Bonfiglioli Vectron

RPS Communication

Quick Guide

Bonfiglioli Forever Forward

Content

1 Introduction

Modbus definitions for reading/writing data in a device do not exactly fit parameter access of inverters. Therefore it is necessary to map the register to the parameter number.

The current inverter software supports access to 16-Bit and 32-Bit parameters via Modbus Function Code 0x03 Read Holding Registers, 0x06 Write Single Register and 0x10 Write Multiple Registers.

The Modbus Block Read function allows the user to read up to 64 16-Bit parameters in one block transfer.

For the Block Read function a different access method is required, which does not conflict with the current method of accessing parameters.

The Modbus Block Write function allows the user to write up to 32 16-Bit parameters in one block transfer.

2 Modbus (RTU)

2.1 Single read / write parameter

2.1.1 Frame Description

Request:

Address	Function	Dataset/Parameter	Number of Register	CRC

Read response:

Address	Function	No. Bytes	Parameter Value	CRC

NOTE

The whole code is written in hexadecimal digits.

The **Address** field contains the slave address in the Modbus message. Valid slave node addresses are in the range of 1-247. A master addresses a slave by placing the slave address in the address field of the message. When the slave returns its response, it places its own address in the response address field to let the master know which slave is responding.

The Address is set in Parameter **1376**.

The **Function** Code indicates to the inverter what kind of action to perform. The most important functions for Bonfiglioli RPS Inverters are:

- Function 03 (0x03) to read out 16 and 32bit parameter / Block read
- Function 06 (0x06) to write single 16bit parameter
- Function 16 (0x10) to write single 32bit parameter / Block write

Dataset / Parameter is using the following algorithm:

DataSet/Parameter = parameter number + (data set * 4096)

The Datasets are connected as follows:

EEPROM	RAM
Dataset 0	Dataset 5
Dataset 1	Dataset 6
Dataset 2	Dataset 7
Dataset 3	Dataset 8
Dataset 4	Dataset 9

The actual Values are always stored in Dataset 0 respectively Dataset 5.

Writing to Dataset 0 means that the same value will be written to Dataset 1 to 4. Writing to Dataset 5 means that the same value will be written to dataset 6 to 9.

For example, in order to write dataset 5 of parameter 830 (830 + 5*4096 = 20702) you have to put **533E**.

If it is necessary to write parameter **permanently / automatically** to the inverter, it is **indispensable** to write them to dataset 5, 6, 7, 8 or 9. These Datasets are located on the internal RAM, which will be **erased** after a reboot/shutdown of the inverter. If you wrote continuously to dataset 0 to 4, you would write to the internal EEPROM, which is limited to 1000000 writings. The only exception is, when you have to set a parameter manually/once to a specific value, to do adjustments or similar.

CAUTION

Inverter damage

Inverter damage due to inappropriate addressing.

- Do NOT use dataset 0-4 for automatic write processes.

For automatic write processes, it is necessary to use dataset 5-9 and **you must not use dataset 0-4**.

Otherwise you will destroy the controller of the AEC and it will need to be replaced.

The inverter is always working in dataset 1 respectively 6. To write parameters it is recommended to use data set 5.

No.Bytes: For 16bit values there will be 2 Bytes; for 32bit values there will be 4 Bytes

Number of Register: To read out or to write 16bit values the number of register is 1. For 32bit values the number of register is 2.

Parameter Value: The value of the read out value or the new value of the parameter.

The **CRC** is the result of a "Redundancy Checking" calculation that is performed on the message contents. To calculate the **C**yclic **R**edundancy **C**heck please check http://www.simplymodbus.ca/

2.1.2 Examples

2.1.2.1 Read 16bit Register/Parameter

Example: read parameter 213 Active Power (16bit) Dataset 0 of Inverter with Address 1.

Request:

Address	Function	Dataset/Parameter		Number of Register		CRC	
01	03	00	D5	00	01	95	F2

Response of Inverter with Address 1 with 30 kW:

Address	Function	No.Bytes	Parameter Value		CRC	
01	03	02	01	2C	B8	09

2.1.2.2 Write 16bit Register/Parameter

Example: set the parameter's *Power reduction reference* **1020** value (16bit) Dataset 5 of Inverter with Address 1 to 85%.

