Modbus RTU and Modbus TCP Communication

For VONSCH products

Version: Version 2.01

Author: Martin Šebest

VONSCH S.R.O.

7. 7. 2025

VONSCH®
power electronics

Content

1 Introduction

This document serves as a guide for communication using MODBUS RTU and MODBUS TCP on VONSCH products:

Product	Modbus RTU	Modbus TCP	USB connector
UNIFREM	Yes	Not	USB-B
UNIFREM E	Yes	Not	USB-B
QUATROFREM	Yes	Not	USB-B
PHOTO CONTROL 3F	Yes	Not	USB-B
PV COMP	Yes	Not	USB-B
GSE CONTROL	Yes	Not	USB-B
GSE COMP	Yes	Not	USB-B
LOKO BMS	Yes	Not	USB-B
NEXICONTROL	Yes	Yes	USB-C*
NEXIFREM	Yes	Yes	USB-C*
NEXIBOOST	Yes	Yes	USB-C*

* Does not support connection via USB-C – USB-C cable.

2 Common Interface

2.1 Basic Difference Between RTU and TCP

Attribute	Modbus RTU	Modbus TCP
Physical layer	RS-485/RS-232	Ethernet (e.g. RJ45, TCP/IP)
Line layer	Timing + CRC	TCP connection (port 502)
Device identification	Slave Address (1-247)	IP Address + Unit ID
Frame format	Binary framework	Header + Application Framework

2.2 Supported Features

0x03 Read Holding Registers

0x10 Write Multiple Registers

0x17 Read and Write Multiple registers

0x02 Read Discrete Inputs

0x04 Read Input Registers

2.3 Description

Orange – Master (control system) request to slave (inverter)

Green – Positive response from slave (inverter) to master (control system)

Red – Error response for slave to master

2.4 Registers – access to parameters

Access through registers allows access to individual parameters of the inverter, enabling configuration changes or status monitoring. The description of individual parameters is provided in the user manual supplied with the inverter.

2.4.1 Register Map

Address of the register	Number of registers	Meaning of data	Meaning of the block
0x0000	2	Parameter ID = 0	Set of 1
0x0002	2	Parameter ID = 1	
0x0004	2	Parameter ID = 2	
		...	
0x2000	2	Parameter ID = 0	Set of 2
0x2002	2	Parameter ID = 1	
0x2004	2	Parameter ID = 2	
		...	
0x4000	2	Parameter ID = 0	Set of 3
0x4002	2	Parameter ID = 1	
0x4004	2	Parameter ID = 2	
		...	
0x6000	2	Parameter ID = 0	Set of 4
0x6002	2	Parameter ID = 1	
0x6004	2	Parameter ID = 2	
		...	
0x8000	x		Unused
0xE000	2	Password	Cyclic data exchange
0xE002	2	ID0	
0xE004	2	Value0	

Address of the register	Number of registers	Meaning of data	Meaning of the block
0xE006	1	CW	
0xE007	1	REF	
0xE008	1	ID1	
0xE009	1	ID2	
		...	
0xE027	1	ID32	
0xE028	x		Unused
0xE100	2	Access	Cyclic data exchange
0xE102	2	ID0	
0xE104	2	Value0	
0xE106	1	SW	
0xE107	1	ACT	
0xE108	2	Value1	
0xE10A	2	Value2	
		...	
0xE146	2	Value32	
0xE147	1	CW	
0xE148	1	REF	
0xE148	x		Unused

Data meaning:

Data	Description
Parameter ID	*Access to individual parameters* *registry address = (set – 1) * 0x2000 + ID * 2* *The parameter ID is the unique parameter number according to the product documentation*
Password	*Entering a password for protected parameter writing (reading is not protected). The password is entered as 4 ASCII characters.* *Zx ascii character value* $Z4{*}0x1000000+Z3{*}0x10000+Z2{*}0x100+Z1$ *For example, the password "AB12" is 0x32314241*
Access	*Current parameter access level* *0 – read-only, 1 – User access, 2 – Service access*
IDx	*ID of the selected parameter for access using ValueX* *Used to access a group of selectedparameters* *PHOTO CONTROL1F and PHOTO CHARGER products only remember ID1-ID8 after switching off*
ValueX	*Access the valueof the selected parameter according to IDx*
CW	*control word, see control word description*
SW	*status word, reading the current status of the inverter, see the description of the status word*
REF	*notation of the desired value, integer decimal number +/- XXX.X [%]* *I write 98.7% as the number 987*
ACT	*reading of the current value, the whole decimal number* *-UNIFREM - current speed of the inverter, +/- XXXX [rpm]*

Data	Description
	-QUATROFREM, FOTO CONTROL 3f – Current production ID 593, XXXXX [W] *-PHOTO CONTROL 1f, PHOTO CHARGER – Current production ID 278 XXXXX [W]*

The basic parameters of the product are 32 bit data. Therefore, when accessing parameters, it is necessary to access a pair of registers. The parameters are of different types (see the instructions for the converter) and therefore have different meanings.

Data	Description
SETVALUE	*The parameter is stored as a float number according to the IEEE 754 standard*
ENUM	*The parameter is stored as an integer. The given number represents the selected option 1 of n. The elections are numbered from 0.*
BITENUM	*Bit parameter. Each bit corresponds to some information (state), the respective bits are described in the documentation for the parameter. The parameter name also includes the bit number.*
SHORTCUT	*The parameter is stored as an integer. This number represents the ID of the selected parameter.*
DATE	*Date in BCD format (day = [0-3bit] + 10*[4-7bit], month = [8-11bit] + 10*[12-15bit], year = [16-19bit] + 10*[20-23bit], century =[24bit] (0-20century, 1-21 century), day of the week = [28-31bit])*

TIME	*Time in BCD format (second = [0-3bit] + 10*[4-7bit], minute = [8-11bit] + 10*[12-15bit], hour = [16-19bit] + 10*[20-23bit])*
SETVALUE_DPAR	*This parameter is dependent on the settings and can take on any of the previous types*

2.4.2 Read registers 0x03

Request

Function Code	1 Byte	**0x03**
Start Address	2 Bytes	0x0000 – 0xFFFF
Number of registers = **N**	2 Bytes	0x0001 – 0x007D

Response

Function Code	1 Byte	**0x03**
Number of bytes	1 Byte	2 x **N**
Register values	**N** x 2 Bytes	Values

Error response

Error code	1 Byte	**0x83**
Exception code	1 Byte	01, 02, 03, 04

An example of reading the current frequency ID 47, for the UNIFREM converter (address 1).

