ID>!<cr>	telegram understood
	$<ID>~<cr>	telegram not understood
x = 0 generator is switched off x = 1 generator is switched on Note: If the generator was switched on via the interface, it must also be switched off via the interface.		

6.1.3 Querying condition of generator

generator receives	$<ID>Q<cr>	
generator answers	$<ID>Qx<cr>	telegram understood
	$<ID>~<cr>	telegram not understood
x = 0 generator has been switched off x = 1 generator has been switched on via an interface		

6.1.4 Writing error

generator receives	$<ID>Rxx<cr>	
generator answers	$<ID>!<cr>	telegram understood
	$<ID>~<cr>	telegram not understood
reset error with xx = 00. Each error must be reset after it occurs.		

6.1.5 Reading error and information

generator receives	$<ID>R <cr>	
generator answers	$<ID>Rxx y<cr>	telegram understood
	$<ID>~<cr>	telegram not understood
Generator replies with the error code xx 00 = no error 01 = no RF detected 02 = excess temperature 03 = Iprotect 04 = E-search 05 = cycle not completed 06 = timeout 07 = version mismatch The error has to be reset. Information y 0 = Wait on start 1 = Cycle in Trigger Delay 2 = Cycle in Welding 3 = Cycle in Holding 4 = Cycle in Afterburst Delay 5 = Cycle in Afterburst Welding 6 = Waiting on start inactive		

6.1.6 Querying frequency

generator receives	$<ID>S<cr>	
generator answers	$<ID>Sxxxxx<cr>	telegram understood
	$<ID>~<cr>	telegram not understood
Frequency [00000..65535] Hz		

6.1.7 Querying actual temperature

generator receives	$<ID>T<cr>	
generator answers	$<ID>Txxx<cr>	telegram understood
	$<ID>~<cr>	telegram not understood
xxx temperature [000...100]°C		

6.1.8 Querying output

generator receives	$<ID>U<cr>	
generator answers	$<ID>U xxxx yyyy zzz <cr>	telegram understood
	$<ID>~<cr>	telegram not understood
xxxx = output	[0001..9999] W	
yyyy = peak output	[0001..9999] W	
zzz = measuring time in which the output is produced	[000..999]☐ ➔ [0.00s..9.99s]	

6.1.9 Querying energy

generator receives	$<ID>V<cr>	
generator answers	$<ID>U xxxxx yyy <cr>	telegram understood
	$<ID>~<cr>	telegram not understood
xxxxx = energy in Ws	[00001..65535]	
yyy =	measuring time in which the energy is produced in seconds [000..999] ➔☐ [0.00s..9.99s]	

6.1.10 Reading external amplitude

generator receives	$<ID>W<cr>	
generator answers	$<ID>Wxxx<cr>	telegram understood
	$<ID>~<cr>	telegram not understood
xxx = external amplitude in % [000..100]		

Generator running telegram

This telegram allows you to query all the running parameters via one telegram. Please note that the power output measuring time is only valid for peak power mode and the energy and energy measuring time only represent energy mode.

7 Setting parameters

Lower case trigger parameters are used for the settings. The upper case letters represent numbers.
Example: $Aa133<cr> is displayed as $<ID>aXXX<cr> in the parameter list.

7.1 Adjusting the start-stop frequency

The start-stop frequency is adjusted by altering the two digital software potentiometers. The potentiometers can be set between 0..255. For the starting frequency, 0☐ is the highest frequency, 255☐ is the lowest frequency, and for the stop frequency 0☐ is the lowest frequency and 255☐ is the highest frequency.

Set the value to 255, read out the frequency via the telegram $<ID>S<cr> and then change the value accordingly until the desired frequency is set.

7.1.1 Setting the starting frequency potentiometer

generator receives	$<ID>Axxx <cr>	
generator answers	$<ID>!<cr>	telegram understood
	$<ID>~<cr>	telegram not understood
potentiometer digit value XXX = [000..255]		

7.1.2 Reading the potentiometer setting for the starting frequency

generator receives	$<ID>A<cr>	
generator answers	$<ID>Axxx <cr>	telegram understood
	$<ID>~<cr>	telegram not understood
potentiometer digit value XXX = [000..255]		

7.1.3 Switching on and setting the stop frequency potentiometer

generator receives	$<ID>bXXXZ <cr>	
generator answers	$<ID>!<cr>	telegram understood
	$<ID>~<cr>	telegram not understood
potentiometer digit value XXX = [000..255] Z = 1 stop frequency adjustment activated Z = 0 stop frequency adjustment deactivated If the stop frequency is to be adjusted, it is essential that the parameter is Z = 1 for each adjustment. If the correct value has been calculated, the telegram must be preset again using Z = 0		

7.1.4 Reading the stop frequency setting

generator receives	$<ID>b<cr>	
generator answers	$<ID>aXXXZ <cr>	telegram understood
	$<ID>~<cr>	telegram not understood
potentiometer digit value XXX = [000..255] Z = 1 stop frequency adjustment activated Z = 0 stop frequency adjustment deactivated If the stop frequency is to be adjusted, it is essential that the parameter is Z = 1 for each adjustment. If the correct value has been calculated, the telegram must be preset again using Z = 0		

7.2 Further setting parameters

7.2.1 Writing status

generator receives	$<ID>cX<cr>	
generator answers	$<ID>!<cr>	telegram understood
	$<ID>~<cr>	telegram not understood

7.2.2 Reading status

generator receives	$<ID>c<cr>	
generator answers	$<ID>cX<cr>	telegram understood
	$<ID>~<cr>	telegram not understood
X = 0	not configured	
X = 1	configured	
X = 2	locked	

