Technical Documentation

TORRIX RS485

with MODBUS Protocol

Edition: 2020-08
Version: 6
Art. no.: 350187

FAFNIR GmbH • Schnackenburgallee 149 c • 22525 Hamburg • Tel.: +49 / 40 / 39 82 07–0 • Fax: +49 / 40 / 390 63 39

Table of contents

1 TORRIX RS485 1

2 MODBUS Register Map 2

2.1 Values formatted in 16-bit unsigned format 3

2.2 Values formatted in 32-bit floating point format 5

2.3 Exception Codes 9

3 Appendix 10

3.1 Modbus ASCII Request and Response Formats 10

3.2 Modbus ASCII LRC Calculation 11

3.3 Modbus ASCII Communication Examples 12

1 TORRIX RS485

Supported MODBUS Protocol Variants

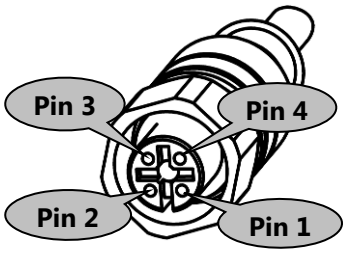
TORRIX RS485 with Modbus protocol supports the Modbus protocol variants ASCII and RTU as described in the Modicon Modbus Protocol Reference Guide. The protocol variant used by the Modbus Master is automatically recognised by the TORRIX RS485 with Modbus protocol.

Wiring Connections

The wiring of the level sensors TORRIX RS485 may only be carried out with the power disconnected. For wiring, proceed as follows:

- If not already connected, plug the coupling of the FAFNIR connection cable onto the M12 connector of the sensor head. First tighten the union nut of the M12 connector by hand and then use an open-ended spanner to tighten the nut further 180°. The tightening torque should be between 100 … 150 Ncm.
- Connect the cable coming from the evaluation unit with the FAFNIR connection cable, for example using an installation sleeve, in the following assignment:

Signal		Colour coding of FAFNIR cables	Assignment of the M-12 coupling
Voltage	+	brown	Pin 1
Communication	A	White	Pin 2
Voltage	–	blue	Pin 3
Communication	B	Black	Pin 4



Pin assignment of the coupling of the FAFNIR connection cable

☞ *On all FAFNIR devices with a RS485 2-wire interface the communication line marked with A is always the positive one (+) and the communication line marked with B is the negative one (-).*

Communication Parameters

Baud rate: 9600
Data bits: 8
Parity: none
Stop bits: 1

Modbus Slave Addresses

The Modbus slave address can be set to any value in the range of 1 to 247. The default slave address is 1.

Configuration

The configuration of the sensor (e.g. changing the slave address) is done with the "TORRIX Configuration Tool" that runs on a computer.

The configuration with a computer must not be executed within potentially explosive areas.

2 MODBUS Register Map

Supported Function Codes

The following function codes are supported:

- Function Code 03 – Read Holding Registers
- Function Code 04 – Read Input Registers
- Function Code 08 – Diagnostics (only sub-function 00 – Return Query Data)

All registers are read only.

Support of different output formats

The following output formats are supported:

16-bit unsigned integer

- Big Endian (most significant byte first): [12]
- Little Endian (least significant byte first): [21]

32-bit floating point

- Big Endian (straight word order, most significant byte first): [12] [34]
- Big Endian Bytes Swapped (straight word order, least significant byte first): [21] [43]
- Little Endian (inverse word order, least significant byte first): [43] [21]
- Little Endian Bytes Swapped (inverse word order, most significant byte first): [34] [12]

All values are available in the supported output formats. Different register areas are used for presenting the different output formats.

Support of different measurement units

The following measurement units are supported.

- Metric: mm, °C, kg/m^3 (is equal to g/l)
- US: inch, °F, lb/ft^3

All measurement values are available in the supported measurement units.
Different register areas are used for presenting the different measurement units.