*Register = 0x2000 * (set – 1) + 2 * ID = 94 = 0x5E*

Address	Function code	Register		Number of registers	
0x01	0x03	0x00	0x5E	0x00	0x02

Address	Function code	Number of bytes	Data			
0x01	0x03	0x04	0x41	0xEA	0x7B	0x6B

The IEEE 754 0x41EA7B6B conversion is 29.31Hz.

It is possible to read multiple parameters in one cycle. For example, if we want to read DC voltage (46) and frequency (47), we can read 4 registers from address 0x5E.

2.4.3 Write registers 0x10

Request

Function Code	1 Byte	**0x10**
Start Address	2 Bytes	0x0000 – 0xFFFF
Number of registers = **N**	2 Bytes	0x0001 – 0x007B
Number of bytes	1 Byte	2 x **N**
Value of registers	**N** x 2 Bytes	Values

Response

Function Code	1 Byte	**0x10**
Start Address	2 Bytes	0x0000 – 0xFFFF
Number of registers	2 Bytes	0x0001 – 0x007D

Error response

Error code	1 Byte	**0x90**
Exception code	1 Byte	01, 02, 03, 04

Writing can be denied if it concerns diagnostic parameters, inaccessible parameters (grayed out), or if the appropriate permissions (service parameters, user password) are not set. The device parameter must be written as a whole 32 bits (always 2 registers). It is possible to write multiple parameters at once.

Example of writing the desired frequency parameter ID 344 to 13Hz (according to IEEE 754 0x41500000) for the UNIFREM inverter (address 1).

*Register = 0x2000 * (set – 1) + 2 * ID = 0x2B0*

Address	Function code	Register		Number of registers		Number of bytes	Data			
0x01	0x10	0x02	0xB0	0x00	0x02	0x45	0x41	0x50	0x00	0x00

Address	Function code	Register		Number of registers	
0x01	0x10	0x02	0xB0	0x00	0x02

2.4.4 Read and write registers 0x17

Request

Function Code	1 Byte	**0x17**
Start Reading Address	2 Bytes	0x0000 – 0xFFFF
Reading count = **M**	2 Bytes	0x0001 – 0x007D
Start Address for Writing	2 Bytes	0x0000 – 0xFFFF
Write count = **N**	2 Bytes	0x0001 – 0x0079
Number of Bytes	1 Byte	2 x **N**
Register values for writing	**N** x 2 Bytes	Values

Response

Function Code	1 Byte	**0x17**
Number of Bytes	1 Byte	2 x **M**
Register Valies Read	**M** x 2 Bytes	Values

Error response

Error code	1 Byte	**0x97**
Exception code	1 Byte	01, 02, 03, 04

The data meaning is the same as for reading registers 0x03 and writing registers 0x10. This function is mainly used for cyclic data exchange.

2.4.5 Cyclic data exchange

This is used to control the inverter using the control system. The control system sends the desired quantities (CW, desired speed, ID of quantities to be read,...) to the inverter, and the inverter responds with its status and the values of the requested quantities. The exchange is optimized so that all required information is exchanged within one communication cycle (request, response). Function 0x17 reading and writing registers is used for this. Two data exchange modes can be used.

2.4.5.1 Mode 1

The master sends CW, REF, ID of parameters to be read, password, and can write one arbitrary parameter. The inverter responds with SW, ACT, and the values of the desired parameters. Example 1, 2, 3.

CW	REF	ID 1	ID2	...	IDx
SW	ACT	Value 1	Value 2	...	Value X

2.4.5.2 Mode 2

The master sends CW, REF, and can write multiple selected parameters. The inverter responds with SW, ACT, and the values of the selected parameters for reading. In this case, the master does not send IDs for the selected parameters, but they are set directly in the inverter. (Note that different products have different numbers of parameter selections, see the register map.) Example 4.

Write Value 1	Write Value 2	...	Write Value X	CW	REF
SW	ACT	Read Value 1	Read Value 2	...	Read Value X

2.4.5.3 Example 1

Writing the start of the UNIFREM inverter (address 1), desired frequency to 25 Hz which is 50%, reading the current status, speed, DC voltage (ID 46), cooler temperature (ID 74), motor current (ID 42), binary input status (ID 184).

Address	Function code	Reading register		Number of registers		Write address		Number of registers		Number B.	Data											
										12 bytes	CW		REF = 50.0% = 500 =		ID1 = 46 = 0x2E		ID2 = 74 = 0x4A		ID3 = 42 = 0x2A		ID4 = 184 = 0xB8	
0x01	0x17	0xE1	0x06	0x00	0x0A	0xE0	0x06	0x00	0x06	0x0C	0x04	0x7F	0x01	0xF4	0x00	0x2E	0x00	0x4A	0x00	0x2A	0x00	0xB8

Address	Function code	Number B.	Data																			
		20	SW		ACT = 0x2EE = 750 rpm		Value1 = 465.45V				Value2 = 63.67°C				Value3 = 1.56A				Value4 = BIN3			
0x01	0x17	0x14	0x07	0x37	0x02	0xEE	0x43	0xE8	0xB9	0xE3	0x42	0x7E	0xB5	0x5C	0x3F	0xC8	0x1E	0xF4	0x00	0x00	0x00	0x04

2.4.5.4 Example 2

Controlling the start of the UNIFREM inverter (address 1) and the desired frequency and reading only the status.

Request: 0x01, 0x17, 0xE1, 0x06, 0x00, 0x02, 0xE0, 0x06, 0x00, 0x02, 0x04, 0x04, 0x7F, 0x01, 0xF4, 0x3A, 0xB6

Response: 0x01, 0x17, 0x02, 0x07, 0x37, 0xFE, 0x52

2.4.5.5 Example 3

Writing the parameter Maximum Frequency (ID 111 = 0x6F) to the value 50Hz (0x42480000) with the user password "AB12" for the UNIFREM inverter (address 1).

