The Multizones Temperature Controller:

R1140, R1300, R2000, R2100, R2200, R2400(S), R2500(S), R4000:

Serial Interface, Data Transfer
Protocol description: ELOTECH - Standard

1. 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

2. Interface, general description

The microprocessor based controllers of the series **R1140, R1300, R2X00**
can be equipped with a serial interface RS-485, RS232-C or TTY(0/20mA).

The interface is electrically isolated by the rest of the device circuit.

The data transfer between the slave and master takes place with the aid of a defined protocol.

The communication is always controlled by the master.
The controller operates as a slave with its own address (1...255).
The address has to be programmed in the configuration level of the controller.

If there are transmission or other errors detected by the slave (controller), it does not accept this data.
The old parameter values are still valid.

All data is transferred in a hexadecimal, ASCII-coded format.

Check criterias:
1. Only ASCII-Codes from 0...9 or A...F ?
   Except for start and stop character?
2. Data format (Parity) o.k. ?
3. Check sum o.k. ?

RS-485 interface data:

Number of drivers/receivers:	32
Transmission character:	symmetric
Max. wire length:	1200 m

TAKE ATTENTION TO THE CORRESPONDING OPERATING MANUALS OF THE CONTROLLER !

3. Interface Parameters

The following parameters have to be programmed either in the configuration level of the controller (Types R1xx0, R2x00) or have to be set with the help of the function switches (Type R 2200):
See controller manual --> configuration level.

Adr	**Unit address**	1 .... 255 (ex works: 1) The master addresses the slave at this address. Each slave has its own address. RS485: It is possible to address 32 slaves with different zones.
For	**Data format**	7E1 7 data, even, 1 stop bit 7o1 7 data, odd, 1 stop bit 7E2 7 data, even, 2 stop bit 7o2 7 data, odd, 2 stop bit 7n2 7 data, none, 2 stop bit 8E1 8 data, even, 1 stop bit 8o1 8 data, odd, 1 stop bit 8n1 8 data, none, 1 stop bit 8n2 8 data, none, 2 stop bit
bAud	**Baud rate**	OFF; 0,3 ... 9,6 kBaud (ex works: 9,6) The baud rate denotes the transmission rate at which one bit is transmitted.

Start bit:
At the beginning of the transmission a start bit (log. 0) is transmitted. It´s purpose is to inform the receiver of the start of a data word (synchronization of the data exchange).

Data bit:
The start bit is (starting with the least significant bit) followed by 7 or 8 data bit,.

Parity bit:
The next bit is the parity bit. It is calculated from the check sum of all data bit and enables the receiver to recognize transmission errors.
EVEN - Parity: The number of the ones transmitted (including the parity bit) must be even.
ODD - Parity: The number of the ones transmitted (including the parity bit) must be odd.
NONE - Parity: There is no parity-bit calculated and transmitted.

Stop bit:
The transmission of a data word is concluded with 2 stop bit (log. 1). The purpose of this is to provide a minimum space between two immediately consecutive data words.

EXAMPLE 1

Data format: (7E2) :		1 Start bit	7 Data bit	Parity (EVEN)	2 Stop bit
	Data word:		111 1100		
	Transmission:	0	0011 111	1	1 1

EXAMPLE 2

Data format: (8o1) :		1 Start bit	8 Data bit	Parity (ODD)	1 Stop bit
	Data word:		1111 1100		
	Transmission:	0	0011 1111	1	1

4. Data Transmission / Protokoll

All data (Hex-Byte) are transmitted in ASCII-format (text characters).

Permitted ASCII characters: 30H ... 39H, 41H ...46H,
0AH, 0DH

All other characters would be ignored.

E.g.: Hex-Byte **2F**H -> "**2**" in ASCII: 32H
"**F**" in ASCII: 46H

Two ASCII characters are thus required for each hex byte.
The only exceptions are:
the start character: (OAH = line feed, LF) and
the end character: (ODH = carriage return, CR).