Request:

Address	Function	Dataset/Parameter		Parameter Value		CRC	
01	06	53	FC	00	55	98	81

Response:

Address	Function	Dataset/Parameter		Parameter Value		CRC	
01	06	53	FC	00	55	98	81

2.1.2.3 Read 32bit Register/Parameter

Example: read the parameter *Response Time TDG Undervoltage 2* **1201** (32bit) Dataset 0 of Inverter with Address 1

Request:

Address	Function	Dataset/Parameter		Number of Register		CRC	
01	03	04	B1	00	02	95	1C

Response of Inverter with Address 1 with 200 ms:

Address	Function	No.Bytes	Parameter Value				CRC	
01	03	04	00	00	00	C8	FB	A5

2.1.2.4 Write 32bit Register/Parameter

Example: set the parameter *Response Time TDG Undervoltage 2* **1201** (32bit) Dataset **5** of Inverter with Address 1 to 25000 ms.

Request:

Address	Function	Dataset / Parameter		No. of Register		No. of Bytes	Parameter Value				CRC	
01	10	54	B1	00	02	04	00	00	61	A8	DF	FA

Response:

Address	Function	Dataset/Para.		No. of Register		CRC	
01	10	54	B1	00	02	01	DF

2.2 Block Read Function

The Modbus Function 0x03 Read Holding Registers will be extended for the reading of up to 64 inverter parameters (16-Bit) in a single block transfer. This is used to reduce the Bus traffic.

The Modbus Block Read Function will allow the reading of up to 64 Inverter Parameters which correspond to contiguous Modbus Register from Reg.No. 0xF01 to 0xF40 (3841-3904 dec.)

As the register numbers do not directly refer to the inverter parameter numbers, a mapping of the register to inverter parameters is required.

This is carried out using two Index-Parameters with 32 indices each, each containing an inverter parameter number selected. The indices 1 – 32 of the first index parameter refer to the register numbers 0xF01 – 0xF20 and the indices of the second index parameter refer the register numbers 0xF21 – 0xF40 respectively.

All parameters corresponding to the Registers defined in the request are read in sequence and the data values then stored in the “Data” field of the Response Frame.

2.2.1 Possible Block Read Parameter

To define the Index 1-32 of parameters *Block read parameters* **1282** and *Block read parameters 2* **767** the VTable function of VPlus can be used. It is possible to set each 16 bit parameter that can be read. Note that only dataset 0 can be read. String parameter and 32 bit parameter are not possible.

The Inverter Parameter **P.301** and **P.850** are 32-bit Parameter, but can also be used in block mode. They are converted to 16-Bit.

The frequency (**P.850**) value has 4 decimal places and is transferred without the decimal point, i.e. the value is multiplied by 10000 (e.g. a frequency value of 50,0025 Hz is stored as 500025 which corresponds to 0x7A139 in HEX format. In order to convert this value in 16-Bits it will be cut to two decimal places, i.e. the stored value will be divided by 100. Using the example above 500025 (50,0025Hz) will be transferred as 5000 (50,00Hz) thus losing the last two decimal places.

The Active Energy (**P.301**) has no decimal places. The convention is done by dividing the value by 1000 respectively by changing from kWh to MWh (e.g. an active Energy value of 6189 kWh will be transferred as 6MWh.

In addition to the Parameters the following value can be mapped to the Modbus read mode:

2000 / 2001, "Low and High Word of P301 (Energy positive)"

3000 / 3001, "Low and High Word of P244 (Working hours)"

4000 / 4001, "Low and High Word of P345 (Operating hours)"

5000 / 5001, "Low and High Word of P302 (Energy negative)"

2.2.2 Frame Description

Request:

Address	Function	Dataset/Parameter		Number of Register		CRC	
01	03	0F	01	00	04	16	DD

Response:

Address	Function	No.Bytes	Data			CRC	
01	03	04	00	…	50	AA	FF

Address and **Function** see above

Dataset/Parameter: Start position to read out the predefined *Block read* Parameter **1282**.

"0F 01" for Index 1 – "0F 20" for Index 32

Number of Register: number of parameters to read out.

No.Bytes: depends on the number of the requested parameters (0x02 – 0x40)

Data:

Hi-Byte 1st.Par.	Lo-Byte 1st Par.	…	Hi-Byte Nth. Par.	Lo-Byte Nth par.

N = No. of Registers

2.2.3 Factory Settings

To change factory settings via Modbus use parameter **1280** to change the register and next step is to change the parameter **1282** or **767**.

