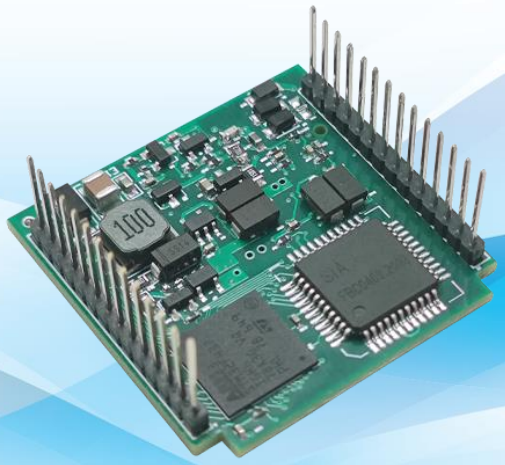
MICROCYBER

MC0313

Modbus to FF Built-in Core Module

User manual



Warning

1. It is forbidden for users to disassemble the temperature module by themselves.
2. Please check if the supply voltage of temperature transmitter meets the power supply voltage requirements in the manual.

Version：**V1.2**

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions. Any suggestions for improvement are welcome.

Company Introduction

Microcyber Corporation established as a high-tech enterprise by the Shenyang Institute of Automation Chinese Academy of Sciences, mainly engages in advanced industrial control systems, equipments, instruments and chips for industrial process automation control solutions in the research, development, production and application. Microcyber undertakes a number of national scientific and technical key task and “863” project, and has Liaoning Province networked control systems engineering research center.

Microcyber successfully developed the first domestically certified fieldbus protocol master stack, the first nationally certified fieldbus instrument, and the first domestic safety instrument certified by German TüV, and co-hosted with other units It has formulated the first domestic industrial Ethernet protocol standard EPA and the first industrial wireless communication protocol standard WIA-PA, which have become IEC international standards.

The products and technologies of Microcyber have won two second prizes of National Science and Technology Progress Award, one National Science and Technology Invention Award, one first prize of Science and Technology Progress of Chinese Academy of Sciences, and one first prize of Liaoning Province Science and Technology Progress. The products are exported to Europe and the United States, etc. In developed countries, top companies in the industry such as Emerson in the United States, Rotork in the United Kingdom, and Bifold in the United Kingdom have adopted Microcyber 's key technologies or key components in their products, and have successfully completed more than 200 large-scale automation engineering projects.

Microcyber is the member of FCG (FieldComm Group) and PNO (Profibus National Organization).
Microcyber passed the Authentication of ISO 9001 Quality System, and has an outstanding innovative R&D team, plentiful practical experiences of design of the Automatic engineering, a leading product series, a huge market network, a strict quality management system and an excellent enterprise culture. All these further a solid foundation of entrepreneurship and sustainable development for Microcyber.

Carrying the ideals of employees, creating customer value and promoting enterprise development.

Content

1 Overview

MC0313 Modbus to FF built-in core module is an built-in conversion module of Modbus RTU protocol and FF protocol developed by Microcyber Corporation. It is one of the Microcyber's M series built-in core modules. This series of built-in core board modules have the characteristics of the same size, the same interface, easy upgrade, simple configuration, etc. It is an ideal choice for users to quickly develop fieldbus devices. The MC0313 Modbus to FF built-in core module, as a Modbus host, communicates with the device with Modbus RTU communication function through TTL interface, and can convert the data in the device to FF device variable output. MC0313 Modbus to FF built-in core module, as shown in Figure 1.1 below:

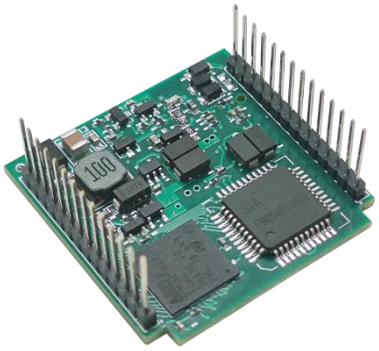


Figure 1.1 MC0313 Modbus to FF Embedded core module

1.1 Features

1.1.1 Same Size

Microcyber's MC series built-in core modules have the same size, 35mm (length) * 35mm (width).

