

LIEBHERR Supermarket Communication

**Modbus-RTU Communication Protocol
Rev. 2.0 Platform - 06/2025**

**Release Status: Final Version V1.0
2025-09-22**

LIEBHERR

Table of Contents

1	Disclaimer.....	3
2	General Information – Physical Layer	3
2.1	Serial configuration.....	3
2.2	Addressing	3
2.3	Physical Layer / Wiring	3
3	Commands.....	4
3.1	Read Coils (0x01)	4
3.2	Read Multiple Holding Registers (0x03)	5
3.3	Write Single Coil (0x05)	6
3.4	Write Single Register (0x06)	6
3.5	Write multiple Coils (0x0F)	7
3.6	Write Multiple Holding Registers (0x10)	7
3.7	Diagnostic (0x08), Subcode 0 (Echo)	8
4	Modbus Diagnostics.....	9
4.1	CRC Checksum.....	10
4.2	Exception Codes	10
5	Description of Modbus Registers.....	11
5.1	Reading Device Identification	11
5.2	Sample: Door State Communication.....	12
5.3	Sample: Set Temperature	12
6	Modbus Table.....	13

1 Disclaimer

This document describes the superset of all available Liebherr Modbus commands. Depending on the appliance type not all commands are relevant. Please refer to the user manual of the appliance to see the entire supported function set.

2 General Information – Physical Layer

2.1 Serial configuration

Physical Layer:	RS485
Baud Rate:	19200 bps (optional 9600, 38400 bps)
Parity / Stopbits:	Even +1 Sb (optional: odd +1 Sb, none + 1 Sb)
Timeout Time:	250 ms

Remarks:

For details, please see Modbus Specification RTU Transmission Mode.

2.2 Addressing

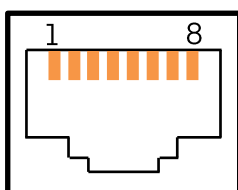
The slaves with *Liebherr Supermarket Communication* can have an address in the range from 1 to 247. Address 0 is used for broadcast messages. It is not a valid controller address.

Due to the mostly heterogeneous bus users each device should be addressed individually. That means the light on or off command is sent to each address individually. Broadcast messages are ignored by the device.

2.3 Physical Layer / Wiring

Modbus is defined as a 2-wire bus. The lines B+ and A- are the differential signal lines in according with EIA/TIA-485 standard. A common line must also interconnect all the devices of the bus (see “Modbus over serial line specification and implementation guide V1.02” from Modbus-IDA.org).

Use standard CAT5 (or better) patch cables to connect the cabinets.



The pin-out of the connector is as follows.

Figure 1 RJ45 female connector

- | | |
|----|-------------|
| 1: | B+ |
| 2: | A- |
| 3: | GND, common |
| 4: | B+ |
| 5: | B+ |
| 6: | GND, common |
| 7: | A- |
| 8: | A- |

Attention: Due to compatibility with other devices in commercial refrigeration the pin-out is different from the standard pin-out.

The Modbus has to be one line (without trunks) and it has to be terminated with a resistor 120 Ohms at the end of the line. A termination resistor with RJ45-plug is available. The resistor is between pin 2 and pin 4 (B+ / A-).

The Modbus master must provide a full termination with Pull Up, Pull Down and termination resistors.

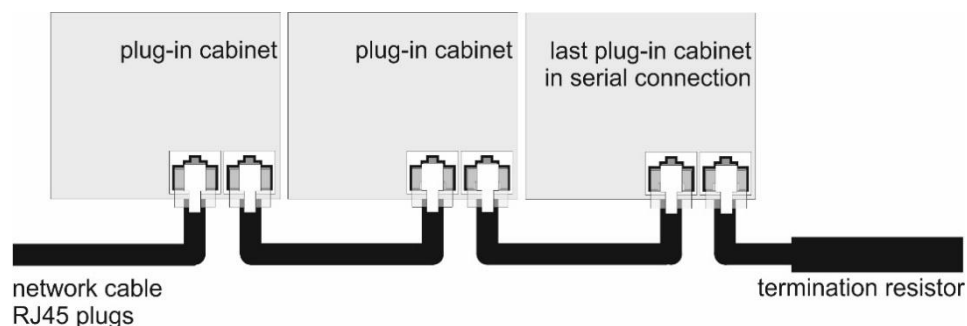


Figure 2 Cable connection between cabinets with termination resistor

3 Commands

For the communication between master and slave devices commands are defined to obtain the data access. In Modbus specification these commands are called *function codes*.

Accessing data needs addressing the related register.

In this documentation the register address is given for each parameter which has to be set in the message with an absolute address without offset.

The register data in the response message are packed as two bytes per register, with the binary contents right justified within each byte. For each register, the first byte contains the high order bits and the second contains the low order bits.

