Serial Communication of the Control2000-Firmware X.17

(First Edition)

Table of contents

1. In general

1.1 Serial Interfaces

This control device is available with two different interfaces:
- RS 232 (Standard)
- RS 485 (optionally via a Plug-in Card)

Communication is exclusively mediated via this interface. The active interface can be chosen by setting a parameter on the device. Rate of transmission and device address can be adjusted as individual parameters. Default settings are as follows:
- 1 start bit
- 8 data bit
- no parity
- 1 stop bit

Notice:
To accept changes of the parameters you have to leave the respective parameter menu.

The following functions can be executed over the interfaces:
- parameterisation
- process data acquisition

1.2 Data transmission

Data transmission is always initiated from the master. “STX” initiates the transmission. After that the data stream follows. The end of transmission is indicated by “DLE + ETX”. A correct transmission receives a “DLE” as acknowledgement from the recipient.

1.3 Protocol

1.3.1 Functional characters

Initiation of transfer	STX = 02 [h] start of Text
Positive receipt	DLE = 10 [h] data link Escape
Transfer end	ETX = 03 [h] end of Text
Negative receipt	NAK = 15 [h] negative acknowledgement from DLE

Notice: DLE as data content needs to be duplicated.

Sender		**Receiver**
STX	→	initiation of transmission
Data byte 1	→	data content
to	→	
Data byte n	→	
DLE	→	end of transmission
ETX	→	
	←	DLE

The data bytes comprise a protocol header and user data.

		←check sum→			←check sum →						
Byte #		1	2	3	4	.	.	.	n		
	STX	dev. add.	status	check sum	job	.	.	.	.	DLE	STX
		←protocol header →			←user data content →						

1.3.2 Header of protocol

Device address	“Slave” (possible range: 1...255)
Status	access mode: Read/Write, Text/Parameter and Error status
Checksum	Byte addition over the relevant data (see above)
Job number	Selection of the data set

1.3.3 User data

Parameters and process data are summarized in data blocks.

1.3.4 Valid status values and job numbers

status value	function	job number
00[h]	read parameters	00[h], 11[h], 12[h], 13[h], 14[h]
08[h]	read process data	05[h], 08[h], 80[h], FC[h]
10[h]	write process data	05[h], 08[h], 80[h], FC[h]
50[h]	read program	00[h] - 27[h]
60[h]	write program	00[h] - 27[h]
80[h]	write parameter	00[h], 11[h], 12[h], 13[h], 14[h]

Within a transmission string a max. break of 1 sec between single bytes is allowed. Longer breaks will be considered as an invalid transfer. There is no explicit error message. All acquired data will be neglected. The next transmission has to begin again with a “STX”.
If a transmission is recognized as failure (e.g. check sum is wrong or job number is invalid), an answer will be given in the protocol header and in the status implemented error type.

Slave → Master

		←check sum→			←check sum →		
Byte #		1	2	3	4		
	STX	dev. add.	status + error type	check sum	job#	DLE	STX
		←protocol header →					

1.3.5 Potential error types for Sio data transfer

2. data length overflow at reception

3. function dependent errors in the status bits 2^0 to 2^2

Error type:

0x01= wrong device address (only RS232) (not transmitted)

0x02=Check sum error (not transmitted)

0x03=unknown job

0x04=length of protocol wrong

0x05=wrong parameter block or wrong parameter value

0x06= wrong parameter index (text / program)

status Hex	function	job number	error
00	parameter read	block number too big	5
08	read process data	unknown job	3
10	write process data	wrong protocoll length	4
		[128] alarm not found	5
		[252] alarm not found	5
50	read program	wrong program block number	3
60	write program	wrong program number	5
		wrong profile block number	5
		wrong profile step	6
		wrong protocol length	4
		invalid time	5
		value range overflow	5
		unknown profile	6
80	write parameter	block number too big	3
		value range overflow	5
		wrong protocol length	4

1.4 Design of the data content

1.4.1 Overview of the protocol header

action of the master

Device address slave 1-255

Status	Access mode	
	read set of parameters	0x00
	write process parameters	0x80
	read process data	0x08
	write process data	0x10
	read programs	0x50
	write programs	0x60
Check sum	addition of bytes over the data content (without check sum) (only low byte of the sum)	
Job number	Selection of the data set	
User data	only in write operations parameter block (see table parameter data) or process data (see table process data) or 2 Byte text index + “text” + 0x00 (end mark).	