P.1282 Index	Factory Setting	Corresponding Modbus Reg.No
1	213 – Active Power	0xF01
2	222 - DC-Link Voltage	0xF02
3	255 – Heat Sink Temperature	0xF03
4	256 – Inside Temperature	0xF04
5	260 – Current Error	0xF05
6	852 – Power Supply Current	0xF06
7	853 – Power Supply Voltage	0xF07
8	855 – DC Power	0xF08
9	879 – Apparent Power	0xF09
10	1090 – Solar Status	0xF0A
11	850 – Frequency	0xF0B
12	250 – Digital Inputs	0xF0C
13	860 – DC Current	0xF0D
14	301 – Active Energy	0xF0E
15	875 – Apparent Power a	0xF0F
16	876 – Apparent Power b	0xF10
17	877 – Apparent Power c	0xF11
18	869 – Active Power a	0xF12
19	870 – Active Power b	0xF13
20	871 – Active Power c	0xF14
21	863 – Current a	0xF15
22	864 – Current b	0xF16
23	865 – Current c	0xF17
24	866 – Power Supply Voltage a	0xF18
25	867 – Power Supply Voltage b	0xF19
26	868 – Power Supply Voltage c	0xF1A
27	872 – Reactive Power a	0xF1B
28	873 – Reactive Power b	0xF1C
29	874 – Reactive Power c	0xF1D
30	878 – Reactive Power	0xF1E
31	1161- Reference Value Reactive Power	0XF1F
32	222 - DC-Link Voltage	0xF20

P.767 Index	Factory Setting	Corresponding Modbus Reg.No
1	211 – R.m.s. Current	0xF21
2	223 – Modulation	0xF22
3	229 – Reference Percentage Value	0xF23
4	2000 – Active Energy High Word	0xF24
5	2001 – Active Energy Low Word	0xF25
6	243 – Digital Inputs hardware	0xF26
7	249 – Active Data Set	0xF27
8	251 – Analog Input MFI1A	0xF28
9	253 – Analog Input EM-S1INA	0xF29

P.767 Index	Factory Setting	Corresponding Modbus Reg.No
10	254 – Digital Outputs	0xF2A
11	257 – Analog Output MFO1A	0xF2B
12	266 – Analog Output EM-S1OUTA	0xF2C
13	277 – STO Status	0xF2D
14	222 - DC-Link Voltage	0xF2E
15	3000 – Working Hours Counter High Word	0xF2F
16	3001 – Working Hours Counter Low Word	0xF30
17	4000 – Operation Hours Counter High Word	0xF31
18	4001 – Operation Hours Counter Low Word	0xF32
19	291 – Peak Value Inside Temperature	0xF33
20	292 – Average Value Inside Temperature	0xF34
21	293 – Peak Value Irms	0xF35
22	294 – Average Value Irms	0xF36
23	295 – Peak Value Active Power, Positive	0xF37
24	296 – Peak Value Active Power, negative	0xF38
25	297 – Average Value Active Power	0xF39
26	851 – Transformer Voltage	0xF3A
27	856 – Power Factor Phase a	0xF3B
28	857 – Power Factor Phase b	0xF3C
29	858 – Power Factor Phase c	0xF3D
30	859 – Power Factor Total	0xF3E
31	861 – Active current	0xF3F
32	862- Reactive current	0xF40

2.2.4 Example

2.2.4.1 Read 4 Parameter in one Block transfer starting at Index 1

Request:

Address	Function	Dataset/Parameter		Number of Register		CRC	
01	03	0F	01	00	04	16	DD

Response:

Address	Function	No.Bytes	Data								CRC	
01	03	08	02	6C	15	85	02	46	01	A1	57	56

With factory settings:

P.213 = 02 6C → 620 →62,0kW

P.222 = 15 85 → 5509 →550,9V

P.255 = 02 46 →582→58,2°C

P.256 = 01 A1 →417→41,7°C

2.3 Block Write Function

The Modbus Function 0x16 Write Holding Registers will be extended to be able to write up to 32 inverter parameters (16-Bit) in a single block transfer. This is used to reduce the Bus traffic.

The Modbus Block Write Function will allow writing of up to 32 Inverter Parameters, which correspond to contiguous Modbus Register from Reg.No. 0xF51 to 0xF70 (3921 - 3952 dec.).