Request: 0x01, 0x17, 0xE1, 0x02, 0x00, 0x04, 0xE0, 0x00, 0x00, 0x06, 0x0C, 0x32, 0x31, 0x42, 0x41, 0x00, 0x00, 0x00, 0x6F, 0x42, 0x48, 0x00, 0x00, 0xBD, 0x8D

Response: 0x01, 0x17, 0x08, 0x00, 0x00, 0x00, 0x6F, 0x42, 0x48, 0x00, 0x00, 0x95, 0xF0

When reading back, we verified that ID0 remained unchanged and Value0 is set to the correct value.

2.4.5.6 Example 4

Write the start of the UNIFREM converter (address 1), the desired frequency at 25 Hz which is 50%, the maximum current (ID 5) at 10 A (0x41200000). Read the current state, speed, DC voltage (ID 46), Cooler temperature (ID 74), Motor current (ID 42), Binary input status (ID 184). The selected quantities are set in the changewhether, the IDs are not sent by communication.

0x01		Address
0x17		Function code
0xE1		Reading register
0x06		
0x00		Number of registers
0x0A		
0xE1		Write address
0x46		
0x00		Number of registers
0x04		
0x08	8 bytes	Number of bytes
0x41	Max. I = 10A = 0x41200000	Data
0x20		
0x00		
0x00		
0x04	CW	
0x7F		
0x01	REF = 50.0% = 500 = 0x1F4	
0xF4		

0x01		Address
0x17		Function code
0x14	20	Number B.
0x07	SW	Data
0x37		
0x02	ACT = 0x2EE = 750 rpm	
0xEE		
0x43	Value1 = 465.45V	
0xE8		
0xB9		
0xE3		
0x42	Value2 = 63.67°C	
0x7E		
0xB5		
0x5C		
0x3F	Value3 = 1.56A	
0xC8		
0x1E		
0xF4		
0x00	Value4 = BIN3	
0x00		
0x00		
0x04		

2.5 Control and status word

2.5.1 Control Word – CW

The control word "CW" is used to control the inverter (prepare the inverter, start, stop, ...).

Bit	Title	Note
0	ON	Negated OFF1
1	No OFF 2	Negated OFF2 – No Coast Stop
2	No OFF 3	Negated OFF3 – No Quick Stop
3	Enable Operation	
4	Enable Ramp Generator	UNIFREM Only
5	Unfreeze Ramp	UNIFREM Only
6	Enable Setpoint	UNIFREM Only
7	Fault Acknowledge	
8-9	-	Unused
10	Control by PLC	
11-15	-	Unused

Bit	=	Status
0	0->1	The inverter will enter the "**Switched On**" state if Coast or Quick Stop is not active.
	1->0	Emergency stop, the inverter will slow down according to set ramp. When zero rpm is reached, the inverter disconnects the voltage on the motor. Enters **the "Switching Off" state** and continues to the "**Ready For Switching On**" state if Coast or Quick Stop is not active
	0	It exits the "**Switching On Inhibited**" state and enters the "**Ready For Switching On**" state.
1	1	Continuation of the operation (Coast Stop – inactive).

Bit	=	Status
	0	Emergency stop, the voltage at the output of the inverter is immediately disconnected and the motor catches up with the remaining virtue. It enters the "**Switching On Inhibited**" state.
2	1	Continuation of the operation (Quick Stop – inactive).
	0	Emergency stop, the inverter will slow down after the set Quick stop ramp. The inverter is in the "**Switching Off**" state. When zero rpm is reached, the inverter disconnects the voltage on the motor. It enters the "**Switching On Inhibited**" state.
3	1	START the inverter. The inverter connects the voltage to the motor and smoothly increases the frequency to the desired value.
	0	STOP the gearbox. The inverter smoothly reduces the output frequency to zero and disconnects the voltage from the motor.
4	1	Normal operation of the start/stop ramps generator output.
	0	Forcing a zero desired frequency value at the output of the start/stop ramps generator.
5	1	Allows the ramp output to change.
	0	Stops the change at the ramp exit.
6	1	Normal operation of the ramp inlet.
	0	Sets the ramp input to zero.
7	0->1	Acknowledgment of a malfunction
	0	Normal operation
10	1	The inverter accepts the master's commands (accepts CW)
	0	The inverter ignores the master's commands (ignores the CW)

2.5.2 Status Word – SW

The status word "SW" shows the current status of the inverter. (Ready, in start,...)

Bit	Title	Note
0	Ready To Switch On	
1	Ready To Operate	
2	Operation Enabled	
3	Error Present	
4	No OFF2	No Coast Stop
5	No OFF3	No Quick Stop
6	Switching On Inhibited	
7	Warning Present	
8	Speed Error within tolerance	UNIFEREM only, 0 for others
9	Control Requested	
10	f Or n Reached	UNIFEREM only, 0 for others
11	Operation	
12	Set b0	Active Set
13	Set b1	
14	F demanded < 0	UNIFEREM only, 0 for others
15	-	Unused

