LIEBHERR

SUFsg 5001 / SUFsg 7001

Documentation Modbus TCP/IP

Technical Specification

7 March 2022 Liebherr Appliance Division

Update 15 May 2023 (add chapter MOD BUS Common Description)

Table of contents

Common Description

The **Liebherr SUFsg 5001/7001 models are equipped with a LAN interface that uses the MOD Bus protocol,** even when TCP/IP transmission is used. The MOD Bus protocol **is tunneled through the TCP/IP.**
The result is that a programmer has to implement the MOD Bus protocol and the TCP/IP tunnel.
This document describes both actions.

Legal Notice

This document is available free of charge for you. It enables you to develop your own software. This is the complete information regarding interfaces. Please keep in mind that **Liebherr does not offer any support further than this document**.
The information is provided as it is, without warranty of any kind. It could contain wrong information**.**
Integrating the devices into your own system via Modbus is done at **your own risk**.

Liebherr cannot be made liable for any damage caused by integration of the appliances into external systems.

Liebherr always suggests to **use the existing Liebherr SmartMonitoring system** as a ready to use monitoring and datalogger system or to integrate appliances in an existing network via the alarm-relay and the 4-20mA analogue output.

Do not share this document outside of your organization without the permission of Liebherr!

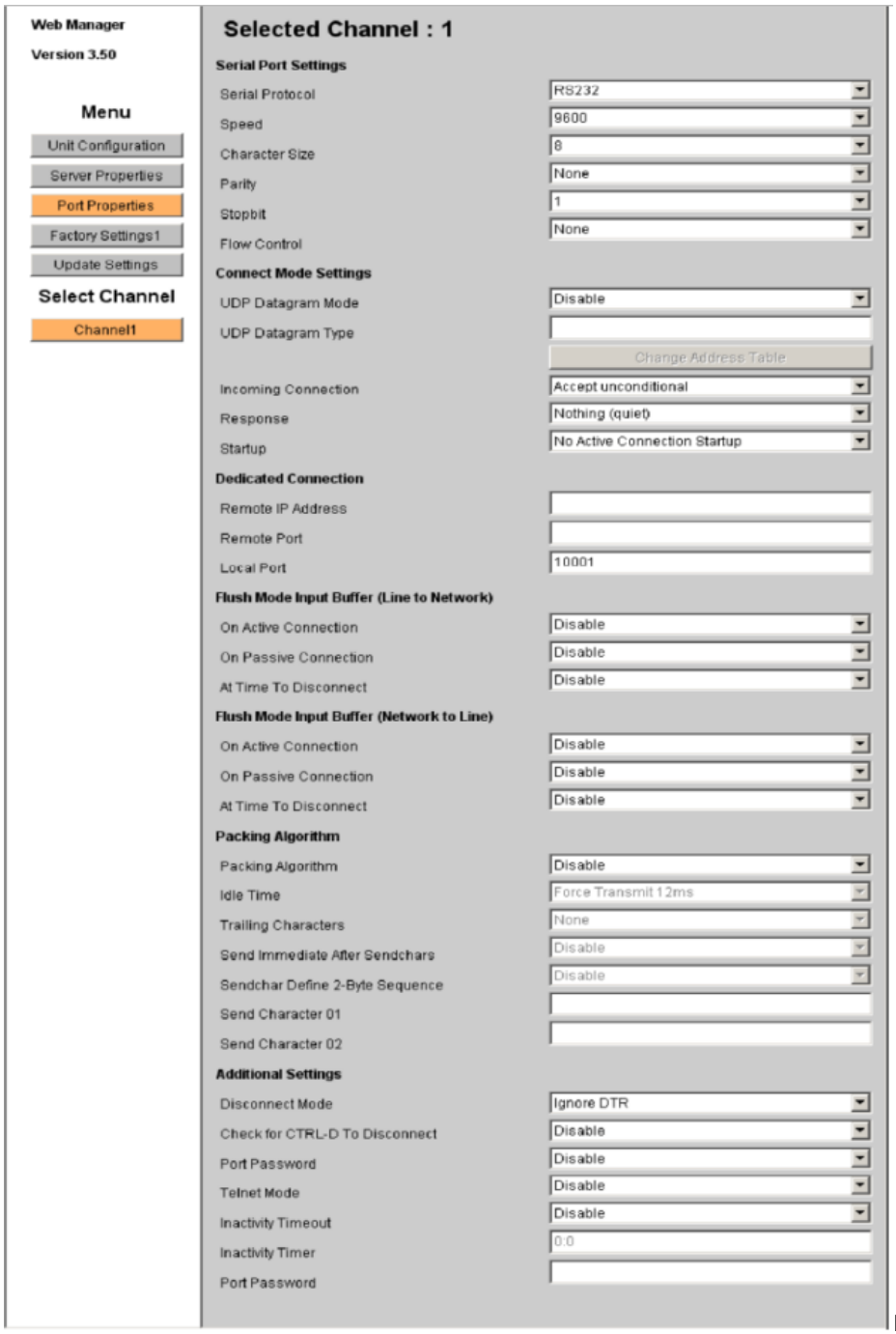
Network Specification

TCP/IP common description (Lantronix XPort)

This description applies to the controllers including the Ethernet interface based on the
XPort-03 Module of Lantronix Inc. This document is only an overview, which describes how the chambers use the XPort services. Some tools and a complete documentation can be downloaded from
www.lantronix.com
You can download the Lantronix Tool “Device Installer” as well from the Homepage of Lantronix.