As the register numbers do not directly refer to the inverter parameter numbers, a mapping of the register to inverter parameters is required.

This is carried out using an Index-Parameter with 32 indices, each containing an inverter parameter number selected. The indices 1 – 32 of the index parameter refer to the register numbers 0xF51 – 0xF70 respectively.

All parameters corresponding to the Registers defined in the request and the data values are written in sequence.

2.3.1 Possible Block Write Parameter

To define the Index 1-32 of parameter *Block Write Parameters* **600** the Vtable function of Vplus can be used. Furthermore it is possible to change the factory settings via Modbus command. Use the parameter **1280** to change the register and next step is to change the parameter **600** to the wanted setting.

It is possible to set each 16 bit parameter that can be written.

Note that the dataset is defined the following way:

Value = Parameternumber + DS * 4096

All parameters written with Block Write Mode are not written into the EEPROM. That means that there is no difference between DS0 and DS5, DS1 and DS6 etc.

NOTE

This is only valid for Block Write Mode.

String parameter and 32 bit parameter are not possible.

2.3.2 Frame Description

Request:

Address	Function	Parameter		#Register		# Bytes	Data			CRC	
01	10	0F	51	00	04	08	00	...	00	XX	XX

Response:

Address	Function	Parameter		#Register		CRC	
01	10	0F	51	00	04	XX	XX

Address and **Function** see above.

Dataset/Parameter: Startposition to write the predefined Block write Parameter **600**.

"0F 51" for Index 1 – "0F 70" for Index 32

Number of Register: Number of parameters to write.

No.Bytes depends on the number of the requested parameters (0x02 – 0x40)

Data:

Hi-Byte 1st.Par.	Lo-Byte 1st Par.	...	Hi-Byte Nth. Par.	Lo-Byte Nth par.

N = No. of Registers

2.3.3 Factory Settings

P.600 Index	Factory Setting	Corresponding Modbus Reg.No
1 - 32	524 – Fix Percentage Value	0xF51 – 0xF70

To change factory settings via Modbus use parameter **1280**, to change the register and next step is to change the parameter **600**.

2.3.4 Example

2.3.4.1 Write 2 Parameter in one Block transfer starting at Index 1

Request:

Address	Function	Parameter		#Register		# Bytes	Data				CRC	
01	10	0F	51	00	02	04	00	01	00	02	XX	XX

With factory settings:

P.524 = 1

P.524 = 2

Response:

Address	Function	Parameter		#Register		CRC	
01	10	0F	51	00	02	XX	XX

3 Modbus TCP

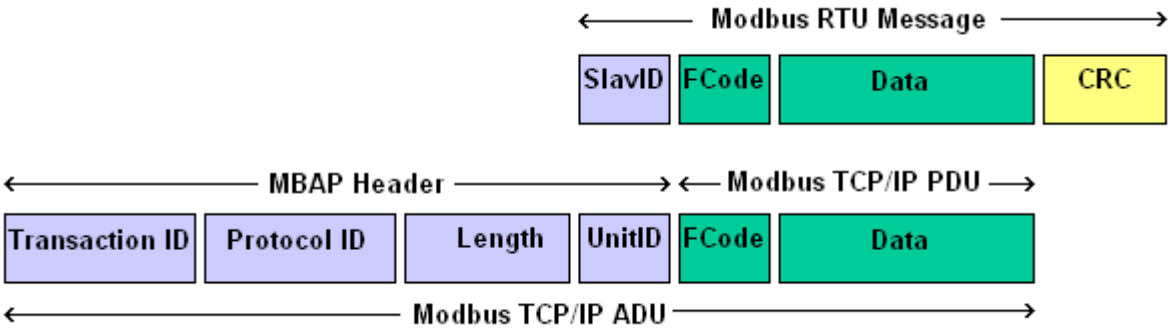
The frequency inverter can be controlled from a PLC or another master device via an Ethernet interface and a serial to Ethernet Modbus gateway, using Modbus/TCP protocol

The MODBUS/TCP communication protocol is a Client/Server related protocol. A Modbus/TCP communication is always initiated by the client (e.g. PLC). The server nodes (inverters) do not communicate with each other.

Modbus/TCP communication is established by Modbus/TCP client to TCP/IP port #502 on side of Modbus/TCP server.

