



Product description: MODBUS TCP/IP for RCTC

Function: MODBUS TCP/IP communication with SIOX units connected to RCTC

General Information

RCTC operates as a MODBUS slave on the Ethernet. When a MODBUS request is received, appropriate SIOX communications are carried out to transfer the requested data to/from the intended SIOX module. RCTC then assembles a reply which is sent back to the client. A maximum of three MODBUS clients can be connected to the unit simultaneously.

Up to 62 SIOX modules can be addressed in the range 1 - 63 (the RCTC occupies address 60). By using Group Addressing even more modules can be accessed, see SIOX Addressing for more information.

Data Representation

In a SIOX module, data is organized in 16-bit parameters whose numbering starts at 0 and extends upwards to the last parameter used, which is device dependent. Please refer to the specific SIOX module manual for a detailed parameter description.

From MODBUS's point of view, the device has only one data block. The same data can be reached via several MODBUS functions, either via 16-bit register accesses or by using bit-accesses.

For register accesses the following apply: Register no. 1 corresponds to parameter no. 0, register no. 2 corresponds to parameter no. 1 and so on. Function Codes 4, 6 and 16 supports read/writes to EEPROM. This is done by adding an offset of hex 8000 to the parameter number. Please note that continuously writing to an EEPROM parameter is prohibited as it wears out the memory cell. Each parameter is only guaranteed 1 million writes.

For bit accesses the following apply: Input/Coil no. 1 is the LSB (Least Significant Bit) in parameter no. 0 whereas Input/Coil no. 16 is the MSB (Most Significant Bit) in parameter no. 0. Likewise, Input/Coil no. 17 is the LSB (Least Significant Bit) in parameter no. 1 and so on. Please refer to the specific SIOX module manual to find out where the actual I/O bits are located.

For example, assume a SIOX module that has seven outputs located in parameter no. 8, starting at the LSB.

To calculate the output address for the first output, use the following formula:

$$\text{Address} = (\text{parameter no.} \times 16) + \text{bit no.}$$

$$\Rightarrow (8 \times 16) + 1 = 129$$

The six other outputs are accessed at the succeeding addresses, 130 - 135.

Supported Function Codes

The following Function Codes are supported:

- 01 Read Coils (1-16384) corresponding to parameters 0 - 1023.
- 02 Read Discrete Inputs (1-16384) corresponding to parameters 0 - 1023.
- 03 Read Holding Registers (1-16384) corresponding to parameters 0 - 16383.
- 04 Read Input Registers (1 - 16384) corresponding to parameters 0 - 16383.
- 05 Write Single Coil (1-16384) corresponding to parameters 0 - 1023.
- 06 Write Single Register (1 - 16384) corresponding to parameters 0 - 16383.
- 15 Write Multiple Coils (1-16384) corresponding to parameters 0 - 1023.
- 16 Write Multiple Registers (1 - 16384) corresponding to parameters 0 - 16383.

SIOX Addressing

The address of the SIOX module that a MODBUS message wants to access has to be stored in the Unit Identifier position in the MBAP (ModBus Application Protocol) header. This is the last byte of the seven header bytes. The SIOX address is stored in the six lower bits. Three additional Group Addresses can be used, 61, 62 and 63. Under each Group up to 63 (1-63) additional addresses are available. The two upper bits in the Unit Identifier byte controls the Group Address handling:

- 0 0 Group Address 0 (no Group Address sent).
- 0 1 Group Address 61. Add 64 to the address.
- 1 0 Group Address 62. Add 128 to the address.
- 1 1 Group Address 63. Add 192 to the address.

Example: for a module at address 5 on Group 63 the address to set in the Unit Identifier will be 5 + 192 = 197

TCP Port Number

To connect to the unit, use the standard MODBUS TCP/IP port number 502.



Protected Parameters

To prevent accidental writes to configuration parameters, which could affect the operation of the module, a few parameters are write protected. However, note that a MODBUS error status will not be returned if a write is attempted. The write protected parameters are no. 0 - 7 and no. 16 - 22, corresponding to MODBUS registers no. 1 - 8 and no. 17 - 23.

Data Limitations

Msg 01/02: Max coils/discrete inputs to read: 128
Msg 03/04: Max registers to read: 125
Msg 15: Max coils to write: 128
Msg 16: Max registers to write: 16

Error Codes

Following error codes are returned:
Hex 0B: Module silent.
Hex 01: Illegal function specified.
Hex 02: Illegal data adress specified.
Hex 03: Illegal data value specified.

Smoke Control system specific functions

Reading of Damper Modules Status Register

Each damper module has a status register at parameter hex 20 which can be read via MODBUS TCP/IP. Use the Read Holding Registers or Read Input Registers functions to read the data as a 16-bit word. To access parameter hex 20 in a module, the register number hex 21 (decimal 33) should be used.

Use the Read Coils or Read Discrete Inputs to read the data as one or more bits. The bits (1-16) in parameter hex 20 have the following meanings when set. Some bits are not used.

Status flag	Modbus			
	0 base	1 base	Bit	
Standby state	32.0	33.1	513	Normal funktion
Fire state	32.1	33.2	514	External/Internal fire alarm
Damper error	32.3	33.4	516	Function test fail
Function test state	32.4	33.5	517	Set when function test is running
Damper closed	32.8	33.9	521	
Damper open	32.9	33.10	522	
Smoke detector normal state	32.12	33.13	525	
Local Fire Alarm Point state	32.13	33.14	526	Fire alarm from local fire detector
Smoke detector service needed	32.14	33.15	527	
Smoke loop broken	32.15	33.16	528	Will result in local fire alarm

Commands	Modbus			
	0 base	1 base	Bit	
Reset local alarms	44.2	45.3	707	Flag clears automatically
Function test	44.4	45.5	709	Flag clears automatically

Analog output	Modbus			
	0 base	1 base	Bit	
Analog out, damper *)	6	7		Hex 1000 (dec 4096) represents 100% open = 10V

*) Please note that the function (set active by default) for copying the local analog input (0-10V) to damper analog output must be disabled.