Protocol Description

If nothing else is described in this document, the XPort Module is used with factory default settings. (see picture below)
Note:
The description of MODBus-Protocol is needed for a successful connection to an appliance-controller. The MODBus address at the appliance-controller should always be set to 1 (default value).
Pressing the button “Factory Settings 1” sets the XPort always back to the configuration needed to communicate internally with the appliance-controllers.



Internet Browser Java Config.

Network Interface

RJ45 Ethernet 10BASE-T or 100BASE-TX (auto-sensing)
Compatibility
Ethernet: Version 2.0/IEEE 802.3

Used TCP/UDP protocols and ports (factory settings)

Port	Protocol	Used as
TCP Port 80	http	webserver for configurations
TCP Port 9999	telnet	telnet for configurations not implemented in webserver
TCP Port 10001	Raw binary	Serial MODBus data tunnel used for communication with Liebherr chambers
UDP port 30718		With IP address = 255.255.255.255 used for detecting XPort-03 devices in the network
UDP Port 68	Dhcp	DHCP-Client

Detecting XPort-03 devices in the network (UDP)

For detecting all XPort-03 (BINDER Chamber) a UDP Broadcast (IP=255.255.255.255) on port 30718 has to be send. Every XPort available answers with 30 Byte of data.

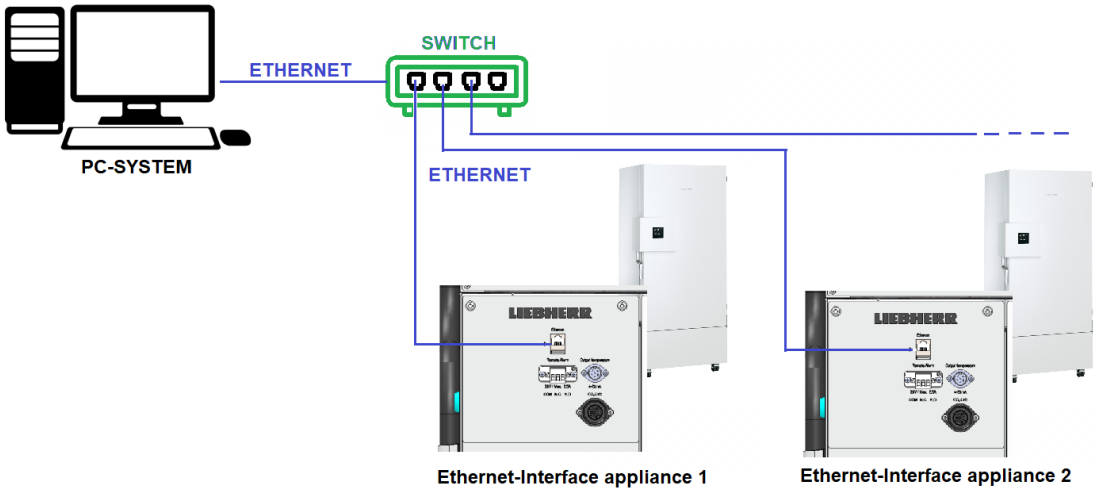
The IP of the XPort can be seen in the returning frame under Internet Protocol -> Source
The last 8 bytes of the data include the MAC address of the XPort.

Return data example (hex bytes):
..... 00 20 4A 8A A0 F7
MAC = 00-02-4A-8A-A0-F7

The MAC is labeled on the chamber near the Ethernet port. With the MAC it is possible to identify the answering chambers.

Connecting a chamber via TCP/IP

To read and write data, port 10001 is used as a TCP/IP tunnel. This means the MODBus protocol is tunneled in RAW binaries through the XPort device. Only the MODBus telegram is sent to the controller.