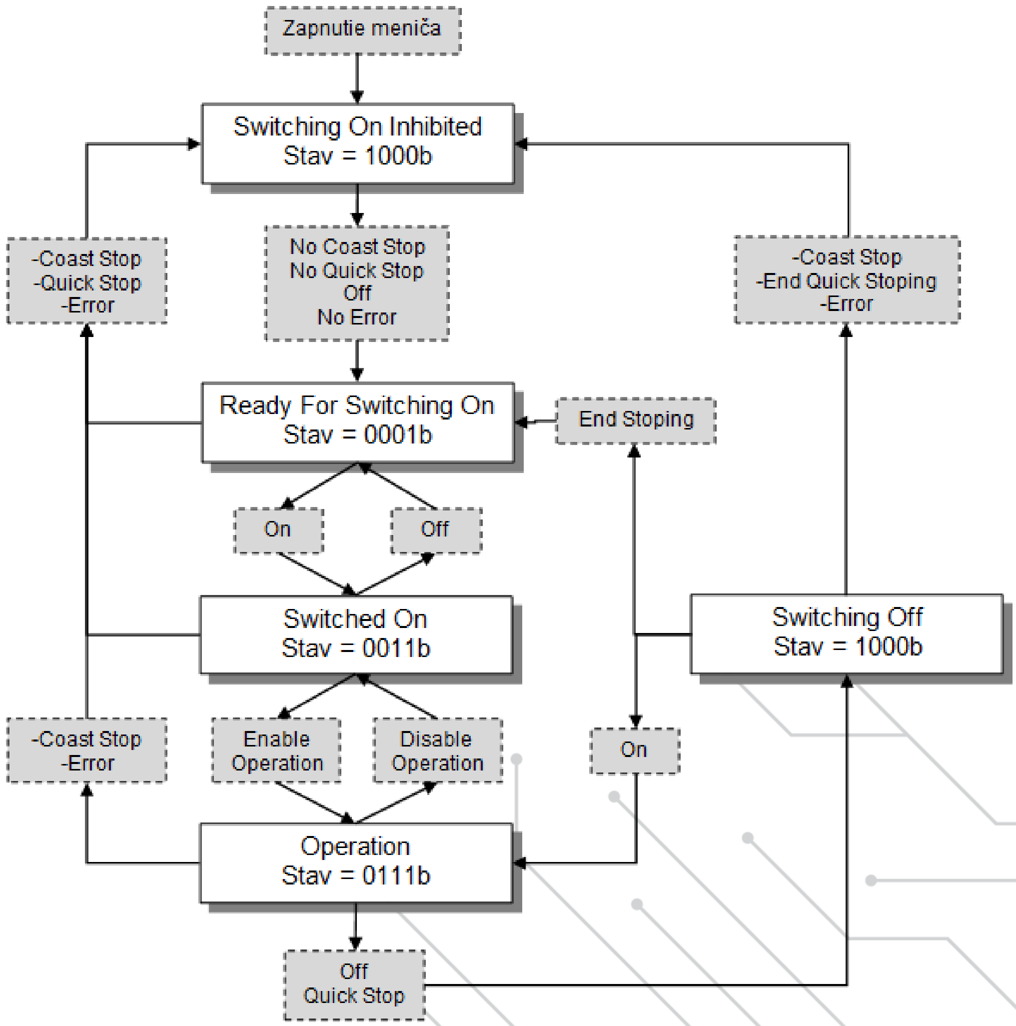
Bit	=	Status
6,2,1,0	1000	"**Switching On Inhibited**" state. It means that the Coast or Quick Stop is active (no voltage is generated at the output, the quick stop has run out), or the inverter is in error state, or the inverter is initializing.

Bit	=	Status
	0001	"**Ready For Switching On**" state. It means that Coast and Quick Stop are not active and the converter is not even in error.
	0011	"**Switched On**" state. It means that the inverter is ready to accept the inverter start command. "**Switched Off**" status. The inverter is switched off in a controlled manner either by Quick Stop or by an emergency stop - by deactivating **ON**.
	0111	"**Operation**" status. It means that the inverter is in the start.
3	1	The inverter Error state.
	0	No Error.
4	1	No Coast Stop is active.
	0	Coast Stop is active.
5	1	Quick Stop is not active.
	0	Quick Stop is active.
7	1	Warning occurred.
	0	No warning.
8	1	The frequency of the inverter is equal to the desired value.
	0	The frequency of the inverter is different from the desired value.
9	1	The inverter will respond to the control word "CW" received from the MODBUS master.
	0	The inverter will ignore the control word "CW".
10	1	The frequency of the inverter is equal to the desired value.
	0	The frequency of the inverter is different from the desired value.
11	1	The inverter is running, generates voltage at the output.

Bit	=	Status
	0	The inverter is in STOP, it does not generate voltage at the output.
12,13	00	The 1st set of parameters is active.
	01	The 2nd set of parameters is active.
	10	The 3rd set of parameters is active.
	11	The 4th set of parameters is active.
14	1	The desired frequency value is negative.
	0	The desired frequency value is positive.

2.5.3 State Machine

The individual states of the inverter are shown by white rectangles. The displayed State corresponds to bits 6,2,1,0 of the State Word "SW". Gray rectangles, signify an event that causes a transition to another state.



2.5.4 State Machine Operation

I suppose I don't want to use Coast Stop and Quick Stop (are inactive). The simple operation of fault confirmation and start entry is:

error and I want to confirm it

if (SW.**bit3** == 1 && **Button_enter**)

CW = 0x0486;

state Switching On Inhibited

else if (SW.**bit6** == 1)

CW = 0x0406;

Ready for Switching On status and I don't want to start

else if (SW.**bit1** == 0 && !**Button_Start**)

CW = 0x0407;