1.1.2 Same Interface

Microcyber's MC series built-in core modules adopt 2.0 spacing double row 14 pin connectors, which are functionally compatible.

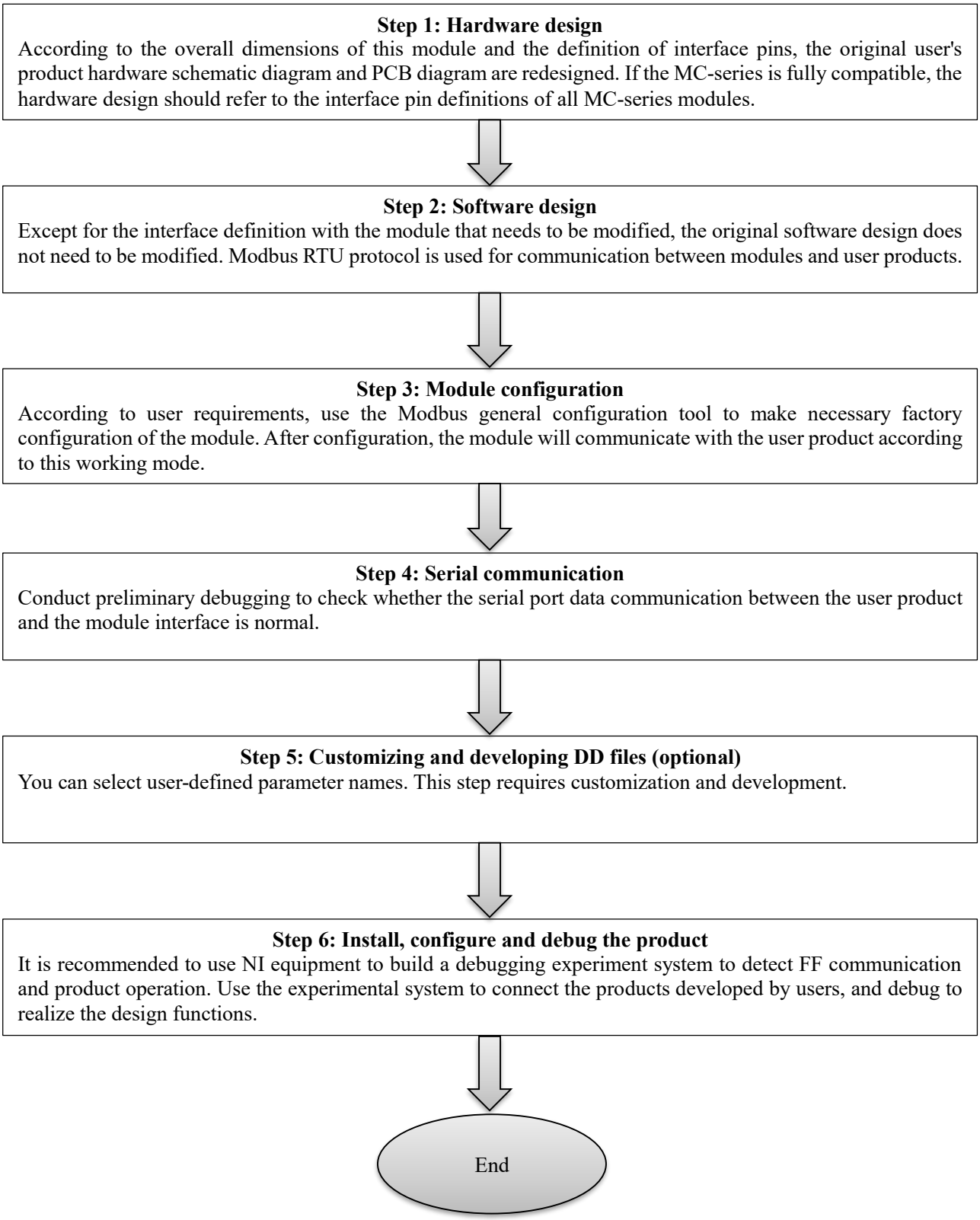
1.1.3 Easy to Upgrade

Replace different built-in core modules of Microcyber's MC series, and immediately implement devices with different protocols.