Assigning a fix IP Address if no DHCP is available

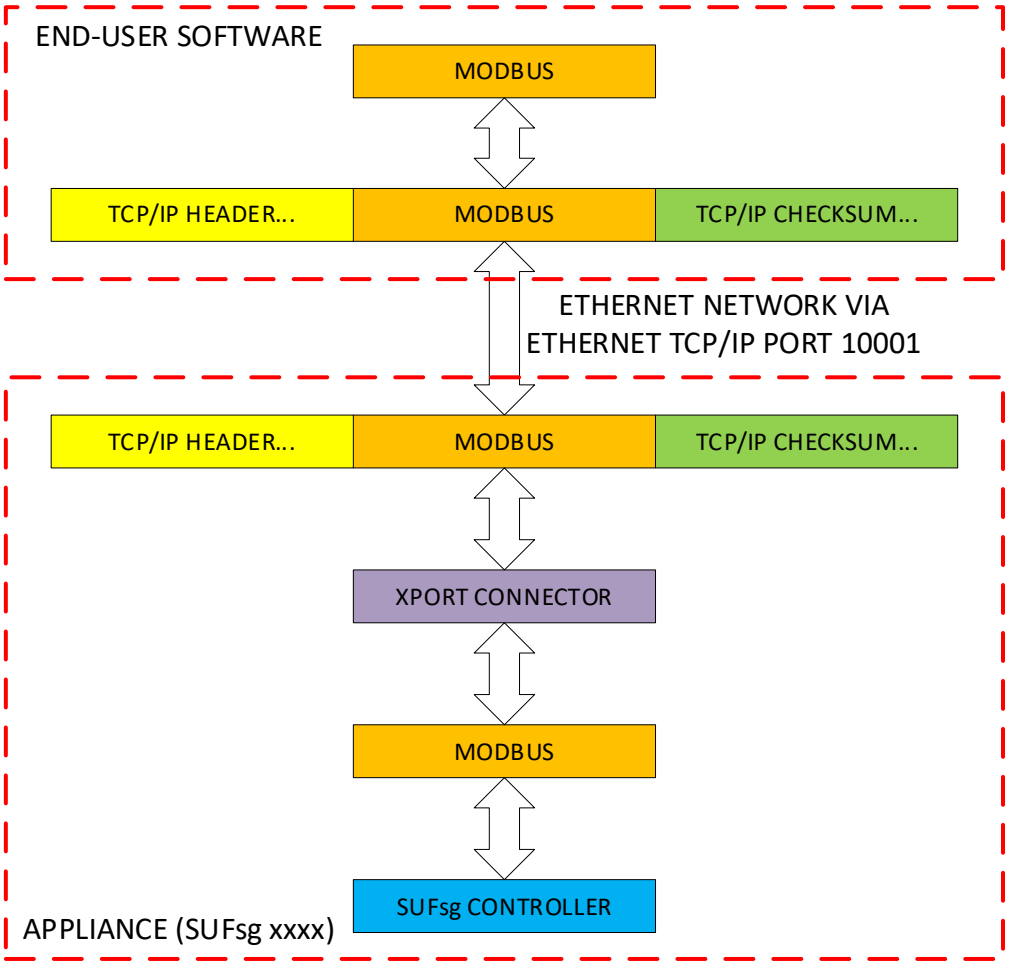
The unit's IP address must be configured before it can work correctly on a network. You have several options for assigning an IP to your unit. E.g. you manually assign the IP address via the network using Lantronix Device Installer software.

The unit's IP address is normally set to 0.0.0.0 at the factory. Information how to read-out and set the network settings of the appliance can be found within the appliances' manual. The appliance is DHCP enabled as the default.

To manually assign an IP address:
1. Click Start > Programs > Lantronix > DeviceInstaller > DeviceInstaller. If your PC has more than one network adapter, a message is displayed. Select an adapter and click OK.
Note: If the unit already has an IP address (e.g., DHCP has assigned an IP address), click the Search icon and select the unit from the list of Lantronix device servers on the local network.
2. Click the Assign IP icon
3. If prompted, enter the hardware address and click "Next".
4. Select "Assign a specific IP address" and click "Next".
5. Enter the IP address. The Subnet mask displayed automatically based on the IP address; if desired, you may change it. On a local network, you can leave the Default gateway blank (all zero). Click Next.
6. Click the Assign pushbutton and wait several seconds until a confirmation message is displayed. Click Finish.

Communication with the appliance

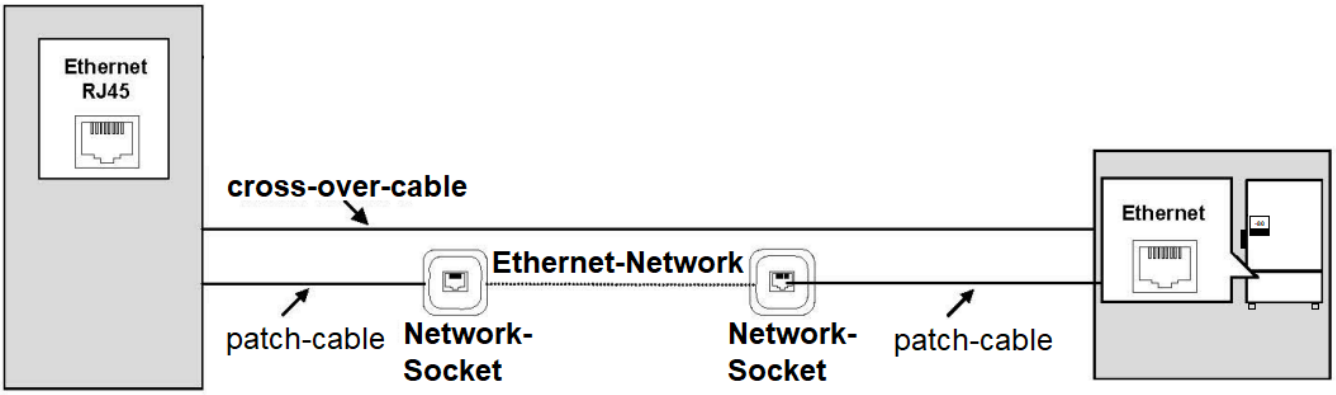
TCP/IP is used to tunnel the MOD Bus telegram through the Ethernet network connection. The default port is 10001.
The controller can only answer 1 request at a time, i.e. several requests may not be sent in parallel.