2.1 Values formatted in 16-bit unsigned format

The values formatted in 16-bit unsigned format can be read out using the following two Function Codes:

- Function Code 03 – Read Holding Registers
- Function Code 04 – Read Input Registers

You must add 30001 or 40001 to the address shown in the table to get the register number. Please consider that the address shown in the table has a hexadecimal format while the register number has a decimal format.

16-bit, unsigned integer

Address		Description
Format [12]	**Format [21]**	
0x0000	0x0100	Serial number (upper digits)
0x0001	0x0101	Serial number (lower digits)
0x0002	0x0102	Firmware version (digits 1 and 2)
0x0003	0x0103	Firmware version (digits 3 and 4)
0x0004	0x0104	Protocol version
0x0005	0x0105	Probe type
0x0006	0x0106	Probe length in mm
0x0007	0x0107	Probe length in inch
0x0008	0x0108	Number of floats
0x0009	0x0109	Number of discrete temperature sensors
0x000A	0x010A	Status of probe
0x000B	0x010B	Number of density modules

Values

Protocol version

This is the version of the MODBUS protocol that is supported by the probe.

A value of e.g. 0x0103 stands for protocol version 1.03.

Existing protocol versions:

- 1.00 – first release of the MODBUS protocol
- 1.01 – minor corrections (with no functional relevance)
- 1.02 – added registers for density measurement
- 1.03 – added support for the Modbus protocol variant RTU

Probe type

- 1 = Basic
- 2 = Standard
- 3 = Advanced
- 4 = Flex

Number of discrete temperature sensors

The number of discrete temperature sensors depends on the variant of the probe. Not all probes can be equipped with discrete temperature sensors.

Status of probe

The probe can be in one of the following states:

- 0 = ok
- 1 = internal error

Number of density modules

The number of density modules that a probe can be equipped with depends on the variant of the probe. Not all probes can be equipped with density modules.

2.2 Values formatted in 32-bit floating point format

The values formatted according to the IEEE 754 single-precision 32-bit floating point format can be read out using the following two Function Codes:

- Function Code 03 – Read Holding Registers
- Function Code 04 – Read Input Registers

You must add 30001 or 40001 to the address shown in the table to get the register number. Please consider that the address shown in the table has a hexadecimal format while the register number has a decimal format.
Two consecutive 16-bit registers must be read to get the complete 32-bit floating point value. If a requested 32-bit floating point value is either not supported by the probe type or the probe has an error the returned value will be NaN (0x7FA00000).

32-Bit Floating Point, Measurement Units: Metric (mm, °C, kg/m^3)

Address				Description
Format [12][34]	**Format [21][43]**	**Format [43][21]**	**Format [34][12]**	
0x0020 0x0021	0x0120 0x0121	0x0220 0x0221	0x0320 0x0321	Level of product (upper float)
0x0022 0x0023	0x0122 0x0123	0x0222 0x0223	0x0322 0x0323	Level of water (lower float)
0x0024 0x0025	0x0124 0x0125	0x0224 0x0225	0x0324 0x0325	Average temperature
0x0026 0x0027	0x0126 0x0127	0x0226 0x0227	0x0326 0x0327	Temperature of temperature sensor 1 * (position near the bottom of the probe)
0x0028 0x0029	0x0128 0x0129	0x0228 0x0229	0x0328 0x0329	Temperature of temperature sensor 2 *
0x002A 0x002B	0x012A 0x012B	0x022A 0x022B	0x032A 0x032B	Temperature of temperature sensor 3 *
0x002C 0x002D	0x012C 0x012D	0x022C 0x022D	0x032C 0x032D	Temperature of temperature sensor 4 *
0x002E 0x002F	0x012E 0x012F	0x022E 0x022F	0x032E 0x032F	Temperature of temperature sensor 5 * (position near to the top of the probe)
0x0030 0x0031	0x0130 0x0131	0x0230 0x0231	0x0330 0x0331	Location of temperature sensor 1 * (position near the bottom of the probe)
0x0032	0x0132	0x0232	0x0332	Location of temperature sensor 2 *