I want to start

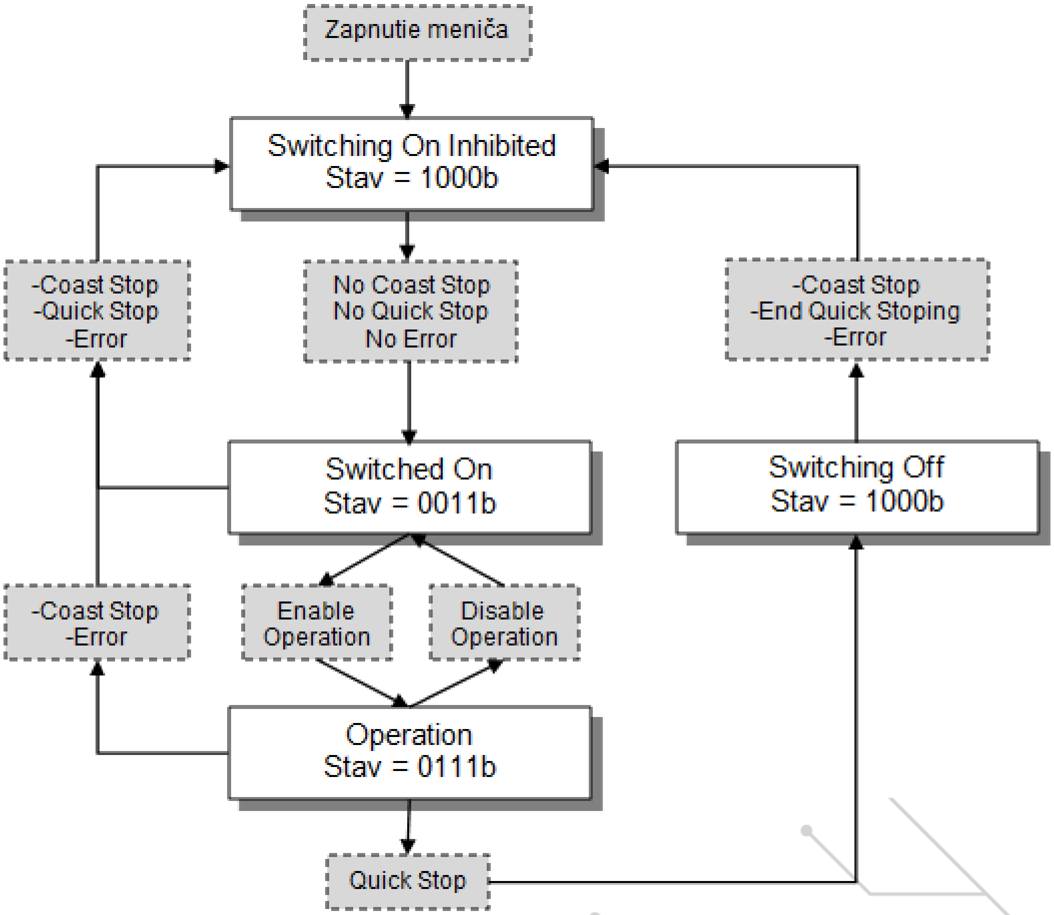
else

CW = 0x047F;

After start, the inverter connects the voltage to the motor and smoothly increases the frequency to the desired value.

At the stop, the inverter smoothly reduces the frequency to zero and disconnects the voltage from the motor.

2.5.5 Simplified State Machine



2.5.6 Simplified State Machine Operation

For the same assumptions as in normal state machine:

There is an error and I want to confirm it

if (SW.**bit3** == 1 && **Button_enter**)

CW = 0x0486;

I want to start

else if (**Button_Start**)

CW = 0x047F;

I don't want to start

else

CW = 0x0476;

2.6 Discrete Inputs – Device Status

The status of the inverter can be monitored via discrete inputs. They directly allow access to the selected bit states. Selected parameters are transmitted here, which can also be accessed via registers.

2.6.1 Discrete Input Map

Entrance address	Number of inputs	Meaning of data
0x0000	32	Inverter Status
0x0020	32	Error E1 – E32
0x0040	32	Error E33 – E64
0x0060	32	Warning W1 – W32
0x0080	32	Warning W33 – W64
0x00A0	32	Bin. Inputs + log. Blocks
0x00C0	x	Unused

2.6.2 Read discrete inputs 0x02

Request

Function Code	1 Byte	**0x02**
Start Address	2 Bytes	0x0000 – 0xFFFF
Number of inputs = **N**	2 Bytes	0x0001 – 0x07D0

Response

Function Code	1 Byte	**0x02**
Number of Bytes	1 Byte	**M**
Input values	**M** x 1 Byte	Values

if (n <> 0) then m = (n / 8) + 1; Else M = 0;

Error response

Error code	1 Byte	**0x82**
Exception code	1 Byte	01, 02, 03, 04

Example of reading the running status of the UNIFREM inverter (address 1).

Address	Function code	Input		Number of inputs	
0x01	0x02	0x00	0x02	0x00	0x01
Address	Function code	Number of Bytes	Data		
0x01	0x02	0x01	0x00		

The inverter is in the Stop state.

It is also possible to read multiple inputs in one cycle.

2.7 Input registers – history, statistics

Historical records of events and statistics are mapped to the input registers. Historical records are written gradually, and once they are full, they continue again from the beginning, so that the old values are overwritten by the new ones. Based on the timestamp (date and time), it is possible to build a historical sequence. In inverter there is a parameter History index (UNIFREM ID=243, QUATROFREM, FOTO CONTROL 3F ID=285, FOTO CONTROL 1F, FOTO CHARGER ID=53), which determines the next position for writing the event to the history. The value (index – 1) determines the last (latest) record in the history.

2.7.1 Map of Input Registers

UNIFREM, QUATROFREM, FOTO CONTROL 3F, NEXICONTROL, NEXIFREM, NEXIBOOST

Address	Number of inputs	Data meaning		Data type
0x0000	1	Record 0	Event	Integer 0x000-0x03F E1-E64 Fault 0x100-0x13F Warning W1-W64 0x200-... Specific event
0x0001	1		Reserved	
0x0002	2		Error time	TIME
0x0004	2		Date of error	DATE
0x0006	1		ID1	SHORTCUT
0x0007	1		ID2	SHORTCUT
0x0008	1		ID3	SHORTCUT
0x0009	1		ID4	SHORTCUT
0x000A	1		ID5	SHORTCUT
0x000B	1		ID6	SHORTCUT
0x000C	2		Value1	By parameter type in ID1
0x000E	2		Value2	By parameter type in ID2

Address	Number of inputs	Data meaning	Data type
0x0010	2	Value3	By parameter type in ID3
0x0012	2	Value4	By parameter type in ID4
0x0014	2	Value5	By parameter type in ID5
0x0016	2	Value6	By parameter type in ID6
0x0018	6	Reserved	
0x001E	30	Record 1	See Entry 0
0x003C		...	
0x77E2	30	Record 1023	See Entry 0
0x7800		Unused	

2.7.2 Read Input Registers 0x04

Request

Function Code	1 Byte	**0x04**
Start Address	2 Bytes	0x0000 – 0xFFFF
Number of registers = **N**	2 Bytes	0x0001 – 0x007D

Response

Function Code	1 Byte	**0x04**
Number of Bytes	1 Byte	2 x **N**
Register values	**N** x 2 Bytes	Values

Error response

Error code	1 Byte	**0x84**
Exception code	1 Byte	01, 02, 03, 04

An example of retrieving a record of the last fault from the UNIFREM converter (address 1).