MOD BUS Common Description

Physical

The interface of the controller(s) and the data logger (PC) is galvanically isolated from signals and supply and supports up to 31 Slaves (Chambers) and one Master. The system uses a 4 wire full duplex bus which is used only in half duplex mode. This means there is only a request or a response at one time.



Master-slave principle

The communication between a PC (master) and an instrument (slave) using MOD Bus takes place according to the master-slave principle in the form of data request/instruction - response. The master controls the data exchange; the slaves only have a response function. They are identified by their instrument addresses. A maximum of 255 addresses can be accessed.

Transmission mode (RTU)

The transmission mode used is the RTU mode (Remote Terminal Unit). Data is transmitted in binary form (hexadecimal) with 8 bits, 16 bits for integers, and 32 bits for floating values. The LSB (least significant bit) is transmitted first. The ASCII operating mode is not supported. Data format The data format describes the arrangement of a byte transmitted.

The data format is as follows:

Baud rate:	9600 bits/sec
Data word:	8 bit
Parity bit:	none
Stop bit:	1

Instrument address

The address of the slaves can be set between 1 and 255. Address 0 is reserved.
Possibilities of data exchange
Query Data request/instruction from the master to a slave via the appropriate address. The slave accessed then responds.
Broadcast Instruction from the master to all slaves via address 0. The slaves in the system do not respond. A data request is not appropriate in this case.
For example, a certain set point can be transmitted to all slaves. The correct acceptance of the value by the slaves should in that case be checked by a subsequent read-out of the set point.

Timing of the communication

Start and end of a data block are identified by transmission pauses. The maximum permitted interval between two consecutive characters is three times the time for transmitting a character.
The character transmission time (time for transmitting a character) depends on the baud rate and on the data format used.

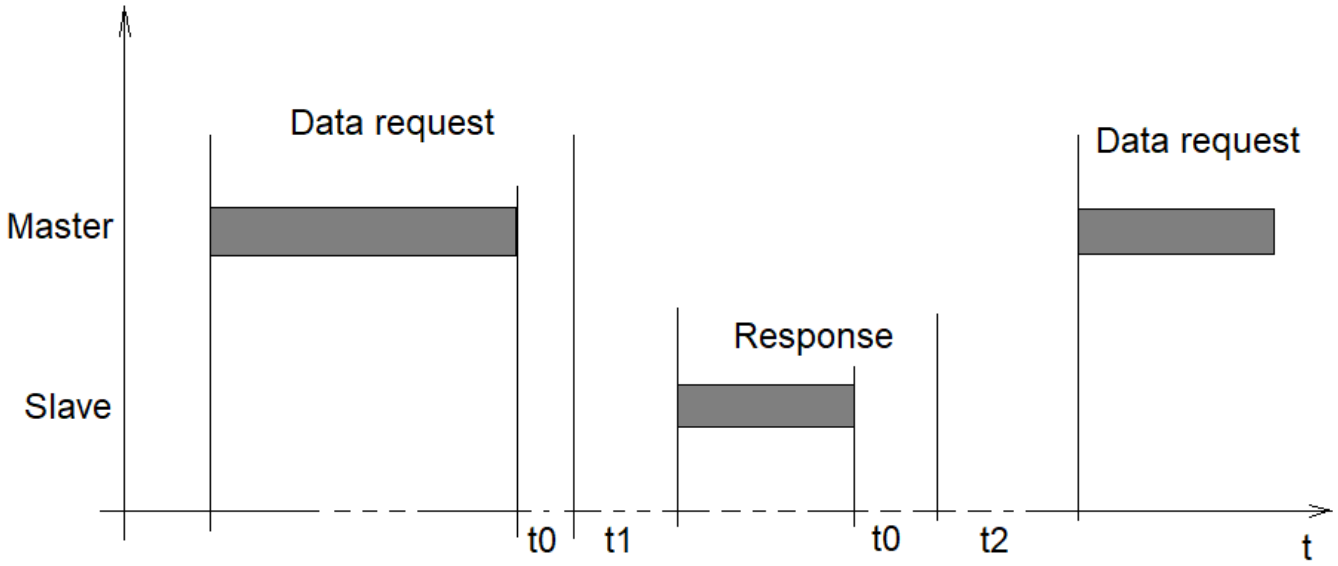
Character transmission time [us] = 937,5
A maximum of 31 slaves can be accessed via the RS422 interface.

Data request from master
Transmission time = n characters * 1000 * x bits/baud rate
Marker for data request end
3 characters * 1000 * x bits/baud rate = 2.813 ms
Processing the data request by the slave (200 ms max.)