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

4.1 Terms

Start character:	introduces transfer of a data block. All characters in front of the start character are ignored.	(1 ASCII)
Device address:	designates a specific multi zones device	(2 ASCII)
Zone address:	designates the specific controller zone	(2 ASCII)
Instruction code:	tells the device (slave) what it must do	(2 ASCII)
Parameter code:	designates each individual parameter that can be called in the device.	(2 ASCII)
Parameter group code:	some parameters are combined to a group (E.g.: the feedback parameters P, I, D and the switching cycle time C). All this parameters can be set and called up with one instruction.	(2 ASCII)
Parameter value:	states the value of a parameter	(6 ASCII)
Response:	acknowledgement message of the device in response to a master instruction.	(2 ASCII)
Check sum:	the two´s complement of the sum of all the hex bytes of a data block without the start and the end characters. Serves the purpose of recognizing transmission errors.	(2ASCII)
End character:	concludes the transmission of a data block	(1 ASCII)

5. Instruction and Response

The master (computer) can issue the following instructions to the slave (controller):

a. Send parameter: Instruction code 10 H

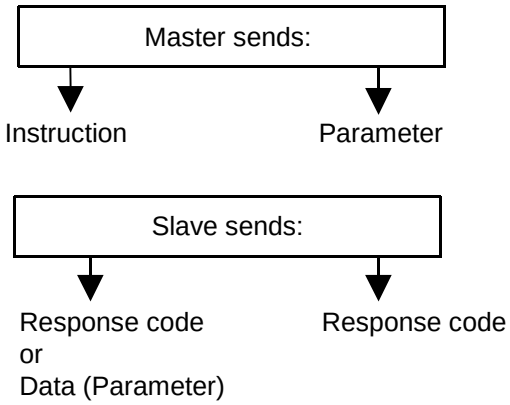
b. Send parameter group: Instruction code 15 H

d. Accept parameter: Instruction code 20 H

e. Accept parameter and store with powerfail protection: Instruction code 21 H
Take care:
The non volatile memory permits max. 1.000.000 write cycles.

Provided, that the slave understood the instruction, it always responds by sending a complete data block.
The typical interval between master instruction and slave response (time-out) is 5...10 ms..
The slave repeats the received instruction code.

5.1 Instruction



5.2 Response (with error code)

00H (no error): **- acknowledge, no error (Instruction executed)**

01H (error message): **- Parity error**

02H (error message): **- Check sum error**

03H (error message) Procedure error:

The device (slave or controller) reports „procedure error“ if unknown instruction or parameter codes or parameter group codes are stated.

Other error messages as follows:

1. The master wishes to edit an illegal alarm value.
   Remedy: initially allow the alarm by means of the alarm configuration.
2. The slave (controller) is operating in the automatic mode (controller mode).
   The master wishes to edit the manual response ratio (62H).
   First switch over to mode „HAnd“.
3. The slave can´t store the parameter „manual output ratio“ power fail.
4. The slave (controller) is configured as a 2-point (heating-off) controller:
   The master requests the „cooling“ parameters.

04H (error message) Non - compliance with specified range:
The slave reports „non-compliance with specified range“ in the following instances:

1. Alarm configuration: Signal contact
   The master wishes to edit +1100 (max. allowed: +999).

2. Actual measuring and controlling range: 0 ... 400°C.
   The master sends 430°C.

05H (error message): **- Zone number not allowed / available.**

06H (error message) Parameter is a read only parameter:

The slave (controller) reports „parameter is only read parameter“ if a read parameter is to be edited via the master.

E. g.:

1. The master sends the response output ratio, although the slave is in the automatic mode (parameter 62H).
2. The master sends status word 1 (parameter 70H) to the slave.
3. The master sends actual process value.
4. The master sends the current setpoint (parameter 20H). Edit SP1 or SP2.
5. 3-point-step controller configuration:
   The master sends a manual output ratio value when the slave is in „HAnd“-mode.

FEH (error message): **- Error during writing into the powerfail storage**

FFH (error message): **- General error**

6. Parameter Structure

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

Examples:	Dec.	Hex.	Mantissa	Exp.	ASCII
Process value (°C):	215	00D7	00D7	00	30 30 44 37 30 30
Setpoint (°C):	230	00E6	00E6	00	30 30 45 36 30 30
Output ratio, "cooling" (%)	-16	FFF0	FFF0	00	46 46 46 30 30 30
Setpoint ramp (°C/min):	2,2	0016	0016	FF	30 30 31 36 46 46
	The parameter value is calculated as follows: Dec.:2,2 = 22 x 10 Exp. -1 Hex.: = 0016 (mantissa) = FF (exponent)				
Status word (mantissa: low byte)	1	0001	0001	00	30 30 30 31 30 30

Negative mantissas / negative exponents : Built binary two´s complement.