1.4.2 Answer from the slave

device address slave 1-255

Status (+error status) access mode as above

\+ error status

0x00=ok

0x01= wrong device address (only RS232)

0x02=Check sum error

0x03=unknown job

0x04=length of protocol wrong

0x05=wrong parameter block or wrong parameter value

0x06= wrong parameter index (text / program)

Check sum addition of bytes over the data content (without check sum)

(only low byte of the sum)

Job number selection of the data set

User data only in read operations

parameter / process content

programs:

2 Byte index

program number = 1-4 in 2^12-15

profile type = 0-4 in 2^8-11

0 = temperature

1 = humidity

2 = illumination

3 = ventilation

4 = switching contact

sentence number = 1-70 in 2^0-7

+2 Byte time (days *1440+ hh*60+mm)

+2 Byte target value

Text:

2 Byte index

+ Text

+ 0x00 (end mark)

Alarms:

status 00= no alarm + text index (2Byte)

01= acknowledged

02= not acknowledged

1.5 Overview protocol examples

address byte 0	status byte 1	check sum byte 2	job number byte 3	content byte 4-n
01	0x00 (para)	xx	01	temperature controller parameter
01	0x00 (para)	xx	02	humidity controller parameter
01	0x80 (para)	xx	01	temperature controller parameter
01	0x50 (read program)	xx	00-39 4 programs 5 profiles 2 halfs	0x1001 time target value (step 1) 0x1002 time target value (step 1) 0x1446 time target value (profile 5) (step 70)
01	0x60 (write program)	xx	00 program 1 1. profile	0x1001 time target value (step 1) 0x1023 time target value (step 35) (1. half of profile 1)
01	0x60 (write program)	xx	09 program 1 5. profile	0x1424 time target value (step 36) 0x1023 time target value (step 70) (2. half of profile 5)
01	0x08 (process)	xx	05	actual values and target value
01	0x08 (process)	xx	06	read actual values and controller target values
01	0x08 (process)	xx	0x80	read alarm status
01	0x10 (process)	xx	0x80	acknowledge alarm / delete global
01	0x08 (process)	xx	0x80 + 1-20	read alarm memory 1-20
01	0x10 (process)	xx	0x80 + 1-20	alarm memory 1-20 acknowledge / delete

1.6 General definitions

type	length in bytes
signed char	1
unsigned char	1
signed int	2
unsigned int	2

2 Read and write process data

2.1 Readout of internal RTC

process data Job 252 / # FC [h] (RTC-Job) - user data		
&rtc_weekday	weekday (0=Monday ...6=Sunday)	unsigned char
&rtc_hour	hour	unsigned char
&rtc_minute	minute	unsigned char
&rtc_second	second	unsigned char
&rtc_year	year	signed int
&rtc_month	month	unsigned char
&rtc_day	day	unsigned char

2.2 Example: read RTC Job 252 / status 0x08

PC sends	< 2 >	< 1 >	< 8 >	< 5 >	< 252 >	< 16 >	< 3 >
	STX	address	status 0x08	check sum	job	DLE	ETX
Control acknowledges	< 16 >						
	DLE						
Control sends	< 2 >	< 1 >	< 8 >	< 114 >	< 252 >	< 5 >	< 21 >
	STX						
	< 45 >	< 52 >	< 7 >	< 210 >	< 2 >	< 23 >	< 16 >
	User data						DLE
	< 3 >						
	ETX						
PC acknowledges	< 16 >						
	DLE						