The register index of the record is: 243 * 2 = 486 = 0x1E6

Request: 0x01, 0x03, 0x01, 0xE6, 0x00, 0x02, 0x24, 0x00

Response: 0x01, 0x03, 0x04, 0x42, 0x90, 0x00, 0x00, 0xEE, 0x66

The index of the record is 0x42900000 = 72

The input register of zis: 0x18 * (72 – 1) = 0x852

Request: 0x01, 0x04, 0x08, 0x52, 0x00, 0x18, 0x53, 0xB1

Response 0x01, 0x04, 0x30, 0x00, 0x06, 0x7E, 0x90, 0x34, 0x53, 0x00, 0x04, 0x01, 0x18, 0x21, 0x11, 0x00, 0x2E, 0x00, 0x2A, 0x00, 0x2F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF , 0xFF, 0xA6, 0xD0, 0x44, 0x08, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xEA, 0xA4, 0xA5, 0x92, 0xF5, 0x66, 0x37, 0x8E, 0xE8 0x9A, 0x39, 0x6B, 0x72, 0x05

From the response the information is:

Fault 0x0006 = E7 External error

Time = 0x00043453 = 4:34:53

Date = 0x21110118 = 18 January 2011

ID1 = 0x002E = 46 = DC voltage

ID2 = 0x002A = 42 = Motor current

ID3 = 0x002F = 47 = Inverter frequency

ID4 = ID5 = ID6 = 0xFFFF – *no parameters are selected for the record*

Value1 = 0x4408A6D0 = 546.6V

Value2 = 0x00000000 = 0A

Value3 = 0x00000000 = 0Hz

2.8 Exceptions

The error code is always 0x80 + requested function

Exception code	Description
0x01	The feature is not supported. The function with the requested code is not known to the inverter.
0x02	The address of the registers is out of range. Any registry address between the start and end addresses (forinitial + number) is not supported.

Exception code	Description
0x03	The request does not contain all the data, or it is out of scope (the number of possible requested registers in one request is too large).
0x04	It is not possible to access the register or parameter. An error occurred when writing to the parameter: the parameter is read-only, the rights are not set correctly, the value is out of range,...

3 Modbus RTU

3.1 Serial Interface

The physical layer of communication is on RS485.

Baud rate is selectable by a parameter in the inverter. (9600, 19200, 38400, 57600, 115200, 128000 are supported).

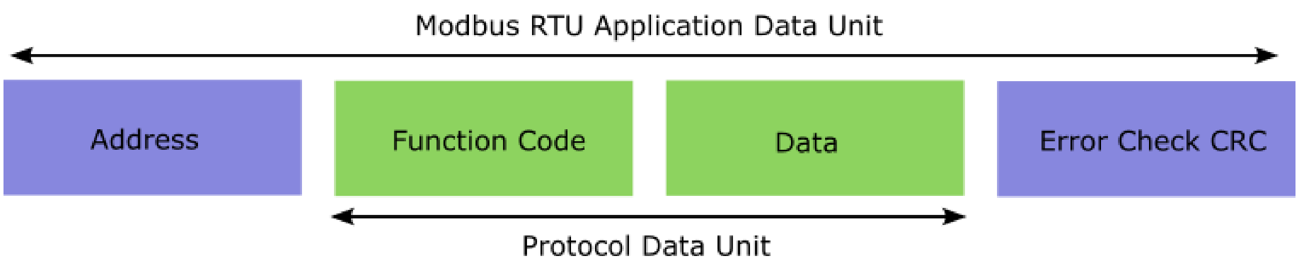
The transmission is 1 start bit, 8 data bits, 1 bit parity (even - even), 1 stop bit.

Description of the wiring of the terminals on the CANNON F 9 Pin connector.

Pin	Meaning
3	R+/T+
8	R-/T-
5	GND (Earth)

3.2 Modbus RTU Protocol Frame

The Modbus protocol in VONSCH devices corresponds to the MODBUS RTU protocol specification, which is managed by the Modbus Organization. The frame is shown in the figure. The inverter is always in the role of a slave and responds to requests from the master. The inverter accepts frames with its address or with a global address of 0. It does not respond to global frames.



Modbus RTU Frame Structure

Where address is the address of the device (inverter). It is adjustable by a parameter in the inverter in the range of 1-99.

CRC (Error check) is a checksum to identify the correctness of the frame.

3.3 Inverter Setup for Modbus Communication

To ensure proper communication of the inverter with the Modbus master, it is necessary to set some inverter parameters. The procedure to set the inverter parameters is described in detail in the user manual supplied with the changer.

UNIFREM, QUATROFREM, PHOTO CONTROL 3F , NEXICONTROL, NEXIBOOST, NEXIFREM

Parameter	Value	Unit	Note
\ MENU \ SETTINGS \ CONTROL \ START STOP RESET			
Start Source	MODBUS MODBUS2	(option)	The inverter accepts CW from MODBUS (2 – simplified status automatic)
\ MENU \ SETTINGS \ CONTROL \ DESIRED FREQUENCY			
Source of demanded frequency.	MODBUS	(option)	The converter accepts REF from MODBUS as the source of the desired value
Reverse Source	According to demanded frequency	(option)	The reverse is applied according to the REF sign.
\MENU\SETTINGS\COMMUNICATION			
Inverter Address	1 – 99		The address of the inverter for communication.
IP Mode	DHCP or Manual	(option)	Method of assigning an IP address
IP address	xxx.xxx.xxx.xxx		IP address for ethernet
Mask	xxx.xxx.xxx.xxx		IP Address Mask
Gate	xxx.xxx.xxx.xxx		IP address for ethernet gateway
Port	1-65535 (502 for Modbus TCP)		Port for Modbus TCP
TCP timeout	0-360	Seconds	If no TCP request is received in the given time, the active connection is closed.
\MENU\SETTINGS\COMMUNICATION\MODBUS			
Baud RS485	9,6 – 115,2	kBps (optional)	BAUD's Communication Speed
MB Inactivity	Error	(option)	Inverter response to communication failure. With the CW suffix there is a stricter

	Warning Error CW CW Warning		condition, the inverter checks the CW and REF recovery.
MB Error timeout	5	[s]	Communication outage time for error evaluation.
MB Var. Timeout	2	[s]	Communication outage time for displaying a warning.
MB Var. Mode	Reset Stop Quick Stop Nothing	(option)	Inverter response to a communication failure warning.
Broadcast	Yes, No	(option)	The inverter can process request with address 0.
DataFormat	No swap Byte swap Word swap Byte & word swap	(option)	The format of the transmitted data, for normal registers. Specifies the order in which bytes are transferred for 16 and 32 bitdata.
\ MENU \ SETTINGS \ COMMUNICATION \ MODBUS \ MODBUS PARAMETERS			
ID0- ID32	-	SHORTCUT	Preset parameters for cyclic data exchange. Needs to be set for mode 2.
\ MENU \ DIAGNOSTICS \ Communication\ MODBUS			
Diagnostics of the transmitted data via the MODBUS RTU protocol from the inverter ice.			
\ MENU \ DIAGNOSTICS \ Communication\ RS LINES			
Diagnostics of inverter serial lines.			