Response of the slave
Transmission time = n characters * 1000 * x bits/baud rate
Marker for response end
3 characters * 1000 * x bits/baud rate = 2.813 ms

Timing scheme

A data request runs according to the following timing scheme:



t0 = end marker = 3 characters = 2.813 ms
t1 = 5 ms ... 250 ms
t2 >= 10 ms

Communication during the internal processing time of the slave

The master must not make any data requests during the internal processing time of the slave. Any data requests during this period are ignored by the slave.

Communication during the response time of the slave

The master must not make any data requests during the response time of the slave. Any data requests during this period cause all data currently on the bus to become invalid.

Arrangement of the data blocks

All data blocks have the same structure:
Data structure

slave address	function code	data field	Checksum CRC16
1 byte	1byte	X byte(s)	2 bytes

Each data block consists of four fields:
Slave address instrument address of a particular slave
Function code function selection (read, write, bit, word)
Data field
-contains the information:
-bit address (word address)
-bit number (word number)
-bit value (word value)
Checksum recognition of transmission errors

Error treatment

Error codes There are five error codes:
1. invalid function
2. invalid parameter address
3. parameter value outside range of values
4. slave not ready
5. write access to parameter denied

Response in case of error

Slave address	Function xx or 80h	Error code	Checksum CRC16
1 byte	1 byte	1 byte	2 bytes

The function code is linked by OR with 0x80, i.e. the MSB (most significant bit) is set to 1.
Example
Data request:

01	02	00	00	00	00	CRC16

Response:

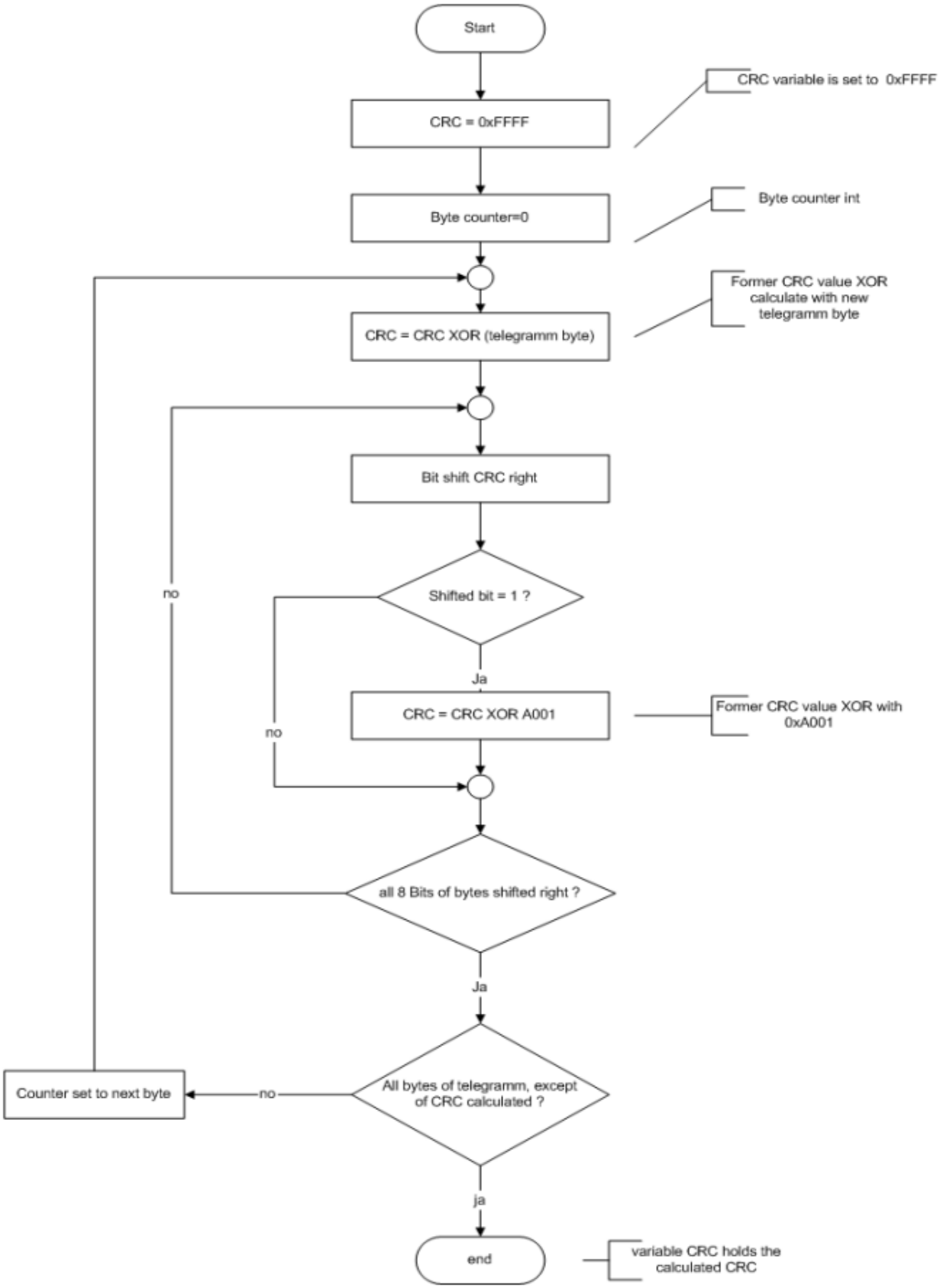
01	82	01	CRC16

Special cases The slave does not respond to the following errors:
- The checksum (CRC16) is incorrect
- The instruction of the master is incomplete or over-defined
- The number of the words or bits to be read is zero
Error code 4 (slave not ready) is not implemented in the controller since the controller always responds within 250 ms to a valid data request.