For more detailed information please refer to the documentation of <http://www.modbus.org>.

3.1 Read Coils (0x01)

This function code is used to read the contents of bit registers in a slave. The request message specifies the starting register address and the number of coils. If the requested number of coils is greater than 1, the requested coils must have continuous addresses.

Request from master (8 bytes):

slave address	1 - 247
function code	0x01
starting register address MSB	0x0000 - 0xFFFF
starting register address LSB	
quantity of coils MSB	N (1 - 2000)
quantity of coils LSB	
CRC LSB	
CRC MSB	

Response from slave (5 + N bytes):

slave address	1 - 247
function code	0x01
byte count registers	N
coils 1-8	1 = LSB, 8= MSB
following coils	(if N > 8)
CRC LSB	
CRC MSB	

Note:

Function code 0x02 (Read discrete inputs) is also supported.

3.2 Read Multiple Holding Registers (0x03)

This function code is used to read the contents of holding registers in a slave. The request message specifies the starting register address and the number of registers. If the requested number of registers is greater than 1, the requested registers must have continuous addresses.

Request from master (8 bytes):

slave address	1 - 247
function code	0x03
starting register address MSB	0x0000 - 0xFFFF
starting register address LSB	
quantity of registers MSB	N (1 - 125)
quantity of registers LSB	
CRC LSB	
CRC MSB	

*Response from slave (5 + 2*N bytes):*

slave address	1 - 247
function code	0x03
byte count registers	2 x N
1 register value MSB	
1 register value LSB	
N register value MSB	(If N > 1)
N register value LSB	
CRC LSB	
CRC MSB	

Note:

Function code 0x04 (Read Input registers) is also supported.

3.3 Write Single Coil (0x05)

This function code is used to write a single bit as a command to either ON or OFF in a slave. A value of 0xFF00 requests the register to be ON. A value of 0x0000 requests it to be OFF. All other values are illegal and will not affect the register. The request message specifies the address of the register to be forced.

The normal response is an echo of the request, returned after the register state has been written.

Request from master (8 bytes):

slave address	1 - 247
function code	0x05
register address MSB	0x0000 - 0xFFFF
register address LSB	
output value MSB	0xFF00 or 0x0000
output value LSB	
CRC LSB	
CRC MSB	

Response from slave (8 bytes):

slave address	1 - 247
function code	0x05
register address MSB	0x0000 - 0xFFFF
register address LSB	
output value MSB	0xFF00 or 0x0000
output value LSB	
CRC LSB	
CRC MSB	

3.4 Write Single Register (0x06)

This function code is used to write the contents of holding registers in a slave. The request message specifies the register address, the number of registers to be written, the quantity of bytes and the register values.

Request from master (8 bytes):

slave address	1 - 247
function code	0x06
register address MSB	0x0000 - 0xFFFF
register address LSB	
register value MSB	
register value LSB	
CRC LSB	
CRC MSB	

Response from slave (8 bytes):

slave address	1 - 247
function code	0x06
register address MSB	0x0000 - 0xFFFF
register address LSB	
register value MSB	
register value LSB	
CRC LSB	
CRC MSB	

3.5 Write multiple Coils (0x0F)

This function code is used to write the contents of multiple bits in a slave. The request message specifies the starting register address, the number of registers to be written, the quantity of bytes and the register values. If the number of coils to be written is greater than 1, the coils must have continuous addresses.

Request from master (9 + N bytes):

slave address	1 - 247
function code	0x0F
starting register address MSB	0x0000 - 0xFFFF
starting register address LSB	
quantity of coils MSB	N (1-1968)
quantity of coils LSB	
byte count registers	N
values 1, coils 1-8	1 = LSB, 8 = MSB
values N, coils 9 following	(If N > 8)
CRC LSB	
CRC MSB	

Response from slave (8 bytes):

slave address	1 - 247
function code	0x0F
starting register address MSB	0x0000 - 0xFFFF
starting register address LSB	
quantity of coils MSB	(1 - 1968)
quantity of coils LSB	
CRC LSB	
CRC MSB	

3.6 Write Multiple Holding Registers (0x10)

This function code is used to write the contents of multiple holding registers in a slave. The request message specifies the starting register address, the number of registers to be written, the quantity of bytes and the register values.

Request from master (8 + 2 x N bytes):

slave address	1 - 247
function code	0x10
starting register address MSB	0x0000 - 0xFFFF
starting register address LSB	
quantity of registers MSB	N (1-123)
quantity of registers LSB	
byte count registers	2 x N
1 register value MSB	
1 register value LSB	
N register value MSB	(If N > 1)
N register value LSB	
CRC LSB	
CRC MSB	

Response from slave (8 bytes):

slave address	1 - 247
function code	0x10
starting register address MSB	0x0000 - 0xFFFF
starting register address LSB	
quantity of registers MSB	(1 - 123)
quantity of registers LSB	
CRC LSB	
CRC MSB	

3.7 Diagnostic (0x08), Subcode 0 (Echo)

This function code is used for diagnostic purposes. Liebherr Open Bus supports Sub-function code 0.