3.1 Message Structure

A Modbus/TCP message is made up of the following fields:



MBAP Modbus Application Header

Field	Length	Description	Client	Server (inverter)
Transaction Identifier	2 Bytes	Identification of a Modbus request/response transaction	Initialized by client	Recopied by the server from received
Protocol Identifier	2 Bytes	0 = Modbus protocol	Initialized by client	Recopied by the server from received
Length	2 Bytes	Number of following bytes (including Unit Identifier)	Initialized by client (request)	Initialized by server (response)
Unit Identifier	1 Byte	Identification of a remote slave connected on a serial line	Initialized by client (request)	Initialized by server (response)

The **Transaction ID** is a counter, so that the master can assign the responses the slaves.

The **Protocol Identifier** is set to 0 for Modbus.

The **Length** depends on the frame Designation of the Modbus Function. For Function 0x03;0x06 and 0x65 it is always 0x0006.

The **Unit Identifier** is equal to the Slave Address of Modbus RTU.

3.2 Example

Request Parameter 222 DC-Link Voltage Dataset 0 2nd Transaction

Transaction ID		Protocol ID		Length		Unit ID	Fun. Code	Dataset / Para.		Number of Register	
00	02	00	00	00	06	01	03	00	DE	00	01

Response P.222 → 0x0063 = 9,9V

Transaction ID		Protocol ID		Length		Unit ID	Fun. Code	No. Bytes	Para. Value	
00	02	00	00	00	05	01	03	02	00	63

4 Inverter Setup

4.1 Protocol

To set the Protocol, do the following:

Under “Special Function” → “RS485/RS232” set the Parameter *Protocol* **395** to value “2 – Modbus-RTU”

In case SyCo is used this setting is not necessary and must not be set.

Bonfiglioli Vectron
Europark Fichtenhain B6
47807 Krefeld
Tel. +(49) 02151 8396 468
Fax +(49) 02151 8396 991
www.bonfiglioli.com
renewables-service@bonfiglioli.com

Bonfiglioli worldwide network

Bonfiglioli Vectron
Europark Fichtenhain B6
47807 Krefeld
Tel. +(49) 02151 8396 468
Fax +(49) 02151 8396 991
www.bonfiglioli.com
renewables-service@bonfiglioli.com

Belgium
France
Germany
United Kingdom

Bonfiglioli Brazil
Travessa Cláudio Armando 171 - Bloco 3
CEP 09861-730 - Bairro Assunção
São Bernardo do Campo - São Paulo
Tel. (+55) 11 4344 1900 - Fax (+55) 11 4344 1906
www.bonfigliolidobrasil.com.br
bonfigliolidobrasil@bonfiglioli.com

Brazil
South America
Central America

Bonfiglioli China
Unit D, 8th Floor, Building D, BenQ Plaza,
No.207 Songhong Road
Shanghai, 200335 China
Tel: +86 21 60391118
Fax: +86 59702957
email: bds.sales@bonfiglioli.com
website: www.bonfiglioli.cn

China

Bonfiglioli India
No. 543, 14th Cross, 4th Phase Peenya
Industrial Area
Bangalore, 560058 India
Tel. +91 (0) 80 2836 1014 / 2836 1015
Fax +91 (0) 80 2836 1016
www.bonfiglioli.in - brpci@bonfiglioli.com

Bangladesh
India
Malaysia
Sri Lanka

Bonfiglioli Italia
Via Sandro Pertini lotto 7b
20080 Carpiano (Milano)
Tel. (+39) 02 985081 - Fax (+39) 02 985085817
www.bonfiglioli.it
customerservice.italia@bonfiglioli.it

Bulgaria
Cyprus
Greece
Italia
Romania

Bonfiglioli USA
3541 Hargrave Drive
Hebron, Kentucky 41048
Tel. (+1) 859 334 3333 - Fax (+1) 859 334 8888
www.bonfiglioliusa.com

Canada
USA

Bonfiglioli
Forever Forward

Bonfiglioli has been designing and developing innovative and reliable power transmission and control solutions for industry, mobile machinery and renewable energy applications since 1956.

Bonfiglioli Riduttori S.p.A.
Via Giovanni XXIII, 7/A
40012 Lippo di Calderara di Reno
Bologna (Italy)

tel: +39 051 647 3111
fax: +39 051 647 3126
bonfiglioli@bonfiglioli.com
www.bonfiglioli.com

Available on the App Store
ANDROID APP ON Google play

COD.VEC 101 R1