Checksum (CRC16)

The checksum (CRC16) serves to recognize transmission errors. If an error is identified during processing, the appropriate instrument does not respond.

Calculation scheme



Example 1

Data request: reading two words from address 1 (CRC16 = 0x0E97)

14	03	00	01	00	02	97	0E

Response: (CRC16 = 0x953E)

14	03	04	03	E8	01	F4	3E	95
			Word 1		Word 2			

Example 2

Instruction: Set bit on address 24 (CRC16 = 0xF80E)

14	05	00	18	FF	00	0E	F8

Response (as instruction):

14	05	00	18	FF	00	0E	F8

Functions

The following functions are available to the controller:

Function number	**Function**
0x03 or 0x04	Reading n words (max. 80 words)
0x06	Writing one word
0x10	Writing n words (max. 80 words)

Reading more than one words

This function reads n words from a defined address.

Data request

Slave address	Function 0x03 or 0x04	Address first word	Number of words	Checksum CRC16
1 byte	1 byte	2 bytes	2 bytes	2 bytes

Response

Slave address	Function 0x03 or 0x04	Number of bytes read	Word value(s)	Checksum CRC16
1 byte	1 byte	1 byte	X byte(s)	2 bytes

Example:

Reading the four floats form base address 0x0037

Word address= 0x0037

Data request:

14	03	00	37	00	08	CRC16

Response:

14	03	10	1999	4348	4CCC	4348	2666	4396	F333	43CA	CRC
			Signal1 200.1		Signal2 200.3		Signal3 300.3		Signal4 405.9		

Writing one word

In the “writing word” function the data blocks for instruction and response are identical.

Instruction

Slave address	Function 0x06	Word address	Word value	Checksum CRC16
1 byte	1 byte	2 bytes	2 bytes	2 bytes

Response

Slave address	Function 0x06	Word address	Word value	Checksum CRC16
1 byte	1 byte	2 bytes	2 bytes	2 bytes

Example Write base address 0x0168 value 100

Instruction:

14	06	01	68	00	64	CRC16

Response (as instruction):

14	06	01	68	00	64	CRC16

Writing more than one words

Instruction

Slave address	Function 0x10	Address first word	Number of words	Number of bytes	Word value(s)	Checksum CRC16
1 byte	1 byte	2 bytes	2 bytes	1 byte	X byte(s)	2 bytes

Response

Slave address	Function 0x10	Address first word	Number of words	Checksum CRC16
1 byte	1 byte	2 bytes	2 bytes	2 bytes

Example:
Write value = 0.66 to base address 0x010F
Instruction:

14	10	01	0F	00	02	04	F5	C2	3F	28	CRC16

Response:

14	10	01	0F	00	02	CRC16

Base address tables

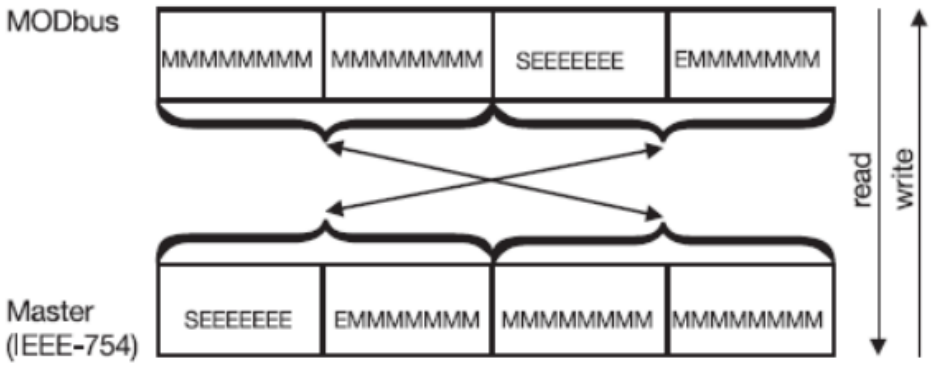
How to read address tables

All process values (variables) together with their addresses, the data type and the access mode are described below.
References are as follows:

R/O access reading only
R/W access reading and writing
char ASCII character (8 bits)
byte byte (8 bits)
int integer (16 bits)
char xx character chain of length xx;
xx = length including chain and character /0
bit x bit No. x
long long integer (4 byte)
int10 int10 Value / 10 = Value

float floating value (4 bytes)
The explanations below apply on condition that the master operates in the IEEE-754 format. Before transmitting a value the byte sequence has to be changed so that it corresponds to the presentation for MODBus (see diagram).