Request from master (6 + N bytes):

slave address	1 - 247
function code	0x08
Subfunction MSB	0x0000
Subfunction LSB	
Data	N Bytes, N <= 250
CRC LSB	
CRC MSB	

Response from slave (6 + N bytes):

slave address	1 - 247
function code	0x08
Subfunction MSB	0x0000
Subfunction LSB	
Data received from master	N Bytes
CRC LSB	
CRC MSB	

The following subcodes are supported:

```

0x0000 Return Query Data
0x0001 Restart Communications Option
0x0002 Return Diagnostic Register
0x000A Clear Counters and Diagnostic Register
0x000B Return Bus Message Count
0x000C Return Bus Communication Error Count
0x000D Return Bus Exception Error Count
0x000E Return Server Message Count

```


4 Modbus Diagnostics

Function Code (hex)	Function Sub Code (hex)	Name	Description
0x08	0x00	Return Query Data	The data passed in the request data field is to be returned (looped back) in the response. The entire response message should be identical to the request.
0x08	0x01	Restart Communications Option	<p>The remote device serial line port must be initialized and restarted, and all of its communications event counters are cleared.</p> <p>A normal response is returned. This occurs before the restart is executed.</p> <p>When the remote device receives the request, it attempts a restart and executes its power-up confidence tests. Successful completion of the tests will bring the port online.</p> <p>A request data field contents of FF 00 hex causes the port's Communications Event Log to be cleared also. Contents of 00 00 leave the log as it was prior to the restart.</p>
0x08	0x02	Return Diagnostic Register	<p>The contents of the remote device's 16-bit diagnostic register are returned in the response.</p> <p>Return</p> <ul style="list-style-type: none"> Bus Message Count Bus Communication Error Count Bus Exception Error Count Server Message Count
0x08	0x0A	Clear Counters and Diagnostic Register	The goal is to clear all counters and the diagnostic register. Counters are also cleared upon power-up.
0x08	0x0B	Return Bus Message Count	The response data field returns the quantity of messages that the remote device has detected on the communications system since its last restart, clear counters operation, or power-up.
0x08	0x0C	Return Bus Communication Error Count	The response data field returns the quantity of CRC errors encountered by the remote device since its last restart, clear counters operation, or power-up.
0x08	0x0D	Return Bus Exception Error Count	The response data field returns the quantity of Modbus exception responses returned by the remote device since its last restart, clear counters operation, or power-up.
0x08	0x0E	Return Server Message Count	The response data field returns the quantity of messages addressed to the remote device, or broadcast, that the remote device has processed since its last restart, clear counters operation, or power-up.

4.1 CRC Checksum

Each transmitted message contains a CRC checksum which is calculated by the transmitting device. The receiver compares this value in the CRC field with a calculated value on receiving the message. An error occurs if the calculated CRC value is not equal to the received one.

The low-order byte (LSB) of the CRC checksum is transmitted first, followed by the high-order byte (MSB).

The following code (in C) is for calculating the CRC checksum:

```
#define MODBUS_POLYNOM 0xA001
unsigned int crc16;

void modbusCRC16 (unsigned char* msg, unsigned char lenMsg) {
    unsigned char byteIndex;
    unsigned char n;
    unsigned char bitValue;

    crc16 = 0xFFFF;
    for(byteIndex=0;byteIndex<lenMsg;byteIndex++) {
        crc16 ^= msg[byteIndex];
        for(n=0;n<8;n++) {
            bitValue = crc16 & 0x0001;
            crc16 = crc16 >> 1;
            if(bitValue == 1) crc16 ^= MODBUS_POLYNOM;
        }
    }
}
```

4.2 Exception Codes

When a master device sends a request to a slave device it expects a normal response. The normal response contains the requested data.

Each device with *Liebherr Supermarket Communication* answers with an exception code when it has received a command from a master but it is not possible to execute the command. The exception codes echoes the received function code with bit 7 set.

The exception response has the following format (5 bytes):

slave address	1 - 247
function code	value + 0x80
exception code	01, 02, 03, 04, 06
CRC LSB	
CRC MSB	

Possible exception codes:

Code	Name	Meaning
01	Illegal function	The received function code is not supported by the slave. The function codes 0x03 and 0x10 are allowed only.
02	Illegal data address	The data address received in the request from a master is not valid in the slave. Only the parameters listed in chapter 5 have valid addresses.
03	Illegal data value	The value contained in the data field of a slave message is not valid because it is out of the allowable range.