Evaluation of user data

Value	Meaning	
5	weekday	5
21	hour	21
45	minute	45
52	second	52
7	year	2002
210		
2	month	2
23	day	23

Result: 21:45:52, Saturday, the 23.02.2002

2.3 Example: Adjusting the internal RTC

New date/time: 16:16:16, Monday, 25.02.2002

New user data

new data	meaning	value
0	weekday	0
16	hour	16
16	minute	16
16	second	16
2002	year	7
		210
2	month	2
25	day	25

Write RTC Job252 / Status 0x10

PC sends	< 2 >	< 1 >	< 16 > < 16 >	< 49 >	< 252 >	< 0 >	< 16 >< 16 >
	STX	address	status 0x10	check sum	job	user data	
	< 16 >< 16 >	< 16 >< 16 >	< 7 >	< 210 >	< 2 >	< 25 >	< 16 >
	User data						DLE
	< 3 >						
	ETX						
Control acknowledges	< 16 >						
	DLE						
Control sends	< 2 >	< 1 >	< 16 >< 16 >	< 13 >	< 252>	< 16 >	< 3 >
	STX	address	status 0x10	check sum	job	DLE	ETX
Pc acknowledges	< 16 >						
	DLE						

2.4 Error messages readout

Every error message can only be read out once. After return of the network, the pointer of the stack is set to the oldest, still remaining message.

process data job 128 / # 80 user data [h] (alarm job)		
&sio_alarm_buffer.year	year	signed int
&sio_alarm_buffer.month	month	unsigned char
&sio_alarm_buffer.day	day	unsigned char
&sio_alarm_buffer.hour	hour	unsigned char
&sio_alarm_buffer.minute	minute	unsigned char
&sio_alarm_buffer.second	second	unsigned char
&sio_alarm_buffer.text	text index (see table appendix A)	signed int
&sio_alarm_buffer.status	status	unsigned char
&sio_alarm_buffer.f_offset	offset	signed int

PC sends	< 2 >	< 1 >	< 8 >	< 137 >	< 128 >	< 16 >	< 3 >
	STX	address	status 0x00	check sum	job	DLE	ETX
Control acknowledges	< 16 >						
	DLE						
Control sends	< 2 >	< 1 >	< 8 >	< 46 >	< 128 >	< 7 >	< 210 >
	STX	address	status 0x08	check sum	job	user data	
	< 2 >	< 26 >	< 5 >	< 45 >	< 4 >	< 1 >	< 142 >
	user data						
	< 248 >	< 0 >	< 243 >	< 16 >	< 3 >		
	user data			DLE	ETX		
PC acknowledges	< 16 >						
	DLE						

Interpretation:

message (text index 1*256+142=398) on 26.02.2002 at 05:45:04

from appendix A → text index 398 → over temperature

Status of message: 248 →F8h

True for: Fxh → alarm active, Cxh → alarm acknowledged and 0xh → alarm deleted, where x=1 → notice, x=2 → small error, x=4 →heavy error and x=8 hardware error see also table appendix B.

PC sends	< 2 >	< 1 >	< 8 >	< 137 >	< 128 >	< 16 >	< 3 >
Control acknowledges	STX	address	status 0x08	check sum	job	DLE	ETX
Control acknowledges	< 16 > DLE						
Control sends	< 2 >	< 1 >	< 8 >	< 46 >	< 128>	< 16 >	< 3 >
	STX	address	status 0x08	check sum	job	user data	
PC acknowledges	< 16 > DLE						

Interpretation:

No user data → no actual message at the moment

2.5 Read actual value, job 5

process data job 5 / #5[h] - user data			
&temp1_actual	actual value temperature	[°C/10]	signed int
&temp1_target	target value temperature	[°C/10]	signed int
&humidity_actual	actual value humidity	[%r.H./10]	signed int
&humidity_target	target value humidity	[%r.H./10]	signed int
&temp2_actual	actual temperature cabinet sensor above	[°C/10]	signed int
&temp3_actual	actual temperature cabinet sensor underneath	[°C/10]	signed int
&conductivity_actual	actual conductivity	[microsiemens/10]	signed int
&illumination_target	target value illumination strength	[%]	signed int
&ventilator_target	target value ventilator	[%]	signed int
&input	door status (intern)	-	unsigned char
&out1	status switch exit 1 (intern)	-	unsigned char
&out2	status switch exit (intern)	-	unsigned char