1.1.4 Simple Configuration

It is easy to operate and use by using the special configuration tool of Microcyber.

1.2 Product Development Process



1.3 Overall Dimensions

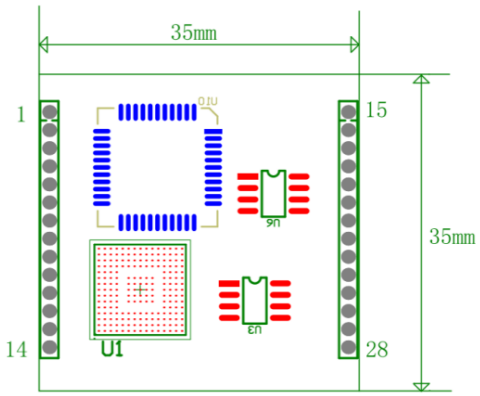


Figure 1.2 External dimensions of embedded core module (unit: mm)

1.4 Module Structure

Figure 1.3 Structure of embedded core module

2 Installation

2.1 External Interface of Module

The terminal distribution and meaning of MC0313 Modbus to FF embedded core module are shown in Figure 2.1 below:

Figure 2.1 Communication interface definition of Modbus to FF Embedded core module

2.2 Introduction to Module Interface Functions

Pin	Name	Description
1	SWDIO	Data input and output (download interface)
2	SWCLK	Clock signal (download interface)
3	NRST	MCU reset signal (download interface)
4	GND	Reference site
5	IO	Reserved GPIO
6	RXD	UART1 input TTL level
7	TXD	UART1 output TTL level
8	IO	Reserved GPIO
9	SPI1_NSS	Reserved SPI interface/GPIO

10	SPI1_MISO	Reserved SPI interface/GPIO
11	SPI1_MOSI	Reserved SPI interface/GPIO
12	SPI1_CLK	Reserved SPI interface/GPIO
13	3.3V_OUT	Power output
14	6.2V_OUT	Power output
15	IO	Reserved GPIO
16	IO	Reserved GPIO
17	WP	Write protection dial switch, any write operation to FF communication module will be rejected, which can prevent the data of FF communication module from being changed at will.
18	RST	Reset the dial switch and restore the transmitter data to the factory state. First, the FF communication module is powered off. Turn the dial switch to the ON position. The module is powered on. The FF communication module is restored to the factory state. Then turn the dial switch back to the OFF position to ensure that the next power on will not misoperate.
19	S/E	Simulation DIP switch can realize simulation function. (Simulation is only used by FF)
20	IO	UART1 is used for RTS control end of 485 communication
21	IO	Reserved GPIO
22	SCL	Reserved I2C pin/GPIO
23	SDA	Reserved I2C pin/GPIO
24	IO	Modbus communication status indicator, effective at low level
25	IO	FF communication status indicator, effective at low level
26	GND	Reference site
27	BUS+	Bus power supply positive
28	BUS-	Bus power supply negative

3 Working Principle

The MC0313 Modbus to FF embedded core module is a protocol conversion module that supports the communication protocol between Modbus and FF. As an FF device, it can communicate with the Modbus device. Through simple configuration, it can read the Modbus data into the FF device and transmit the data to the control system through the FF bus. The system connection diagram of MC0313 Modbus to FF embedded core module is shown in Figure 3.1 below:

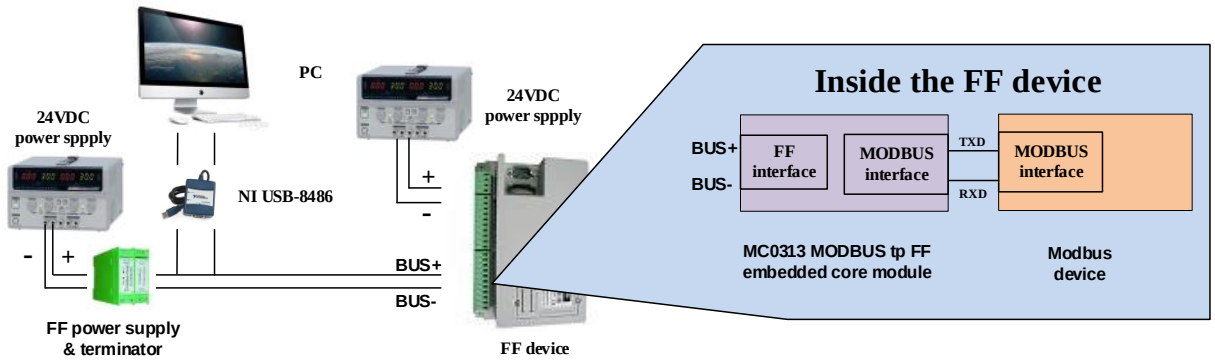


Figure 3.1 System connection diagram of MC0313 Modbus to FF Embedded core module

The MC0313 Modbus to FF embedded core module supports 1 Modbus slave, 8 analog inputs and outputs and 8 discrete inputs and outputs, providing a total of 32 channel accesses. The data collected by the Modbus device is configured to the parameters of the conversion block of the MC0313 Modbus to FF embedded core module through the Modbus register, and then provides data support for the FF system through the channel access function of variables to the AI, AO, DI and DO function blocks. The principle block diagram of MC0313 Modbus to FF embedded core module is shown in Figure 3.2 below:

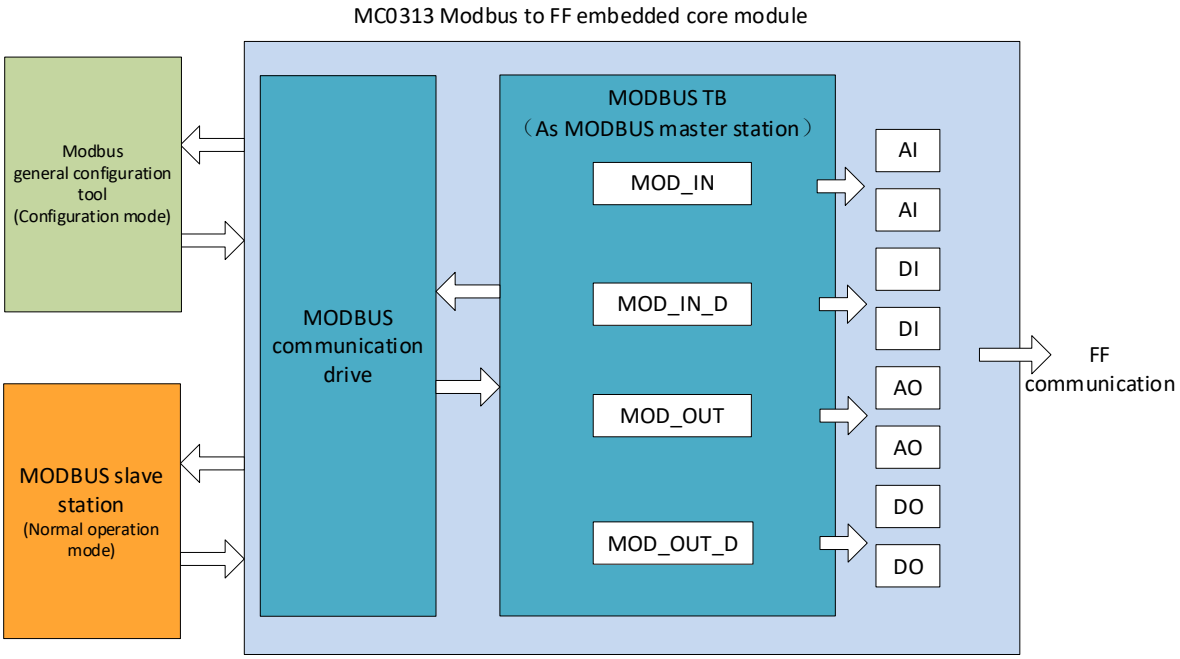


Figure 3.2 Principle block diagram of MC0313 Modbus to FF embedded core module

4 Device Configuration

4.1 Topological Connection

FF device supports multiple network topology connection modes, as shown in Figure 4.1. Figure 4.2 shows the bus connection of FF device. Both ends of the bus need to be connected to the terminal matching resistance to ensure the signal quality. The maximum length of the bus is 1900 meters, and the repeater can be extended to 10 kilometers.