Address				Description
Format [12][34]	**Format [21][43]**	**Format [43][21]**	**Format [34][12]**	
0x0033	0x0133	0x0233	0x0333	
0x0034 0x0035	0x0134 0x0135	0x0234 0x0235	0x0334 0x0335	Location of temperature sensor 3 *
0x0036 0x0037	0x0136 0x0137	0x0236 0x0237	0x0336 0x0337	Location of temperature sensor 4 *
0x0038 0x0039	0x0138 0x0139	0x0238 0x0239	0x0338 0x0339	Location of temperature sensor 5 * (position near the top of the probe)
0x003A 0x003B	0x013A 0x013B	0x023A 0x023B	0x033A 0x033B	Density measured with density module 1 ** (upper density module)
0x003C 0x003D	0x013C 0x013D	0x023C 0x023D	0x033C 0x033D	Density measured with density module 2 ** (lower density module)
0x003E 0x003F	0x013E 0x013F	0x023E 0x023F	0x033E 0x033F	Location of density module 1 ** (upper density module)
0x0040 0x0041	0x0140 0x0141	0x0240 0x0241	0x0340 0x0341	Location of density module 2 ** (lower density module)

* Discrete temperature sensors are not available in all probe variants.

** Density measurement is an optional functionality and not available in all probe variants.

32-Bit Floating Point, Measurement Units: US (inch, °F, lb/ft^3)

Address				Description
Format [12][34]	**Format [21][43]**	**Format [43][21]**	**Format [34][12]**	
0x0420 0x0421	0x0520 0x0521	0x0620 0x0621	0x0720 0x0721	Level of product (upper float)
0x0422 0x0423	0x0522 0x0523	0x0622 0x0623	0x0722 0x0723	Level of water (lower float)
0x0424 0x0425	0x0524 0x0525	0x0624 0x0625	0x0724 0x0725	Average temperature
0x0426 0x0427	0x0526 0x0527	0x0626 0x0627	0x0726 0x0727	Temperature of temperature sensor 1 * (position near the bottom of the probe)
0x0428 0x0429	0x0528 0x0529	0x0628 0x0629	0x0728 0x0729	Temperature of temperature sensor 2 *
0x042A 0x042B	0x052A 0x052B	0x062A 0x062B	0x072A 0x072B	Temperature of temperature sensor 3 *
0x042C 0x042D	0x052C 0x052D	0x062C 0x062D	0x072C 0x072D	Temperature of temperature sensor 4 *
0x042E 0x042F	0x052E 0x052F	0x062E 0x062F	0x072E 0x072F	Temperature of temperature sensor 5 * (position near the top of the probe)
0x0430 0x0431	0x0530 0x0531	0x0630 0x0631	0x0730 0x0731	Location of temperature sensor 1 * (position near the bottom of the probe)
0x0432 0x0433	0x0532 0x0533	0x0632 0x0633	0x0732 0x0733	Location of temperature sensor 2 *
0x0434 0x0435	0x0534 0x0535	0x0634 0x0635	0x0734 0x0735	Location of temperature sensor 3 *
0x0436 0x0437	0x0536 0x0537	0x0636 0x0637	0x0736 0x0737	Location of temperature sensor 4 *
0x0438 0x0439	0x0538 0x0539	0x0638 0x0639	0x0738 0x0739	Location of temperature sensor 5 * (position near the top of the probe)
0x043A 0x043B	0x053A 0x053B	0x063A 0x063B	0x073A 0x073B	Density measured with density module 1 ** (upper density module)
0x043C 0x043D	0x053C 0x053D	0x063C 0x063D	0x073C 0x073D	Density measured with density module 2 ** (lower density module)

0x043E 0x043F	0x053E 0x053F	0x063E 0x063F	0x073E 0x073F	Location of density module 1 ** (upper density module)
0x0440 0x0441	0x0540 0x0541	0x0640 0x0641	0x0740 0x0741	Location of density module 2 ** (lower density module)

* Discrete temperature sensors are not available in all probe variants.