M- 23 bit normalized mantissa
E - exponent (complement base 2)
S - sign bit; 1 = negative; 0 = positive



Example:
Transmitting the decimal value 550:
MODBus: 0x80, 0x00, 0x44, 0x09

TCP/IP COMMUNICATION WITH THE CONTROLLER

UDP General

Always Port 9001
All read and write telegrams must end with CRLF (0x0D0A). Otherwise CRLF must not appear in a telegram.
Appliance Controller Type: T4.x or RD4.

Detecting the Ethernet appliances in the Network (UDP)

Communication:
1. Chamber detection (UDP IP:255.255.255.255):
2. Computer sends "GetConfigDirect" (ASCII format)
3. Appliance answers with chamber information (Timeout:3000ms + max. 500 characters per answer):

"IP:" + strIP + " Port:" + strTCPPort +" MAC:" + strMAC + TAB +
"Chambertype:strChamberType" + TAB +
"Controllertype:strControllerType" + TAB +
"COMType:BCWDirect" + TAB +
"ProgramController:strProgramController";

Explanation of the C# Notation:

strIP = IP address of the chamber as string (xxx.xxx.xxx.xxx)
strTCPPort = UDP Port number decimal as string
strMAC = "xx-xx-xx-xx-xx-xx" as string
strChamberName = Chamber (default), otherwise customer's setting on the T4.x (max 40 characters)
strChamberType = Examples: "CB 160 (E6)", "KT 115 (E6)"... (max 250 characters, no special characters)
strControllerType = "T4.2", "T4.1" .... (max 250 characters and ".", no special characters)
strProgramController = "True" or "False"
TAB = ASCII for [TAB]
CRLF = ASCII for [CR][LF]

TCP General

Always Port 9000.
If writing the value, the value returns as data (corrected, if appropriate)

Connecting an appliance via TCP – general

All read and write telegrams must end with CRLF (0x0D0A), in either direction.
Connection established by computer through port 9000; 4 trials each with a 1000 ms timeout

1. Computer sends command to appliance (e.g. communication test):
"CANIDGetValue:10010010"+CRLF); timeout:1000 ms
2. T4.x transmits byte counts of data to the computer; timeout 1000ms
3. T4.x transmits data to the computer; timeout 1000ms
Before appliance sends data, it transmits a telegram with the data byte counts (decimal , max 100 characters) to the computer. This telegram always must end with CRLR.

Structure:
For transmission, the maximum byte count is100 characters (e.g. 23789 Bytes = 5 characters)
All telegrams have the structure "command:command value“. They are separated by the colon.

If a value is included in the telegram, it is also separated by a colon:
"command:command value:value"
The answer is always "command:command value". If a value string is included, it is separated by a colon.
The appliance answers in case of an error (any error) with the key word "Error" as command.

CANID information:
The computer can request information of CANIDs (ID0xYYYYYYYY, with “Y” meaning CANID in HEX)
Example: CANID ID0x11800080 (Actual temperature value of the appliance)
1. Computer sends to appliance: "BCWDirectGetCANIDInfo:ID0x11800080"
2. Appliance answers to computer:
"BCWDirectGetCANIDInfo:ID0x11800080:"
+ "Befehl=ID0x11800080" + CRLF
+ "HexAdress=11800080" + CRLF
+ "VarType=float" + CRLF
+ "ReadWords=" + CRLF
+ "ConstValue=" + CRLF
+ "LimitLow=123.0" + CRLF
+ "LimitHigh=600.0" + CRLF
+"ReadWrite=RO" + CRLF
+ "Doku=P1_ChamberTemperature" + CRLF
+ "Unit=1"

When using the freeware “Wireshark” this will be the Raw Data of the answer:
42 65 66 65 68 6c 3d 49 44 30 Befehl=ID0
78 31 31 38 30 30 30 38 30 0d 0a 48 65 78 41 64 x11800080..HexAd
72 65 73 73 3d 31 31 38 30 30 30 38 30 0d 0a 56 ress=11800080..V
61 72 54 79 70 65 3d 66 6c 6f 61 74 0d 0a 52 65 arType=float..Re
61 64 57 6f 72 64 73 3d 0d 0a 43 6f 6e 73 74 56 adWords=..ConstV
61 6c 75 65 3d 0d 0a 4c 69 6d 69 74 4c 6f 77 3d alue=..LimitLow=
31 32 33 2e 30 0d 0a 4c 69 6d 69 74 48 69 67 68 123.0..LimitHigh
3d 36 30 30 2e 30 0d 0a 52 65 61 64 57 72 69 74 =600.0..ReadWrit
65 3d 52 4f 0d 0a 44 6f 6b 75 3d 50 31 5f 43 68 e=RO..Doku=P1_Ch
61 6d 62 65 72 54 65 6d 70 65 72 61 74 75 72 65 amberTemperature
0d 0a 55 6e 69 74 3d 31 0d 0a ..Unit=1..

TCP communication test

1. Computer sends to appliance: "CANIDGetValue:10010010"
2. If OK, appliance answers to computer:
"CANIDGetValue:10010010:Communication OK".
Otherwise (in case of any error) the appliance answers with the key word "Error".