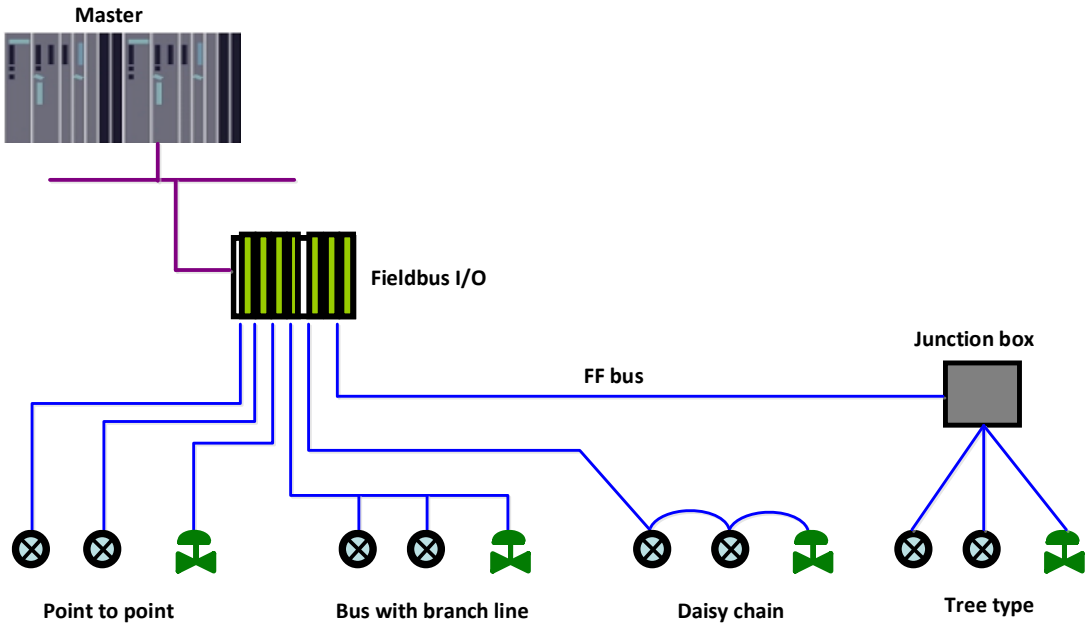


Figure 4.1 FF network topology

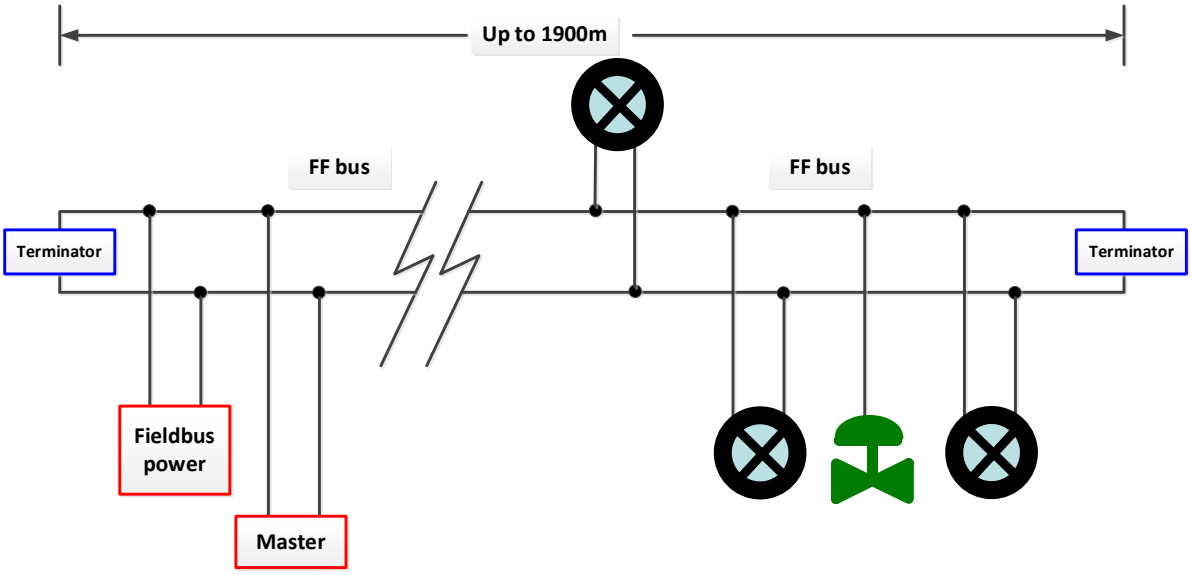


Figure 4.2 FF bus connection

4.2 Function Block Description

The default configuration of MC0313 Modbus to FF embedded core module includes 1 RES function block conforming to FF specification, 6 AI function blocks, 2 AO function blocks, 4 DI function blocks, 4 DO function blocks, 2 PID function blocks and 1 Modbus transducer block (Modbus_TB). The AI, AO, DI and DO function blocks support 8 channels (CHANNEL) respectively, and each CHANNEL can correspond to the analog/discrete input/output parameters of the Modbus transform block.

Function block name	Function block description
Resource (RES)	Resource block, used to describe the characteristics of field equipment, such as equipment name, manufacturer, and serial number. The resource block has no input or output parameters. A device usually has only one resource block.
Modbus_TB(MTB)	Modbus communication parameters can be configured through the transducer block, such as baud rate, stop bit, communication timeout, and Modbus communication configuration parameters.
Analog Input (AI)	The analog input function block is used to obtain the input data of the conversion block and can be transmitted to other function blocks. It has range conversion, filtering, square extraction and other functions.
Analog Output (AO)	The analog output function block is used to transfer the output data to the transform block and act on the physical device.
Discrete Input (DI)	Discrete quantity input function block to obtain input data of transform block and transmit it to other function blocks.
Discrete Output (DO)	The discrete output function block transfers the discrete output data to the transform block and acts on the physical device.
Proportional Integral Derivative（PID）	The proportional, integral and differential function block is a positional automatic control module, which can scale up or down the deviation, accumulate and sum. It includes a variety of functions, such as setpoint adjustment, process parameter (PV) filtering and alarm, feedforward, output tracking and other functions.

4.3 Modbus Transducer Block Parameter Description

Use the NI-Configurator software as an example to illustrate how to configure the Modbus transducer block. As shown in the figure below, transducer block provides 8 analog inputs/outputs and 8 digital inputs/outputs. These parameters are process data. Users can choose to configure them, but at least 1 analog input or 1 digital input should be configured. Floating point numbers are also provided, including 10 unsigned 32-bit, 16-bit and 8-bit numbers and 2 string parameters of 32-bit, which can be used as configuration parameters. These parameters need to be configured by Modbus universal configuration tool. They can not be configured in FF configuration software

and can only be read and written.