2.6 Example: Actual value job 5 / status 0x08

PC sends	< 2 >	< 1 >	< 8 >	< 14 >	< 5 >	< 16 >	< 3 >
Control acknowledges	STX	address	status 0x08	check sum	job	DLE	ETX
Control acknowledges	< 16 > DLE						
Control sends	< 2 >	< 1 >	< 8 >	< 81 >	< 5>	< 4 >	< 179 >
	STX	address	status 0x08	check sum	job	user data	
	< 0 >	< 160 >	< 0 >	< 0 >	< 0 >	< 0 >	< 4 >
	user data						
	< 183 >	< 4 >	< 185 >	< 0 >	< 0 >	< 0 >	< 0 >
	user data						
	< 0 >	< 100 >	< 0 >	< 0 >	< 16 >< 16 >	< 16 >	< 3 >
	user data					DLE	ETX
PC acknowledges	< 16 > DLE						

value	meaning	
4	temperature actual value	120.3 °C
179		
0	temperature target value	16°C
160		
0	relative humidity actual value	0
0		
0	relative humidity target value	0
0		
4	over temperature sensor	120.7°C
183		
4	under temperature sensor	120.9°C
185		
0	conductivity	0 %
0		
0	illumination	0 %
0		
0	ventilation	100 %
100		
0	-	-
0	-	-
16	-	-

3 Read and write parameters

3.1 Read parameter block and target values

Parameter block target values Job 0 / # 00[h] user data		
&target_temperature	signed int	^C
&target_temperature_ramp	unsigned int	(°C/10)/minute
&target_humidity	unsigned char	%r.H.
&target_humidity_ramp	unsigned int	(%r.H./10)/minute
&target_illumintation	unsigned char	%
&target_ventilation	unsigned char	%
&target_power_outlet	unsigned char	-
&target_switch_contact	unsigned char	-

Example: parameter block read target values / status 0x00

PC sends	< 2 >	< 1 >	< 0>	< 1 >	< 0 >	< 16 >	< 3 >
	STX	address	status 0x08	check sum	job	DLE	ETX
Control acknowledges	< 16 >						
	DLE						
Control sends	< 2 >	< 1 >	< 0 >	< 243 >	< 0 >	< 0 >	< 30 >
	STX	address	status 0x08	check sum	job	user data	
	< 0 >	< 10 >	< 50 >	< 0 >	< 1 >	< 50 >	< 100 >
	user data						
	< 1 >	< 0 >	< 16 >	< 3 >			
	user data		DLE	ETX			
PC acknowledges	< 16 >						
	DLE						

Evaluation user data

data	meaning	interpretation
0	temperature target value [°C]	30
30		
0	temperature ramp	1
10	[(1/10°C)/minute]	
50	rel. humidity [%r.H.]	50
0	rel humidity ramp	0.1
1	[(1/10%r.H.)/minute]	
50	illumination [%]	50
100	50 <= ventilation [%]<= 100	100
1	power supply / 1=on 0= off	on
0	clock contact / 1=on 0=off	off

Example: parameter block write target values

User data

new target values	meaning	new data
-10	temperature target value [°C]	255
		246
0.5	temperature ramp	0
	[(1/10°C)/minute]	5
50	rel. humidity [%r.H.]	50
0.1	rel humidity ramp	0
	[(1/10%r.H.)/minute]	1
50	illumination [%]	50
100	50 <= ventilation [%]<= 100	100
off	power supply / 1=on 0= off	0
off	clock contact / 1=on 0=off	0