7.2.3 Writing soft amplitude

generator receives	$<ID>dXXZZ<cr>	
generator answers	$<ID>!<cr>	telegram understood
	$<ID>~<cr>	telegram not understood
XX = steps [00..10] If the value is 00, the function is switched off		
ZZ = ramp-up time [10..50] ☐ [10 ms..50 ms]		

7.2.4 Reading soft amplitude

generator receives	$<ID>d<cr>	
generator answers	$<ID>dXXZZ<cr>	telegram understood
	$<ID>~<cr>	telegram not understood
XX = steps [00..10] If the value is 00, the function is switched off		
ZZ = ramp-up time [10..50] ☐ [10 ms..50 ms]		

7.2.5 Starting factory reset

generator receives	$<ID>eXXXX<cr>	
generator answers	$<ID>!<cr>	telegram understood
	$<ID>~<cr>	telegram not understood
To initiate a factory reset you have to send 2 telegrams for safety reasons. 1st telegram: $<ID>e1234<cr>, the generator responds with $<ID>!<cr> 2nd telegram: $<ID>e6543<cr>, the generator carries out a factory reset and responds with $<ID>!<cr>. If one of the telegrams is incorrect, the process must be started again from the first telegram. The response takes some time as data has to be saved		

7.2.6 Writing input / output configuration

generator receives	$<ID>fWXYZ<cr>	
generator answers	$<ID>!<cr>	telegram understood
	$<ID>~<cr>	telegram not understood
W = 0	amplitude reference value comes from RS232	
W = 1	amplitude reference value comes from analogue input voltage	
X = 0	low HF output active	
X = 1	high HF output active	
Y = 0	low error output active	
Y = 1	high error output active	
Z = 0	low remote input active	
Z = 1	high remote input active	

7.2.7 Reading input / output configuration

generator receives	$<ID>f<cr>	
generator answers	$<ID>fWXYZ<cr>	telegram understood
	$<ID>~<cr>	telegram not understood
W = 0 amplitude reference value comes from RS232 W = 1 amplitude reference value comes from analogue input voltage X = 0 low HF output active X = 1 high HF output active Y = 0 low error output active Y = 1 high error output active Z = 0 low remote input active Z = 1 high remote input active		

7.2.8 Writing PWM start / step

generator receives	$<ID>gXXZZ<cr>	
generator answers	$<ID>!<cr>	telegram understood
	$<ID>~<cr>	telegram not understood
XX = PWM start value [00..31] ZZ = PWM step value [00..31]		

7.2.9 Reading PWM start/step

generator receives	$<ID>G<cr>	
generator answers	$<ID>gXXZZ<cr>	telegram understood
	$<ID>~<cr>	telegram not understood
XX = PWM start value [00..31] ZZ = PWM step value [00..31]		

7.2.10 Writing I limits

generator receives	$<ID>hXXX YYY ZZZ<cr>	
generator answers	$<ID>!<cr>	telegram understood
	$<ID>~<cr>	telegram not understood
XXX = I minimum [000...255] YYY = I maximum [000...255] ZZZ = I protection [000...255]		

7.2.11 Reading I limits

generator receives	$<ID>h<cr>	
generator answers	$<ID>hXXX YYY ZZZ<cr>	telegram understood
	$<ID>~<cr>	telegram not understood
XXX = I minimum [000...255] YYY = I maximum [000...255] ZZZ = I protection [000...255]		

7.2.12 Writing pout

generator receives	$<ID>i XX<cr>	
generator answers	$<ID>!<cr>	telegram understood
	$<ID>~<cr>	telegram not understood
XX = maximum generator output in hundredths E.g. if the generator has a maximum output of 400 W ☐ XX = 04		

7.2.13 Reading Pout

generator receives	$<ID>i<cr>	
generator answers	$<ID>i XX<cr>	telegram understood
	$<ID>~<cr>	telegram not understood
XX = maximum generator output in hundredths E.g. if the generator has a maximum output of 400 W ☐ XX = 04		

7.2.14 Writing temperature

generator receives	$<ID>kXX ZZ<cr>	
generator answers	$<ID>!<cr>	telegram understood
	$<ID>~<cr>	telegram not understood
XX = shut off temperature [00..99] ZZ = fan temperature [00..99]		

7.2.15 Reading temperature

generator receives	$<ID>k<cr>	
generator answers	$<ID>kXX ZZ<cr>	telegram understood
	$<ID>~<cr>	telegram not understood
XX = shut off temperature [00..99] ZZ = fan temperature [00..99]		

7.2.16 Reading timer

generator receives	$<ID>l<cr>	
generator answers	$<ID>l AAAA BB CC DDDD EE FF<cr>	telegram understood
	$<ID>~<cr>	telegram not understood
AAAA = run timer days BB = run timer hours CC = run timer minutes DDDD = ultrasound timer days EE = ultrasound timer hours FF = ultrasound timer minutes		

7.2.17 Reading serial number

generator receives	$<ID>m<cr>	
generator answers	$<ID>m.....<cr>	telegram understood
Reads out the 13-digit serial number and the 6-digit date		

Service hotline

Should you still have questions after reading through the operating instructions thoroughly, please feel free to call our service hotline.

Please have the following information to hand to help us answer your questions quickly.
Device type, serial number (the serial number is on the back of the generator)

Our service hotline number:

+49 (0) 72 48/92 07-0

Produktion, Vertrieb und Service:

Weber Ultrasonics GmbH
Im Hinteracker 7
76307 Karlsbad-Ittersbach
Germany

Phone: +49 (0)72 48 / 92 07-0
Fax: +49 (0)72 48 / 92 07-11
mail@weber-ultrasonics.de
www.weber-ultrasonics.de