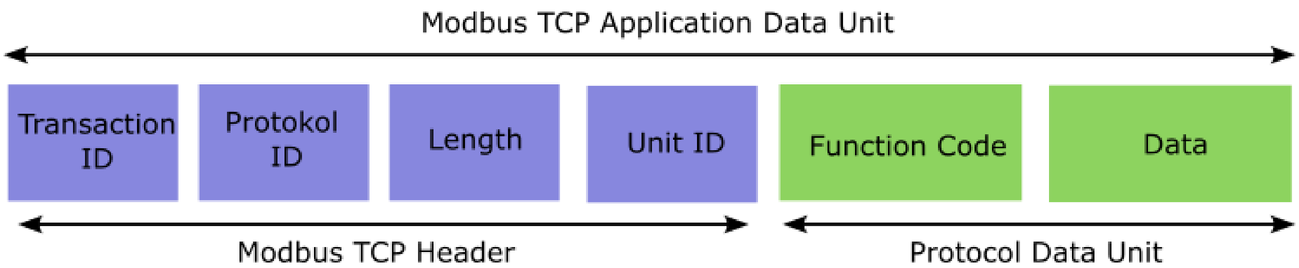
\ MENU \ DIAGNOSTICS \ Communication \ TCP

Diagnostics of transmitted data via the MODBUS TCP protocol from the inverter side.

4 Modbus TCP

4.1 Modbus TCP frame

Each TCP frame has a header called MBAP (Modbus Application Protocol). The header is followed by the PDU (Protocol Data Unit), i.e. the function code (read, write) and the data itself.



Structure of the Modbus TCP frame

Explanation of the Modbus TCP Frame:

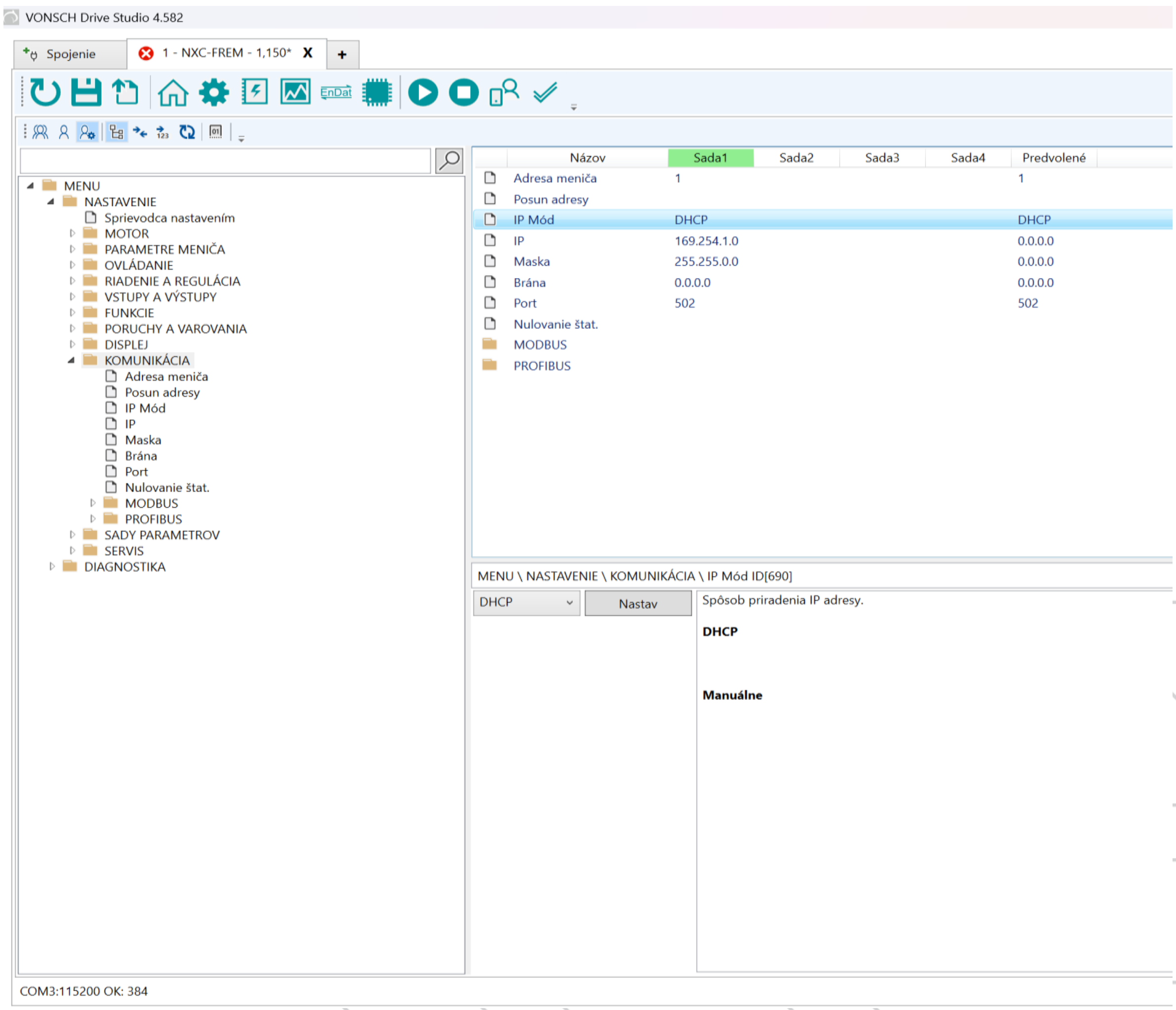
Field	Size (bytes)	Description
Transaction ID	2	Transaction ID
Protocol ID	2	Always 0 (for Modbus)
Length	2	Number of following bytes
Address	1	Matches slave address (from RTU)
Function Code	1	Same as RTU (e.g.0x03, 0x10)
Data	X	Payload (addresses, values)

4.2 Setting the IP mode and IP address of the change

From the factory, the inverters have a DHCP (Dynamic Host Configuration Protocol) mode, which automatically obtains and sets the inverter's IP address after connecting the inverter to the network.

To find out the current IP address of the inverter, or to change the IP address or IP mode, it is advisable to use Vonsch Drive Studio (VDS4). For more information, read the [Vonsch Drive Studio 4 manual].