Example: parameter block write target parameters status 128 / status 0x80

PC sends	< 2 >	< 1 >	< 128 >	< 68 >	< 0 >	< 255 >	< 246 >
	STX	address	status 0x08	check sum	job		
	< 0 >	< 5 >	< 50 >	< 0 >	< 1 >	< 50 >	< 100 >
	user data						
	< 0 >	< 0 >	< 16 >	< 3 >			
	user data		DLE	ETX			
Control acknowledges	< 16 >						
	DLE						
Control sends	< 2 >	< 1 >	< 128 >	< 129 >	< 0 >	< 16 >	< 3 >
	STX	address	status 0x08	check sum	job	user data	
PC acknowledges	< 16 >						
	DLE						

new target values	meaning	new data
16	temperature target value [°C]	0
		16
1.6	temperature ramp [(1/10°C)/minute]	0
		16
50	rel. humidity [%r.H.]	50
0.1	rel humidity ramp [(1/10%r.H.)/minute]	0
		1
16	illumination [%]	16
100	50 <= ventilation [%]<= 100	100
off	power supply / 1=on 0= off	0
off	clock contact / 1=on 0=off	0

Example: write parameter block target values status 128 / status 0x80

PC sends	< 2 >	< 1 >	< 128 >	< 108 >	< 0 >	< 0 >	< 16 >< 16 >
	STX	address	status 0x08	check sum	job	user data	
	< 0 >	< 16 >< 16 >	< 50 >	< 0 >	< 1 >	< 50 >	< 100 >
	user data						
	< 1 >	< 1>	< 16 >	< 3 >			
	user data		DLE	ETX			
control acknowledges	< 16 >						
	DLE						
control sends	< 2 >	< 1 >	< 128 >	< 129 >	< 0 >	< 16 >	< 3 >
	STX	address	status 0x08	check sum	job	user data	
PC acknowledges	< 16 >						
	DLE						

3.2 Parameter block program 1

Parameter block 1 job 17 / # 11 [h] user data		
&prog1_typ	program type 0= day program 1= week program 2= process time 3= real time	unsigned char
&prog1_duration	1 to 14 days	unsigned char
&prog1_cycles	0= permanent or 1 to 100 cycles	unsigned char
&prog1_end	0= hold 1= end 2= controlled out 3= switch off 4= program 1 5= program 2 6= program 3 7= program 4	unsigned char
&prog1_end_signal	0= end signal off 1= end signal on	unsigned char

3.3 Parameter block program 2

Parameter block 2 job 18 / # 12 [h] - user data		
&prog2_typ	program type 0= day program 1= week program 2= process time 3= real time	unsigned char
&prog2_duration	1 to 14 days	unsigned char
&prog2_cycles	0= permanent or 1 to 100 cycles	unsigned char
&prog2_end	0= hold 1= end 2= controlled out 3= switch off 4= program 1 5= program 2 6= program 3 7= program 4	unsigned char
&prog2_end_signal	0= end signal off 1= end signal on	unsigned char

3.4 Parameter block program 3

Parameter block 1 job 17 / # 11 [h] user data		
&prog3_typ	program type 0= day program 1= week program 2= process time 3= real time	unsigned char
&prog3_duration	1 to 14 days	unsigned char
&prog3_cycles	0= permanent or 1 to 100 cycles	unsigned char
&prog3_end	0= hold 1= end 2= controlled out 3= switch off 4= program 1 5= program 2 6= program 3 7= program 4	unsigned char
&prog3_end_signal	0= end signal off 1= end signal on	unsigned char

3.5 Parameter block program 4

Parameter block 1 job 17 / # 11 [h] user data		
&prog4_typ	program type 0= day program 1= week program 2= process time 3= real time	unsigned char
&prog4_duration	1 to 14 days	unsigned char
&prog4_cycles	0= permanent or 1 to 100 cycles	unsigned char
&prog4_end	0= hold 1= end 2= controlled out 3= switch off 4= program 1 5= program 2 6= program 3 7= program 4	unsigned char
&prog4_end_signal	0= end signal off 1= end signal on	unsigned char