Reading a CAN ID value via TCP

1. Computer sends to appliance: "CANIDGetValue:CANIDin32BitHex"
2. If OK, appliance answers to computer:
“CANIDGetValue:CANID in 32Bit Hex:value in string format”.
Otherwise (in case of any error) appliance answers with the key word "Error".

Writing a CANID value via TCP

1. Computer sends to appliance: "CANIDWriteValue:CANID in 32Bit Hex:value in internal format"
2. If OK, appliance answers to computer:
"CANIDWriteValue:CANID in 32Bit Hex:value in internal format "
Otherwise (in case of any error) appliance answers with the key word "Error".

Types of variables (internal format)

Data Format	Representation (always 32bit)
Unsignedbyte	only LSB evaluated AND mask =0x000000FF
Signedbyte	only LSB evaluated AND mask =0x000000FF
Float	acc. to IEEE 754
Longsignedint	-2147483648 up to 2147483647
Longunsignedint	0 up to 4294967295
Binary	00000000 00000000 00000000 00000000 up to 11111111 11111111 11111111 11111111
Binary16	00000000 00000000 00000000 00000000 up to 00000000 00000000 11111111 11111111
Hex	0000 0000 up to FFFF FFFF (longunsignedint)
Hex16	0000 0000 up to 0000 FFFF (unsignedint)
Boolean	LSB is decisive (AND mask 0x00000001)
Enum	see _unsignedint
String	Maximum 40 characters
Date	Date: always yyyymmdd in longsignedint as a decimal value (yyyymmdd). Example: 23.03.2007 = 2007 = year, 03 = month, 23 = day. Results in 20070323 in the telegram.
Timeabs	Time: hhmmsscc, 2 characters each for hours, minutes, seconds, and milliseconds as a decimal value (longunsignedint) 8 characters include milliseconds. When 6 characters, the milliseconds are set to zero. Internal format: always 24 hours. Example: 8 characters: 23:45:38,83 = 23453883 Example: 6 characters: 23:45:38,00 = 234538
Version	xxxxyyzz one Hex value per Byte (longunsignedint) Version numbers for hardware and software xxxx = Byte: Main version numbers 0x0001 up to 0xFFFF (0x0000 = reserved = no information) yy = Byte subversion (0x00 = Beta version, for development only) zz = Byte Bugfix counter
Seconds	Count of seconds (= longunsignedint)
Signedint	AND mask = 0x0000 FFFF
Unsignedint	AND mask = 0x0000 FFFF
Partno	Part number xxxx-xxxx e.g. part number = 5014-01A7 > Decimal value = 1343488423
CANID	Contains a CANID (29 bit) AND mask 1FFF FFFF
Int16Div10	16 Bit Integer (value divided by 10 results in real value)

Units

Parameter	Unit	Unit:Unit number
None	None	0 (default)
Temperature	Kelvin	1
Percent	None	2
Time Relative	Seconds	3
Pressure	mbar	4
Light	SI unit	5
Voltage	Volt	6
Current	Ampere	7
Power	Watt	8
Volume Percent	None	9
Temperature Delta	Kelvin	10
Percent Delta	-	11

CAN ID Table

Value	CAN ID (hex)
Temperature, actual value	0x11400080 (Read Only)
Temperature, set-point value	0x114000C0 (Read/Write)
Door open / closed	0x10010100 (Read Only)

MODBUS Table

Name	Modbus address (read)	Modbus address (write)	Unit	Data-Type	Size (Bytes)	Limit (min)	Limit (max)	Description
PV_Temp_Interior	0x560A	-	°C	T_TFLOAT	4	-99999	99999	Process Value, shows actual temperature of the appliance
SP_Temp_Interior	0x8528	0x4107	°C	T_TFLOAT	4	-99999 (-95°C)	99999 (-40°C)	Set Point, shows/writes temperature-setting (set-point)
PV_Status_Door	0x5900	-	-	T_TBOOL16	2	0	1	Process Value, shows the status of the door (open/close)
PV_Colletive_Alarm	0x8526	-	-	T_TBOOL16	2	0	1	Process Value, shows if an alarm is active
PV_Temp_Condensate	0x5696	-	°C	T_TFLOAT	4	-99999	99999	Process Value, shows the temperature of condenser
PV_Temp_Cascade	0x5722	-	°C	T_TFLOAT	4	-99999	99999	Process Value, shows the temperature of cascade
PV_Temp_Ambience	0x5768	-	°C	T_TFLOAT	4	-99999	99999	Process Value, shows the ambient temperature
PV_Temp_Safety_ Controller	0x5650	-	°C	T_TFLOAT	4	-99999	99999	Process Value, shows the safety-controller sensor.
PV_Status_ Safety_Controller	0x8524	-	-	T_TBOOL16	2	0	1	Process Value, shows the status of safety controller
PV_Enable_Compres-sor_Stage_1	0x8504	-	-	T_TBOOL16	2	0	1	Process Value, shows the status of compressor stage 1
PV_Enable_Compres-sor_Stage_2	0x8505	-	-	T_TBOOL16	2	0	1	Process Value, shows the status of compressor stage 2