5 Description of Modbus Registers

5.1 Reading Device Identification

Liebherr Supermarket Communication provides various ways to identify the connected device.

a) Reading device identification with function code 0x03 (read holding register)

Register address (hex)	Description	Size (word)	Notes
0x0000	Family Code	1	Example: "Li" MSByte: "L" (0x4C) LSByte: "i" (0x69)
0x0001	Device Code MSWord	1	First two letters of model: MSByte: "B" (0x42) LSByte: "R" (0x52)
0x0002	Device Code LSWord	1	Third and fourth letter of model: MSByte: "P" (0x50) LSByte: "v" (0x76)
0x0003	Firmwareversion	1	MSByte, LSByte: e.g. 0x01A8=V1.168

b) Reading the serial number with function code 0x03 (read holding register)

Register address (hex)	Description	Size (word)	Notes
0x1003 to 0x1008	Serial number	6	String of max. 12 ASCII chars Example: "553365861" Hex: 35 35 33 33 36 35 38 36 31 00 00 00

Remark: Only available in LSC2.0

c) Reading the service number with function code 0x03 (read holding register)

Register address (hex)	Description	Size (word)	Notes
0x1009 to 0x100E	Service number	6	String of max. 12 ASCII chars Example: "994614701" Hex: 39 39 34 36 31 34 37 30 31 00 00 00

Remark: Only available in LSC2.0

d) Reading the model number with function code 0x03 (read holding register)

Register address (hex)	Description	Size (word)	Notes
0x100F to 0x1016	Model number	8	String of max. 16 ASCII chars Example: "BRPvg 8401" Hex: 42 52 50 76 67 20 38 34 30 31 00 00 00 00 00 00

5.2 Sample: Door State Communication

*<user **opened** the door*

Request: 0x01 0x01 0x00 0xD1 0x00 0x01 0xAD 0xF3

Response door opened: 0x01 0x01 0x01 **0x01** 0x90 0x48

*<user **closed** the door>*

Request: 0x01 0x01 0x00 0xD1 0x00 0x01 0xAD 0xF3

Response door closed: 0x01 0x01 0x01 **0x00** 0x51 0x88

Description:

Format request: Modbus address 1 byte, function code 1 byte, address 2 byte, number coils 2 byte, CRC 2 byte

Format response: Modbus address 1 byte, function code 1 byte, number bytes 1 byte, value 1 byte, CRC 2 byte

5.3 Sample: Set Temperature

Request air temperature: 0x01 0x03 0x20 0x09 0x00 0x01 0x5F 0xC8

Response: 0x01 0x03 0x02 0x00 0x35 0x78 0x53

Request set temperature: 0x01 0x03 0x20 0x04 0x00 0x01 0xCE 0x0B

Response: 0x01 0x03 0x02 0x00 0x32 0x39 0x91

Description:

Format request: Modbus address 1 byte, function code 1 byte, address 2 byte, number coils 2 byte, CRC 2 byte

Format response: Modbus address 1 byte, function code 1 byte, number bytes 1 byte, value 2 byte, CRC 2 byte

Increase set temperature: 0x01 0x06 0x20 0x04 0x00 0x3C 0xC3 0xDA

Response: 0x01 0x06 0x20 0x04 0x00 0x3C 0xC3 0xDA

Description:

Format request: Modbus address 1 byte, function code 1 byte, address 2 byte, register value 2 byte, CRC 2 byte

Format response: Modbus address 1 byte, function code 1 byte, number bytes 1 byte, register value 2 byte, CRC 2 byte

6 Modbus Table

Address (hex)	Function Code	R/W	Register Description	Comment
0x0000	3	RO	Family code Example: "Li" MSByte: "L" (0x4C) LSByte: "i" (0x69)	
0x0001 to 0x0002	3	RO	First 4 letters of the model number Example: „BRPv" Register 1 "BR" MSByte: "B" (0x42) LSByte: "R" (0x52) Register 2 "Pv" MSByte: "P" (0x50) LSByte: "v" (0x76)	
0x0003	3	RO	Firmwareversion Powerboard, MSB, LSB Example: V1.168 MSByte: 0x01 LSByte: 0xA8	
0x0064	3, 6	RW	Set Temperature setpoint	Unit °C Resolution: 0.1 Format: signed 16 bit integer, Gain 10. Example: 0xFF4C = -18°, 0x50 = +8° Note: Write access to the Set Temperature must be a rounded value. e.g. 0x51 = 8.1°C is not allowed!
0x0065	3	RO	Display Temperature actual control value	
0x006E	3	RO	Air Temperature actual value probe 1	Unit °C Resolution: 0.1 Format: signed 16 bit integer, Gain 10. Example: 0xFF4C = -18.0°, 0x51 = +8.1°
0x006F	3	RO	Evaporator Temperature actual value probe 2	
0x0070	3	RO	Product Temperature actual value probe 3	
0x00C7	3	RO	Allowed Setpoints (bit-mapped) (description see table A)	
0x00C8	1	RO	State Zone 0 = Off 1 = On	Device operation
0x00C9	1	RO	General Alarm 0 = Off 1 = On	General alarm
0x00CA	1	RO	State Compressor 0 = Off 1 = On	Cooling
0x00CB	1	RO	State Defrost 0 = Off 1 = On	Defrost