7. Check Sum

The checksum is formed by subtracting the hex data of a data block (without start- and end characters) from 00H (two´s complement of the sum). Carryovers are disregarded.

Example:

Device address = 01dec.:	01	00 - 01 = FF	(device address no. 1)
Zone number:	01	FF - 01 = FE	(zone no.: 1)
Instruction code:	10	FE - 10 = EE	(send parameter)
Parameter code:	10	EE - 10 = DE	(process value)
Checksum:		**DE**	

Try to program this example first and send it to the device.
Don not forget the start- and stop-byte (but do ignore, when you calculate the check sum).

If this and the unit address, the baudrate and the data format is OK,
the device with the address1 sends the process value of zone no.1.

8. Status Word 1, Parameter code 70H

For each zone is status word 1 (read only) available. It has 8 bit.
Reports alarm states ascertained by the controller.

Mantisse, low byte:

7 6 5 4 3 2 1 0 :	bit 0 = 1	-> System error
bit	bit 1 = 1	-> Sensor error
	bit 2 = 1	-> Restart Lockout aktive (only R4000)
	bit 3 = 1	-> reset-control. A reset was triggered during operation. The device automatically resets bit 3 = 0 if the status word 1 has been read once by the master.
	bit 4 = 1	-> Softstart active
	bit 5 = 1	-> Alarm 1 „on“
	bit 6 = 1	-> Alarm 2 „on“
	bit 7 = 1	-> Setpoint ramp in operation

9. Reset Error bits

Reset of Error bits, Parameter code 0x9d

This parameter is write only!

Bit 0: Clear System error (err8) (only R2x00S und R4000)
Bit 1: Clear Autotune error (err7) (only R2x00S und R4000)
Bit 2: Clear Restart Lock-out (only R4000)
….
Bit 8: Clear self retaining Alarm 1 (only R4000)
Bit 9: Clear self retaining Alarm 2 (only R4000)

10. Data Block Structure

10.1 Master sends „Instruction“, Instruction code: 10H, 15H

Start	**0A**	**xx xx**	**xx xx**	**xx xx**	**xx xx**	**xx xx**	**0D**	End
	Lf	Device-address	Zone-address	Instruction-code	Param.-code	Check sum	CR	

10.2 Master sends "Parameter", Instruction code: 20H, 21H

Start	**0A**	**xx xx**	**xx xx**	**xx xx**	**xx xx**	**xx xx xx xx xx xx**	**xx xx**	**0D**	End
	Lf	Device-address	Zone-address	Instruction-code	Param.-code	Mantissa Exp. Parameter	Check sum	CR	

10.3 Slave sends „Response“ to master:

Start	**0A**	**xx xx**	**xx xx**	**xx xx**	**xx xx**	**xx xx**	**0D**	End
	Lf	Device-address	Zone-address	Response-code Instruction-code	Response Error-code	Check sum	CR	

10.4 Slave sends „Parameter“ or „Parameter group“ to master (Data transfer)

Start	**0A**	**xx xx**	**xx xx**	**xx xx**
	Lf	Device-address	Zone-address	Response-code Instruction-code

xx xx	**xx xx xx xx xx xx**
Param.-code 1	Mantissa Exp. Parameter value 1

xx xx	**xx xx xx xx xx xx**	**xx xx**	**0D**	End
Param.-code n	Mantissa Exp. Parameter value n	Check sum	CR	

xx : 1 ASCII character

11. Typical Transmission Examples

11.1 Transmission example, Instruction code 10 H

The device No.(address) 5 (zone address:01) is called to send a parameter (process value, 10 H) to the master.

Master to device (slave):	Dec.	Hex		ASCII (Hex)
Start character				0A
Device address:	5	05	-->	30 35
Zone address:		01	-->	30 31
Instruction: Send parameter		10	-->	31 30
Parameter code (process value):		10	-->	31 30
Check sum:		DA	-->	44 41
End character:				0D

Transmission to controller: 0A 30 35 30 31 31 30 31 30 44 41 0D

Device (slave) to master:	Dec.	Hex		ASCII
Start character				0A
Device address:	5	05	-->	30 35
Zone address:		01	-->	30 31
Send parameter (repeats instruction):		10	-->	31 30
Parameter code (process value):		10	-->	31 30
Parameter value:	225	00E1.00	-->	30 30 45 31 30 30
Check sum:		F9	-->	46 39
End character:				0D

Transmission to master: 0A 30 35 30 31 31 30 31 30 30 30 45 31 30 30 46 39 0D

11.2Transmission example, Instruction code 15 H

The device No.12 (zone address:01) should send the parameter group 0AH to the master.