In VDS4, after connecting the inverter to a PC using a USB-A to USB-C cable and starting communication, we find the current IP mode and IP address in the SETUP / COMMUNICATION folder.



If we want to change the IP address of the device, we first need to change the IP mode to MANUAL and then change the IP address to the one we want. If we already know the IP address of our inverter, or we have just changed it, we can close the current tab and go back to the connection tab, select the Modbus TCP connection type and write the IP address that we have just set or found out in the IP address window.

5 Annex

5.1 Example of communication via Modbus TCP with the mbpool application

5.1.1 MBPOOL

mbpoll is an open-source command-line tool for testing and diagnosing Modbus devices, which supports both Modbus RTU (over serial) and Modbus TCP (over network). It is mainly used to read/write registers in Modbus slave devices.

Basic features of mbpoll:

- Works as a Modbus master (client)
- Supports both modbus RTU and modbus TCP
- It allows reading and writing of holding registers, input registers, digital inputs/outputs, as well as setting the slave ID of the device, timeout or repeating requests
- Available for Linux, Windows and macOS.

5.1.2 Read Holding registers 0x03 via Modbus TCP

The mbpoll command via the command line **can be used to read** , which reads the register value, for example, 1518, which is the parameter ID 759 (voltage 3.3 volts).

mbpoll 169.254.1.0 -p 502 -a 1 -1 -0 -r 1518 -t 4:float -B

After successfully read in the command line, we should see the value of the register:

```
-- Polling slave 1...
[1518]:          3.32454
```

Thus, the current voltage value is 3.32454 Volts.

Explanation of the mbpoll command:

Parameter	Meaning
Mbpoll	Tool for communication via Modbus RTU/TCP
169.254.1.0	IP address of the Modbus TCP device (slave)
-p 502	TCP port – standard for Modbus TCP
-a 1	Slave address of the device (Unit ID = 1)
-1	Uses Modbus TCP (not RTU)
-0	Read only once (instead of repeating in a cycle)
-r 1518	Register (holding register) with address 1518 (PAR ID 759 = Power supply 3.3 volts)
-t 4:float	Data type : 32-bit float (Modbus type 4 = holding register)
-B	Big-endian byte order

A detailed example of reading the current voltage value of 3.3V ID 759, for the NEXICONTROL converter (address 1).

*Register = 0x2000 * (set – 1) + 2 * ID = 759 = 0x05EE*

TCP Request:

Header: 00 01 00 00 00 06 01

Application data: 03 05 EE 00 02

FULL : 00 01 00 00 00 06 01 03 05 EE 00 02

Explanation of TCP frame to read holding registers:

Field	Value	Description
Transaction ID	0x0001	Request number
Protocol ID	0x0000	Modbus TCP standard
Length	0x0006	Number of following bytes, rest of frame: Unit ID + PDU (6 bytes)
Unit ID	0x01	Slave Address (Unit ID)
Function code	0x03	Reading holding registers
Starting Address	0x05EE	Register 1518
Quantity of Regs	0x0002	2 registers (because 32 bit float = 2x16 bit)

Transaction ID		Protocol ID		Length		Unit ID	Function code	Starting Address	Quantity of registers	
0x00	0x01	0x00	0x00	0x00	0x06	0x01	0x03	0x05EE	0x00	0x02

Response:

00 01 00 00 00 07 01 03 04 40 55 15 F8

Transaction ID		Protocol ID		Length		Unit ID	Function code	Number of bytes	Data			
0x00	0x01	0x00	0x00	0x00	0x07	0x01	0x03	0x04	0x40	0x55	0x15	0xF8

The conversion 0x405515F8 according to IEEE 754 is 3.32454Hz.

5.1.3 Write Holding register 0x10 via Modbus TCP

The following command can be used to write to the register 2514, which is the parameter ID 1257 (the desired AC power for NEXICONTROL), the value 4500.5.

mbpoll 169.254.1.0 -p 502 -a 1 -1 -0 -r 2514 -t 4:float -B -W 4500.5

Explanation of the mbpoll command:

Parameter	Meaning
Mbpoll	Tool for communication via Modbus RTU/TCP
169.254.1.0	IP address of the Modbus TCP device (slave)
-p 502	TCP port – standard for Modbus TCP
-a 1	Slave address of the device (Unit ID = 1)
-1	Uses Modbus TCP (not RTU)
-0	Read only once (instead of repeating in a cycle)

-r 2514	Register (holding register) with address 2514 (PAR ID 1257 = Desired AC power)
-t 4:float	Data type : 32-bit float (Modbus type 4 = holding register)
-B	Big-endian byte order
-W	Writes the following value to the registry

Header: 00 01 00 00 00 0B 01

Application data: 10 09 D2 00 02 04 45 90 28 00

FULL: 00 01 00 00 00 0B 01 10 09 D2 00 02 04 45 90 28 00

Explanation of the TCP frame to write holding registers:

Field	Value	Description
Transaction ID	0x0001	Request number
Protocol ID	0x0000	Modbus TCP standard
Length	0x000B	Number of following bytes
Unit ID	0x01	Slave Address (Unit ID)
Function code	0x10	Write multiple registers
Starting Address	0x09D2	Register 1518
Quantity of Regs	0x0002	2 registers (because 32 bit float = 2x16 bit)
Byte Count	0x04	4 bytes of data (2 registers)
Register data	0x45902800	4500,5

Transaction ID		Protocol ID		Length		Unit ID
0x00	0x01	0x00	0x00	0x00	0x0B	0x01

Function code	Starting Address		Number of registers		Byte count	Register data			
0x10	0x09	0xD2	0x00	0x02	0x04	0x45	0x90	0x28	0x00

Answer:

The server responds with the same packet for confirmation.

Header: 00 01 00 00 00 0B 01

Application data: 10 09 D2 00 02 04 45 90 28 00

FULL: 00 01 00 00 00 0B 01 10 09 D2 00 02 04 45 90 28 00

Transaction ID		Protocol ID		Length		Unit ID
0x00	0x01	0x00	0x00	0x00	0x0B	0x01

Function code	Starting Address		Number of registers		Byte count	Register data			
0x10	0x09	0xD2	0x00	0x02	0x04	0x45	0x90	0x28	0x00