Address (hex)	Function Code	R/W	Register Description	Comment
0x00D1	1	RO	State Door 0 = DOOR_CLOSED 1 = DOOR_OPEN	Door 1 (1 = Open)
0x00E8	1	RO	Error Air Sensor 0 = Off 1 = On	Error probe 1
0x00E9	1	RO	Error Evaporator Sensor 0 = Off 1 = On	Error probe 2
0x00EC	1	RO	Temperature Alarm low 0 = Off 1 = On	Low goods temperature alarm
0x00ED	1	RO	Temperature Alarm high 0 = Off 1 = On	High goods temperature alarm
0x03E9	1	RO	State Presentation light 0 = Off 1 = On	Lighting
0x1000	3	RO	Capabilities Bit 0: Energy Saver 1 = available	Capabilities indicate if a dedicated function is supported by the connected appliance. Any access (R/W) to a non-supported function will result in an undefined register content. The values are not valid.
			Capabilities Bit 1: RTC 1 = available	
			Capabilities Bit 2: CAN configured 1 = available	
			Capabilities Bit 3: Temperature Unit °C / °F 1 = available	
			Capabilities Bit 4: Presentation Light 1 = available	
			Capabilities Bit 5 – 15: RFU	

Address (hex)	Function Code	R/W	Register Description	Comment
0x1001	3	RO	States bit 0 – 1: Presentation Light 0 = PRESENTATIONLIGHT_OFF 1 = PRESENTATIONLIGHT_ALWAYS_OFF 2 = PRESENTATIONLIGHT_ON 3 = PRESENTATIONLIGHT_ALWAYS	
			States bit 2: Compressor 0 = Off 1 = On	
			States bit 3: RFU	
			States bit 4: Alarm Relay 0 = Off 1 = On	
			States bit 5: Energy Saver 0 = Off 1 = On	
			States bit 6: Maintenance 0 = Off 1 = On	
			States bit 7: Temperature Unit 0 = Off 1 = On	
			States bit 8 - 15: RFU	
0x1002	3	RO	Errors / Alarms bit 0 – 1: Device Alarm 0 = ALARM_OFF 1 = ALARM_ON 2 = ALARM_QUITTED	
			Errors / Alarms bit 2: Error Ambient Sensor 0 = ALARM_OFF 1 = ALARM_ON	
			Errors / Alarms bit 3: RFU	
			Errors / Alarms bit 4: Safety Temperature Control 0 = Off 1 = On	
			Errors / Alarms bit 5 – 15: RFU	
0x1003 To 0x1008	3	RO	Serial Number	String of max. 12 ASCII chars i.e. "553365861"
0x1009 To 0x100E	3	RO	Service Number	String of max. 12 ASCII chars i.e. "994614701"
0x100F To 0x1016	3	RO	Model Number	String of max. 16 ASCII chars. i.e. " BRPvg 8401"

Address (hex)	Function Code	R/W	Register Description	Comment
0x1017	3	RO	Number of Zones min: 1 max: 2	defines the number of Cooling-Zones Addresses 0x3000 and higher are for zone 2 and thus, only valid if number of zones is 2.
0x1018	3	RO	Compressor Speed	Compressor-Speed [rpm]
0x1019	3	RO	Operating Hours	Operating Hours of Appliance [h]
0x101A To 0x101B	3	RO	RTC Date/Time	Local time from appliance Format: UTC ISO 8601 Example: Unix Timestamp: 1721881216 GMT: Jul 25 2024 04:20:16 GMT+0000
0x101C To 0x101D	3, 6	RW	Set UTC Date/Time	To set the RTC on the appliance write UTC Date/Time write UTC Offset send command "Set RTC" The local offset to UTC in 1 min steps Format: UTC ISO 8601 Example: Unix Timestamp: 1721881216 GMT: Jul 25 2024 04:20:16 GMT+0000 Your Time Zone: Jul 25 2024 06:20:16 GMT+0200 (Mitteleuropäische Sommerzeit) Relative: 2 hours ago (see https://www.unixtimestamp.com/)
0x101E	3, 6	RW	Set UTC Offset Example: Jul 25 2024 06:20:16 GMT+0200 ==> Set UTC Offset = 120	
0x1023	3	RO	CM SW-Version MSB, LSB e.g.: 220 = V2.20	SW-Version of CM012
0x1025	3	RO	Ambient Temperature	Temperature-Sensor Unit °C (fix) Resolution: 0.1 Format: signed 16 bit integer, Gain 10. Example: 0xFF4C = -18°, 0x50 = +8°
0x1026	6	RW	Cmd Presentation Light 0 = PRESENTATIONLIGHT_OFF 1 = PRESENTATIONLIGHT_ALWAYS_OFF 2 = PRESENTATIONLIGHT_ON 3 = PRESENTATIONLIGHT_ALWAYS_ON	Switch On/Off Presentation Light