Master to device (slave):	Dec.	Hex		ASCII (Hex)
Start character				0A
Device address:	12	0C	-->	30 43
Zone address:		01	-->	30 31
Instruction: Send parameter group		15	-->	31 35
Parameter group code (0AH):		0A	-->	30 41
Check sum:		D4	-->	44 34
End character:				0D

Transmission to device: 0A 30 43 30 31 31 35 30 41 44 34 0D

Device to master:	Dec.	Hex		ASCII
Start character				0A
Device address:	12	0C	-->	30 43
Zone address:		01	-->	30 31
Send parameter group (repeats instruction):		15	-->	31 35
1. Parameter code, process value:		10	-->	31 30
Parameter value	248	00F8.00	-->	30 30 46 38 30 30
2. Parameter code, actual setpoint:		20	-->	32 30
Parameter value	250	00FA.00	-->	30 30 46 41 30 30
3. Parameter code, actual output ratio:		60	-->	36 30
Parameter value	42	002A.00	-->	30 30 32 41 30 30
4. Parameter code, Status word 1:		70		37 30
Parameter value	00	0000.00	-->	30 30 30 30 30 30
Check sum:		C2	-->	43 32
End character:				0D

Transmission to master:

0A 30 43 30 31 31 35 31 30 30 30 46 38 30 30 32 30 30 30 46 41 30 30
36 30 30 30 32 41 30 30 37 30 30 30 30 30 30 30 43 32 0D

Please take attention:

The number and the sequence of the parameters which are transmitted, could be different.
It is dependent from the instruments type and the instruments configuration.
Therefore it is essential, that the transmitted parametercode will be evaluated and assigned to the actual value of the dependent parameter.

11.3Transmission example, Instruction code 20 H

The device No.27 (zone address: 01) gets the instruction:
"Take over the parameter 1 P (xp-heating, Parameter code: 40H) into the RAM.

Master to device (slave):	Dec.	Hex		ASCII
Start character:				0A
Device address:	27	1B	-->	31 42
Zone address:		01	-->	30 31
Instruction code:		20	-->	32 30
Parameter code (xp-heating):		40	-->	34 30
Parameter value:	5	0005.00	-->	30 30 30 35 30 30
Check sum:		7F		37 41
End character:				0D

Transmission to device: 0A 31 42 30 31 32 30 34 30 30 30 30 35 30 30 37 41 0D

Device to master:	Dec.	Hex		ASCII
Start character				0A
Device address:	27	1B	-->	31 42
Zone address:		01	-->	30 31
Instruction code (repeats instruction):		20	-->	32 30
response* (acknowledged) :		00	-->	30 30
Check sum:		C4	-->	43 34
End character:				0D

Transmission to master: 0A 31 42 30 31 32 30 30 30 43 34 0D

* If the device has understood the instruction issued by the master, it acknowledges with the response 00 H (acknowledge).

In the event of transmission or other errors, the device responds here with the appropriate error code.

11.4Transmission example, Instruction code 21 H

The device No.2 (zone address: 01) gets the instruction:
"Take over the parameter SP1 (Setpoint 1, Parameter code: 21H) and store power fail.

Master to device (slave):	Dec.	Hex		ASCII
Start character				0A
Device address:	2	02	-->	30 32
Zone address:		01	-->	30 31
Instruction code:		21	-->	32 31
Parameter code (Setpoint 1):		21	-->	32 31
Parameter value:	235	00EB.00	-->	30 30 45 42 30 30
Check sum:		D0	-->	44 30
End character:				0D

Transmission to device: 0A 30 32 30 31 32 31 32 31 30 30 45 42 30 30 44 30 0D

Device to master:	Dec.	Hex		ASCII
Start character				0A
Device address:	2	02	-->	30 32
Zone address:		01	-->	30 31
Instruction (repeats instruction) :		21	-->	32 31
Response * (acknowledged):		00	-->	30 30
Check sum:		DC	-->	44 43
End character:				0D

Transmission to master: 0A 30 32 30 31 32 31 30 30 44 43 0D

* If the device has understood the instruction issued by the master, it acknowledges with the response 00 H (acknowledge).

In the event of transmission or other errors, the device responds here with the appropriate error code.