Example: read parameter program 1 / status 0x00

PC sends	< 2 >	< 1 >	< 0 >	< 18 >	< 17 >	< 16 >	< 3 >
	STX	address	status 0x00	check sum	job	DLE	ETX
control acknowledges	< 16 >						
	DLE						
control sends	< 2 >	< 1 >	< 0 >	< 142 >	< 17 >	< 2 >	< 14 >
	STX	address	status 0x00	check sum	job	user data	
	< 100 >	< 7 >	< 1 >	< 16 >	< 3 >		
	user data			DLE	ETX		
PC acknowledges	< 16 >						
	DLE						

Evaluation user data

value	meaning
2	process time program
14	duration 14 days
100	100 cycles
7	continue with program 4
1	end signal on

Example: read parameter program 2 / status 0x00

PC sends	< 2 >	< 1 >	< 0 >	< 19 >	< 18 >	< 16 >	< 3 >
	STX	address	status 0x00	check sum	job	DLE	ETX
control acknowledges	< 16 >						
	DLE						
control sends	< 2 >	< 1 >	< 0 >	< 29 >	< 18 >	< 3 >	< 1 >
	STX	address	status 0x00	check sum	job	user data	
	< 3 >	< 2 >	< 1 >	< 16 >	< 3 >		
	user data			DLE	ETX		
PC acknowledges	< 16 >						
	DLE						

Evaluation user data

value	meaning
3	real time program
1	duration 1 day
3	3 cycles
2	controlled off
1	end signal on

Example write parameter program 4 / status 0x80

Status 0x80

Controller address 12

job 20 → parameter program 4

User data

value	meaning
0	day program
1	always duration 1 day for day program
0	always 0 for day program
0	always 0 for day program
0	always 0 for day program

Example: read parameter program 2 / status 0x00

PC sends	< 2 >	< 12 >	< 128 >	< 161 >	< 20 >	< 0 >	< 1 >
	STX	address	status 0x80	check sum	job	user data	
	< 0 >	< 0 >	< 0>	< 16 >	< 3>		
	user data			DLE	ETX		
control acknowledges	< 16 >						
	DLE						
control sends	< 2 >	< 12>	< 128 >	< 160 >	< 20 >	< 16 >	< 3 >
	STX	address	status 0x00	check sum	job	DLE	ETX
PC acknowledges	< 16 >						
	DLE						

4 Read and write program profile

4.1 Program 1 - profile 1 (temperature) - read first half

PC sends	< 2 >	< 1 >	< 80 >	< 81 >	< 0 >	< 16 >	< 3 >
	STX	Address	status 0x50	check sum	job	DLE	ETX
Control acknowledges	< 16 >						
	DLE						
Control sends	< 2 >	< 1>	< 80 >	< 203 >	< 0 >	< 16 >	< 3 >
	STX	Address	status 0x50	check sum	job	DLE	ETX
	< 16 >< 16 >	< 1 >	< 1 >	< 104 >	< 0 >	< 10 >	
	user data						
	< 16 >< 16 >	< 2 >	< 1 >	< 164 >	< 0 >	< 30 >	
	user data						
	< 16 >< 16 >	< 3 >	< 4 >	< 56 >	< 0 >	< 30 >	
	user data						
	< 16 >< 16 >	< 4 >	< 4 >	< 116 >	< 0 >	< 10 >	
	user data						
	< 16 >< 16 >	< 5 >	< 255 >	< 255 >	< 0 >	< 0 >	
	user data						
	< 16 >< 16 >	< 6 >	< 255 >	< 255 >	< 0 >	< 0 >	
	user data						
	.						
	.						
	< 16 >< 16 >	< 34 >	< 255 >	< 255 >	< 0 >	< 0 >	
	user data						
	< 16 >< 16 >	< 34 >	< 255 >	< 255 >	< 0 >	< 0 >	
	user data						
	< 16 >	< 3 >					
	DLE	ETX					
PC acknowledges	< 16 >						
	DLE						