Address (hex)	Function Code	R/W	Register Description	Comment
0x1800	5	WO	Cmd Confirm Maintenance Reset state Maintenance on the Fridge/Freezer Appliance	<p>Write Single Coil</p> <p>This function code is used to write a single output to either ON or OFF in a remote device. The requested ON/OFF state is specified by a constant in the request data field. A value of FF00 hex requests the output to be ON. A value of 0000 requests it to be OFF. All other values are illegal and will not affect the output.</p> <p>The Request PDU specifies the address of the coil to be forced. Coils are addressed starting at zero. Therefore coil numbered 1 is addressed as 0. The requested ON/OFF state is specified by a constant in the Coil Value field. A value of 0XFF00 requests the coil to be ON. A value of 0X0000 requests the coil to be off. All other values are illegal and will not affect the coil.</p>
0x1801	5	WO	Cmd Confirm Device Alarm By confirming the Device Alarm on the Fridge/Freezer Appliance the alarm-state will change to ALARM_QUITTED	
0x1802	5	WO	RFU	
0x1803	5	WO	Cmd Set RTC To set RTC on the appliance first write "Set UTC Date/Time" and "Set UTC Offset". By sending this command CM012 will write the UTC to the appliance.	
0x1804	5	WO	Cmd Set Energy Saver Start Energy Save Function on the appliance	

Address (hex)	Function Code	R/W	Register Description	Comment
Zone 1				Address range from 0x2000 to 2FFF is exclusive to zone 1.
0x2000	3	RO	Capabilities bit 0 – 3: Zone Type 0 = Cooling 3 = Freezing	Capabilities indicate if a dedicated function is supported by the connected appliance. Any access (R/W) to a non-supported function will result in an undefined register content. The values are not valid.
			Capabilities bit 4: Manual Defrost 1 = available	
			Capabilities bit 5: DoorLock 1 = available	
			Capabilities bit 6: Air Sensor 1 = available	
			Capabilities bit 7: Evaporator Sensor 1 = available	
			Capabilities bit 8: Safety Sensor 1 = available	
			Capabilities bit 9: Product Sensor 1 = available	
			Capabilities bit 10 – 15: RFU	
0x2001	3	RO	States bit 0: Door 0 = DOOR_CLOSED 1 = DOOR_OPEN	
			States bit 1: Fan 0 = Off 1 = On	
			States bit 2: Defrost 0 = Off 1 = On	
			States bit 3: Manual Defrost 0 = Off 1 = On	
			States bit 4 - 5: Door Lock 0 = Unlock 1 = Lock 2 = Emergency access	
			States bit 6: State Zone 0 = Off 1 = On	
			States bit 7: State Cleaning Mode 0 = Off 1 = On	
			States bit 8: Product Sensor 0 = Off 1 = On	

Address (hex)	Function Code	R/W	Register Description	Comment
			States bit 9 - 15: RFU	
0x2002	3	RO	Errors / Alarms bit 0-1: Door alarm 0 = ALARM_OFF 1 = ALARM_ON 2 =ALARM_QUITTED	
			Errors / Alarms bit 2-3: Power failure alarm 0 = ALARM_OFF 1 = ALARM_ON 2 =ALARM_QUITTED	
			Errors / Alarms bit 4-5: Temperature alarm low 0 = ALARM_OFF 1 = ALARM_ON 2 =ALARM_QUITTED	
			Errors / Alarms bit 6-7: Temperature alarm high 0 = ALARM_OFF 1 = ALARM_ON 2 =ALARM_QUITTED	
			Errors / Alarms bit 8: Error air sensor 0 = ALARM_OFF 1 = ALARM_ON	
			Errors / Alarms bit 9: Error evaporator sensor 0 = ALARM_OFF 1 = ALARM_ON	
			Errors / Alarms bit 10: Error product sensor 0 = ALARM_OFF 1 = ALARM_ON	
			Errors / Alarms bit 11: Error safety sensor 0 = ALARM_OFF 1 = ALARM_ON	
			Errors / Alarms bit 12-15: RFU	
0x2003	3	RO	Display Temperature	Temperature shown in the appliance UI
0x2004	3, 6	RW	Set Temperature	<p>"Set Temperature" can be set within the given range between "Min Set Temperature" and "Max Set Temperature"</p> <p>Unit °C / °F: see state "Temperature Unit" Resolution: 0.1 Format: signed 16 bit integer, Gain 10. Example: 0xFF4C = -18°, 0x50 = +8°</p>