** Density measurement is an optional functionality and not available in all probe variants.

2.3 Exception Codes

The following exception codes are supported:

- 01 - Illegal Function
- 02 - Illegal Data Address
- 03 - Illegal Data Value

01 - Illegal Function

The exception code Illegal Function is returned under the following circumstances:

- A function code other than 03 , 04 or 08 is used
- The function code 08 is used with a sub-function code other than 0000

02 - Illegal Data Address

The exception code Illegal Data Address is returned under the following circumstances:

- The function code 03 or 04 is used with a starting address that is not included in the TORRIX Modbus register map
- The function code 03 or 04 is used with a correct starting address but the number of registers requested results in an address that is not included in the TORRIX Modbus register map

03 - Illegal Data Value

The exception code Illegal Data Value is returned under the following circumstances:

- The function code 03 or 04 is used and the number of registers requested is either 0 or higher than 34

3 Appendix

3.1 Modbus ASCII Request and Response Formats

Requests and Responses for Function Codes 03 and 04

Request

Start Character	Slave Address	Function Code	Register Start Address	Number of Registers	LRC	End Characters
1 char	2 chars	2 chars	4 chars	4 chars	2 chars	2 chars
':'	1 to 247	03 or 04	Address hi and lo	Number hi and lo *	***	cr and lf
0x3A	0x01 to 0xF7	0x03 or 0x04				0x0D, 0x0A

Regular Response

Start Character	Slave Address	Function Code	Byte Count	Register Value	...	Register Value	LRC	End Characters
1 char	2 chars	2 chars	2 chars	2 chars		2 chars	2 chars	2 chars
':'	1 to 247	03 or 04	**	0 to 65535		0 to 65535	***	cr and lf
0x3A	0x01 to 0xF7	0x03 or 0x04		0x0000 to 0xFFFF		0x0000 to 0xFFFF		0x0D, 0x0A

Exception Response

Start Character	Slave Address	Function Code	Exception Code	LRC	End Characters
1 char	2 chars 1 to 247	2 chars	2 chars	2 chars	2 chars
':'	1 to 247	FC + 0x80	01, 02 or 03	***	cr and lf
0x3A	0x01 to 0xF7	0x83 or 0x84	0x01, 0x02 or 0x03		0x0D, 0x0A

* Number of Registers is the number of registers requested and must be in the range of 1 to 34

** Byte Count is the number of bytes (not registers) returned

*** see Appendix LRC Calculation

3.2 Modbus ASCII LRC Calculation

The LRC of a message must be calculated as follows:

1. Sum up all bytes of the message, excluding the starting character {:} and the ending characters {cr} and {lf}. The sum is build up in a one-byte field, so all carries will be discarded.
2. Calculate the ones-complement of the sum by subtracting the sum from 0xFF.
3. Calculate the twos-complement of the sum by adding 0x01 to the ones-complement of the sum.

Example

This example shows how the LRC of the complete message {:}010300200006**D6**{cr}{lf} must be calculated.

Note:

- control characters of the message shown in brackets
- LRC of the message shown in bold characters

LRC calculation

1. Sum up all bytes without the start and end characters in a one-byte field
   0x01 + 0x03 + 0x00 +0x20 + 0x00 + 0x06 = 0x2A

2. Ones-complement of the sum (0xFF - sum)
   0xFF - 0x2A = 0xD5

3. Twos-complement of the sum (ones-complement of the sum + 0x01)
   0xD5 + 0x01 = 0xD6

⇨ The LRC of the message 010300200006 is D6 and must be placed just before the ending characters {cr} and {lf}. The complete message of this example has the format {:}010300200006D6{cr}{lf}.