Evalution of user data

value	meaning	
16	program number, profile, x. half	program number 1 profile 1, 1. half
1	program step	1
1	time in minutes	1*256+256+104=360min=0.6:00
104		
0	target value	10°C
10		
16	program number, profile, x. half	program number 1, profile, 1.half
2	program step	2
1	time in minutes	1*256+164=420min=07:00
164		
0	target value	10°C
10		
16	program number, profile, x. half	program number 1, profile, 1.half
3	program step	3
4	time in minutes	4*256+56 =1080 min= 18:00
56		
0	target value	10°C
10		
16	program number, profile, x. half	program number, profile 1, 1.half
4	program step	4
4	time in minutes	4*256+116= 1140min = 19:00
116		
0	target value	10°C
10		
16	program number, profile, x. half	program number 1, profile 1, 1. half
5	program step	5
255	time in minutes	not filled
255		
0	target value	
0		
. .	. .	. .
16	program number, profile, x. half	program number 1, profile 1, 1. half

1	program step	35
255	time in minutes	not filled
255		
0	target value	
0		

Every non filled step consists of a correct address of the program, the profile and the profile half. The time step 0xFFFF shows that this step is not filled. The target value for empty steps has to be written 0x0000. The program steps are to be filled in continuous succession.

The temperature profile from program 1 looks like the following:

program 1 temperature profile			
number	day	time	target value
1	1	06:00	10°C
2	1	07:00	30°C
3	1	18:00	30°C
4	!	19:00	10°C

4.2 Important notes for reading and writing of programs

All programs consist of 5 profiles in two halves. A program always has to be completely written, even if it only consists of only a few steps. The length of a program is independent of the apparatus.

The blocks of a program always have to be in the correct succession. In case of an error, one has to begin from the start. In fact, the individual blocks could be read individually, but the writing cannot be chosen freely.

In addition, every program contains a parameter block in which the succession and the “end-behaviour” is documented.

The programmer has always to make sure through back reading of the written programs and parameters that all parameters and profiles are correctly transferred.

It is highly recommended to start with the reading and interpretation of the programs to understand the structure of the programs correctly.

Appendix A- possible error messages (firmware X.17)

Index	Text	
	Service	
106	Interval cabinet	notice
107	Interval cold machine	notice
108	Interval illumination	notice
109	Interval illumination	notice
	Alarms	
136	temperature pre-alarm	
137	temperature main alarm	
138	humidity pre alarm	
139	humidity main alarm	
140	conductivity pre alarm	
141	conductivity main alarm	
281	sensor temperature 1	error
282	sensor temperature 2	error
283	sensor temperature 3	error
284	sensor humidity	
285	sensor conductivity	
286	cold: no pressure	error
287	cold: pressure too high	error
288	door open	notice
289	emergency exit	error
290	emergency program	
291	water low	
292	water bad	
293	temperature too high	
294	temperature too low	
295	ventilation	error
296	humidity too high	
297	humidity too low	
	Program status	
298	pre-selection program 1	notice
299	pre-selection program 2	notice
300	pre-selection program 3	notice
301	pre-selection program 4	notice
302	start program 1	notice
303	start program 2	notice
304	start program 3	notice
305	start program 4	notice

306	end program 1	notice
307	end program 2	notice
308	end program 3	notice
309	end program 4	notice
398	over temperature	alarm
399	danger of icing	
400	refill time	
401	plug	
402	pump	
403	plug	
404	Ext. sensor	

Appendix B- potential combinations of the status byte (firmware X.17)

Meaning of status byte during alarm messages		
decimal	hex	meaning
0	00h	message deleted
191	C1h	notice acknowledged
194	C2h	small error, acknowledged
196	C4h	heavy error, acknowledged
200	C8h	hardware error, acknowledged
241	F1h	notice, new message
242	F2h	small
244	F4h	heavy error, new message
248	F8h	hardware error, new message