Address (hex)	Function Code	R/W	Register Description	Comment
0x2005	3	RO	Min Set Temperature	
0x2006	3	RO	Max Set Temperature	
0x2007	3	RO	Value Temperature Alarm low only valid if alarm is active	Actual temperature in case of Temperature Alarm Unit °C / °F: see state "Temperature Unit" Resolution: 0.1 Format: signed 16 bit integer, Gain 10. Example: 0xFF4C = -18°, 0x50 = +8°
0x2008	3	RO	Value Temperature Alarm high only valid if alarm is active	
0x2009	3	RO	Sensor Air Temperature	Temperature sensor Unit °C (fix) Resolution: 0.1 Format: signed 16 bit integer, Gain 10. Example: 0xFF4C = -18°, 0x51 = +8.1°
0x200A	3	RO	Sensor Evaporator Temperature	
0x200B	3	RO	Sensor Product Temperature	
0x200C	3	RO	Safety Sensor Temperature	
0x200D	3	RO	Lower Temperature Setting Limit low	Configured temperature for generating temperature alarm Unit °C / °F: see state "Temperature Unit" Resolution: 0.1 Format: signed 16 bit integer, Gain 10. Example: 0xFF4C = -18°, 0x50 = +8°
0x200E	3	RO	Lower Temperature Setting Limit high	
0x200F	3, 6	RW	Temperature Alarm Delay	Delay from temperature out of range till temperature alarm Unit: minutes Range: 1 - 255
0x2010	3, 6	RW	Door Alarm Delay	Delay from door open till door alarm Unit: seconds Range 1 - 255
0x2011	3	RO	Power Failure Temperature Max only valid if alarm is active	Shows the maximum temperature at a power failure
0x2012	3	RO	Power Failure Temperature Min only valid if alarm is active	Shows the minimum temperature at a power failure

Address (hex)	Function Code	R/W	Register Description	Comment
0x2800	5	WO	Cmd Unlock Door Lock	<p>Write Single Coil</p> <p>This function code is used to write a single output to either ON or OFF in a remote device. The requested ON/OFF state is specified by a constant in the request data field. A value of FF00 hex requests the output to be ON. A value of 0000 requests it to be OFF. All other values are illegal and will not affect the output. The Request PDU specifies the address of the coil to be forced. Coils are addressed starting at zero. Therefore coil numbered 1 is addressed as 0. The requested ON/OFF state is specified by a constant in the Coil Value field. A value of 0XFF00 requests the coil to be ON. A value of 0X0000 requests the coil to be off. All other values are illegal and will not affect the coil.</p>
0x2801	5	WO	Cmd Set Manual Defrost	
0x2802	5	WO	Cmd Confirm Door Alarm	
0x2803	5	WO	Cmd Confirm Temperature Alarm low	
0x2804	5	WO	Cmd Confirm Temperature Alarm high	
0x2805	5	WO	Cmd Confirm Power Failure Alarm	

Address (hex)	Function Code	R/W	Register Description	Comment
Zone 2				Address range from 0x3000 to 3FFF is exclusive to zone 2.
0x3000	3	RO	Capabilities bit 0-3: Zone Type 0 = Cooling 3 = Freezing	Capabilities indicate if a dedicated function is supported by the connected appliance. Any access (R/W) to a non-supported function will result in an undefined register content. The values are not valid.
			Capabilities bit 4: Manual Defrost 1 = available	
			Capabilities bit 5: DoorLock 1 = available	
			Capabilities bit 6: Air Sensor 1 = available	
			Capabilities bit 7: Evaporator Sensor 1 = available	
			Capabilities bit 8: Safety Sensor 1 = available	
			Capabilities bit 9: Product Sensor 1 = available	
			Capabilities bit 10-15: RFU	
0x3001	3	RO	States bit 0: Door 0 = DOOR_CLOSED 1 = DOOR_OPEN	
			States bit 1: Fan 0 = Off 1 = On	
			States bit 2: Defrost 0 = Off 1 = On	
			States bit 3: Manual Defrost 0 = Off 1 = On	
			States bit 4-5: Door Lock = Unlock 1 = Lock 2 = Emergency access	
			States bit 6: State Zone 0 = Off 1 = On	
			States bit 7: State Cleaning Mode 0 = Off 1 = On	
			States bit 8: Product Sensor 0 = Off 1 = On	