3.3 Modbus ASCII Communication Examples

Read the serial number of the probe

The serial number is available in the register pairs 0x0000/0x0001 (Big Endian) and 0x0100/0x0101 (Little Endian).

Reading the serial number in Big Endian format from a probe with slave address 1 using Function Code 03 looks like this:

Request: {:}010300000002FA{cr}{lf}

Response: {:}010304000186ABC6{cr}{lf}

Evaluation:

Register 0x0000 = 0x0001

Register 0x0001 = 0x86AB

⇨ Serial number = 0x000186AB = 100011

Read all static values of the probe

Reading all static values in Big Endian format from a probe with slave address 1 using Function Code 03 looks like this:

Request: {:}01030000000BF1{cr}{lf}

Response: {:}010316000186AB05070201010000020B0B006F00020005000016{cr}{lf}

Evaluation:

Serial number, Register 0x0000/0x0001 = 0x000186AB = 100011

Firmware version, Register 0x0002/0x0003 = 0x05070201 ➔ V5.7.2.1

Protocol version, Register 0x0004 = 0x0100 ➔ 100

Probe type, Register 0x0005 = 0x0002 ➔ Standard

Probe length in mm, Register 0x0006 = 0x0B0B ➔ 2827 mm

Probe length in inch, Register 0x0007 = 0x006F ➔ 111 inch

Number of floats, Register 0x0008 = 0x0002 ➔ 2 floats

Number of discrete temperature sensors, Register 0x0009 = 0x0005 ➔ 5 temp. sensors

Status of probe, Register 0x000A = 0x0000 ➔ Status = ok

Read the level of product

The level of product is available as 32-bit floating point value in metric and US units.

Relevant registers for the level of product in metric units:

- 0x0020/0x0021 – format [12][34], Big Endian
- 0x0120/0x0121 – format [21][43], Big Endian, Bytes Swapped
- 0x0220/0x0221 – format [43][21], Little Endian, Bytes Swapped
- 0x0320/0x0321 – format [34][12], Little Endian

Relevant registers for the level of product in US units:

- 0x0420/0x0421 – format [12][34], Big Endian
- 0x0520/0x0521 – format [21][43], Big Endian, Bytes Swapped
- 0x0620/0x0621 – format [43][21], Little Endian, Bytes Swapped
- 0x0720/0x0721 – format [34][12], Little Endian

Reading the level of product from a probe with slave address 1 in metric units and format [43][21] (Little Endian, Bytes Swapped) using Function Code 04 looks like this:
Request: {:}010402200002D7{cr}{lf}
Response: {:}010404BC942B4438{cr}{lf}
Evaluation:
Register 0x0220 = 0xBC94
Register 0x0221 = 0x2B44

⇨ Level of product = 0x442B94BC = 686,324 mm

Read level of product, level of water and average temperature

The level of product is available as 32-bit floating point value in metric and US units.
Reading level of product, level of water and average temperature from a probe with slave address 1 in metric units and format [12][34] (Big Endian) using Function Code 03 looks like this:
Request: {:}010300200006D6{cr}{lf}
Response: {:}01030C442B9408434EC94A419E93C807{cr}{lf}
Evaluation:
Level of product, Registers 0x0020/0x0021 = 0x442B9408 ➔ 686,313 mm
Level of water, Registers 0x0022/0x0023 = 0x434EC94A ➔ 206,786 mm
Average temperature, Registers 0x0024/0x0025 = 0x419E93C8 ➔ 19,822 °C

FAFNIR GmbH
Schnackenburgallee 149 c
22525 Hamburg, Germany
Tel.: +49 / 40 / 39 82 07–0
Fax: +49 / 40 / 390 63 39
E-mail: info@fafnir.com
Web: www.fafnir.com