Address (hex)	Function Code	R/W	Register Description	Comment
			States bit 9-15: RFU	
0x3002	3	RO	Errors / Alarms bit 0-1: Door alarm 0 = ALARM_OFF 1 = ALARM_ON 2 =ALARM_QUITTED	
			Errors / Alarms bit 2-3: Power failure alarm 0 = ALARM_OFF 1 = ALARM_ON 2 =ALARM_QUITTED	
			Errors / Alarms bit 4-5: Temperature alarm low 0 = ALARM_OFF 1 = ALARM_ON 2 =ALARM_QUITTED	
			Errors / Alarms bit 6-7: Temperature alarm high 0 = ALARM_OFF 1 = ALARM_ON 2 =ALARM_QUITTED	
			Errors / Alarms bit 8: Error air sensor 0 = ALARM_OFF 1 = ALARM_ON	
			Errors / Alarms bit 9: Error evaporator sensor 0 = ALARM_OFF 1 = ALARM_ON	
			Errors / Alarms bit 10: Error product sensor 0 = ALARM_OFF 1 = ALARM_ON	
			Errors / Alarms bit 11: Error safety sensor 0 = ALARM_OFF 1 = ALARM_ON	
			Errors / Alarms bit 12-15: RFU	
0x3003	3	RO	Display Temperature	Temperature shown in the appliance UI
0x3004	3, 6	RW	Set Temperature	<p>"Set Temperature" can be set within the given range between "Min Set Temperature" and "Max Set Temperature"</p> <p>Unit °C / °F: see state "Temperature Unit"</p> <p>Resolution: 0.1</p> <p>Format: signed 16 bit integer, Gain 10.</p> <p>Example: 0xFF4C = -18°, 0x50 = +8°</p>

Address (hex)	Function Code	R/W	Register Description	Comment
0x3005	3	RO	Min Set Temperature	
0x3006	3	RO	Max Set Temperature	
0x3007	3	RO	Value Temperature Alarm low only valid if alarm is active	Actual temperature in case of temperature alarm Unit °C / °F: see state "Temperature Unit" Resolution: 0.1 Format: signed 16 bit integer, Gain 10. Example: 0xFF4C = -18°, 0x50 = +8°
0x3008	3	RO	Value Temperature Alarm high only valid if alarm is active	
0x3009	3	RO	SensorAir Temperature	Temperature sensor Unit °C (fix) Resolution: 0.1 Format: signed 16 bit integer, Gain 10. Example: 0xFF4C = -18°, 0x51 = +8.1°
0x300A	3	RO	SensorEvaporator Temperature	
0x300B	3	RO	SensorProduct Temperature	
0x300C	3	RO	Safety Sensor Temperature	
0x300D	3	RO	Lower Temperature Setting Limit low	Configured temperature for generating temperature alarm Unit °C / °F: see state "Temperature Unit" Resolution: 0.1 Format: signed 16 bit integer, Gain 10. Example: 0xFF4C = -18°, 0x50 = +8°
0x300E	3	RO	Lower Temperature Setting Limit high	
0x300F	3, 6	RW	Temperature Alarm Delay	Delay from temperature out of range till temperature alarm Unit: minutes Range: 1 - 255
0x3010	3, 6	RW	Door Alarm Delay	Delay from door open till door alarm Unit: seconds Range 1 - 255
0x3011	3	RO	Power Failure Temperature Max only valid if alarm is active	Shows the maximum temperature at a power failure
0x3012	3	RO	Power Failure Temperature Min only valid if alarm is active	Shows the minimum temperature at a power failure

Address (hex)	Function Code	R/W	Register Description	Comment
0x3800	5	WO	Cmd Unlock Door Lock	<p>Write Single Coil</p> <p>This function code is used to write a single output to either ON or OFF in a remote device. The requested ON/OFF state is specified by a constant in the request data field. A value of FF00 hex requests the output to be ON. A value of 0000 requests it to be OFF. All other values are illegal and will not affect the output. The Request PDU specifies the address of the coil to be forced. Coils are addressed starting at zero. Therefore coil numbered 1 is addressed as 0. The requested ON/OFF state is specified by a constant in the Coil Value field. A value of 0XFF00 requests the coil to be ON. A value of 0X0000 requests the coil to be off. All other values are illegal and will not affect the coil.</p>
0x3801	5	WO	Cmd Set Manual Defrost	
0x3802	5	WO	Cmd Confirm Door Alarm	
0x3803	5	WO	Cmd Confirm Temperature Alarm low	
0x3804	5	WO	Cmd Confirm Temperature Alarm high	
0x3805	5	WO	Cmd Confirm Power Failure Alarm	

Table A – Bitmapping register 0x00C7

Lowbyte (Bit 0 = LSB)				Highbyte (Bit 8 = LSB)			
Bit 0	-23°C	Bit 4	-19°C	Bit 8	-4°C	Bit 12	4°C
Bit 1	-22°C	Bit 5	-18°C	Bit 9	1°C	Bit 13	5°C
Bit 2	-21°C	Bit 6	-6°C	Bit 10	2°C	Bit 14	6°C
Bit 3	-20°C	Bit 7	-5°C	Bit 11	3°C	Bit 15	7°C