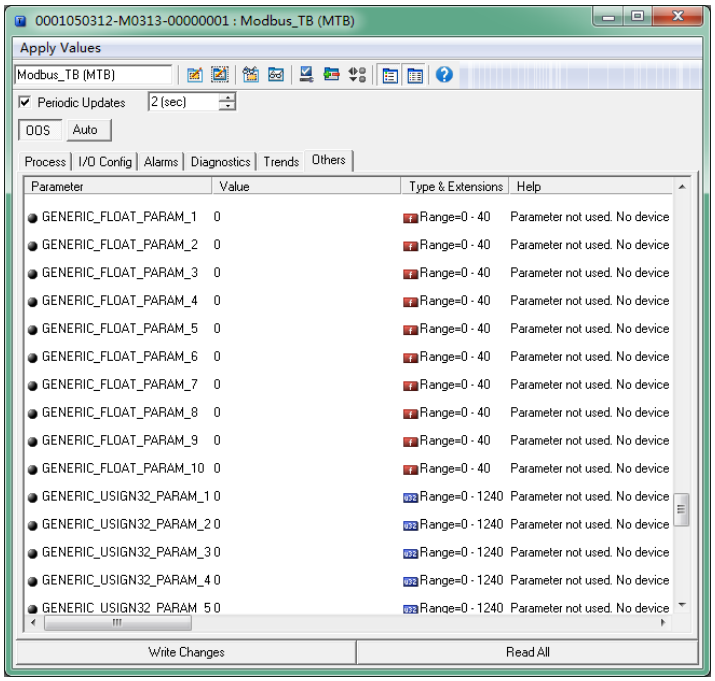


Figure 4.3 Modbus transducer block

4.3.1 BAD_STATUS Parameter Description

If communication fails, the corresponding bit is set to 1, otherwise it is 0.

Bit	Parameter description	Bit	Parameter description	Bit	Parameter description	Bit	Parameter description
0	MOD_IN1	8	MOD_OUT1	16	MOD_IN_D1	24	MOD_OUT_D 1
1	MOD_IN2	9	MOD_ OUT 2	17	MOD_IN_D 2	25	MOD_ OUT_D 2
2	MOD_IN3	10	MOD_ OUT 3	18	MOD_IN_D 3	26	MOD_ OUT_D 3
3	MOD_IN4	11	MOD_ OUT 4	19	MOD_IN_D 4	27	MOD_ OUT_D 4
4	MOD_IN5	12	MOD_ OUT 5	20	MOD_IN_D 5	28	MOD_ OUT_D 5
5	MOD_IN6	13	MOD_ OUT 6	21	MOD_IN_D 6	29	MOD_ OUT_D 6
6	MOD_IN7	14	MOD_ OUT 7	22	MOD_IN_D 7	30	MOD_ OUT_D 7
7	MOD_IN8	15	MOD_ OUT 8	23	MOD_IN_D 8	31	MOD_ OUT_D 8

4.3.2 ERR_LOOK_RESULT Parameter Description

The ERR_LOOK_RESULT parameter provides the input and output parameter negative response data query function, the user can query the response value of each input and output data, read the ERR_LOOK_RESULT parameter to get the communication response data of the parameter (the query parameter setting needs to use the Modbus general configuration tool) . ERR_LOOK_RESULT, 0 means no error, 0x01 – 0x0B is the negative response data of the Modbus standard, 0xFF means the master-slave communication failed.

Value	Parameter description	Value	Parameter description
0x00	OK	0x06	Slave Device Busy
0x01	Illegal Function	0x08	Memory Parity Error
0x02	Illegal Data Address	0x0A	Gateway Path Unavailable
0x03	Illegal Data Value	0x0B	Gateway Target Device Failed To Response
0x04	Slave Device Failure	0xFE	Function Code Mismatch
0x05	Acknowledge	0xFF	Commmunication Failure

4.4 Modbus Communication Parameter Setting

The Modbus communication parameters of MC0313 Modbus to FF embedded core module need to be set through the special Modbus general configuration tool software. The parameters are divided into 10 parts: common parameters, analog input parameters, analog output parameters, digital input parameters, digital output parameters, floating point data parameters, USIGN32 data parameters, USIGN16 data parameters, USIGN8 data parameters, Octet data parameters, etc. Users can flexibly configure various information of the Modbus slave devices to be accessed.

The configurable common parameters of the Modbus general configuration tool software are shown in Figure 4.4 below, and the specific configurable data are shown in Appendix 3.

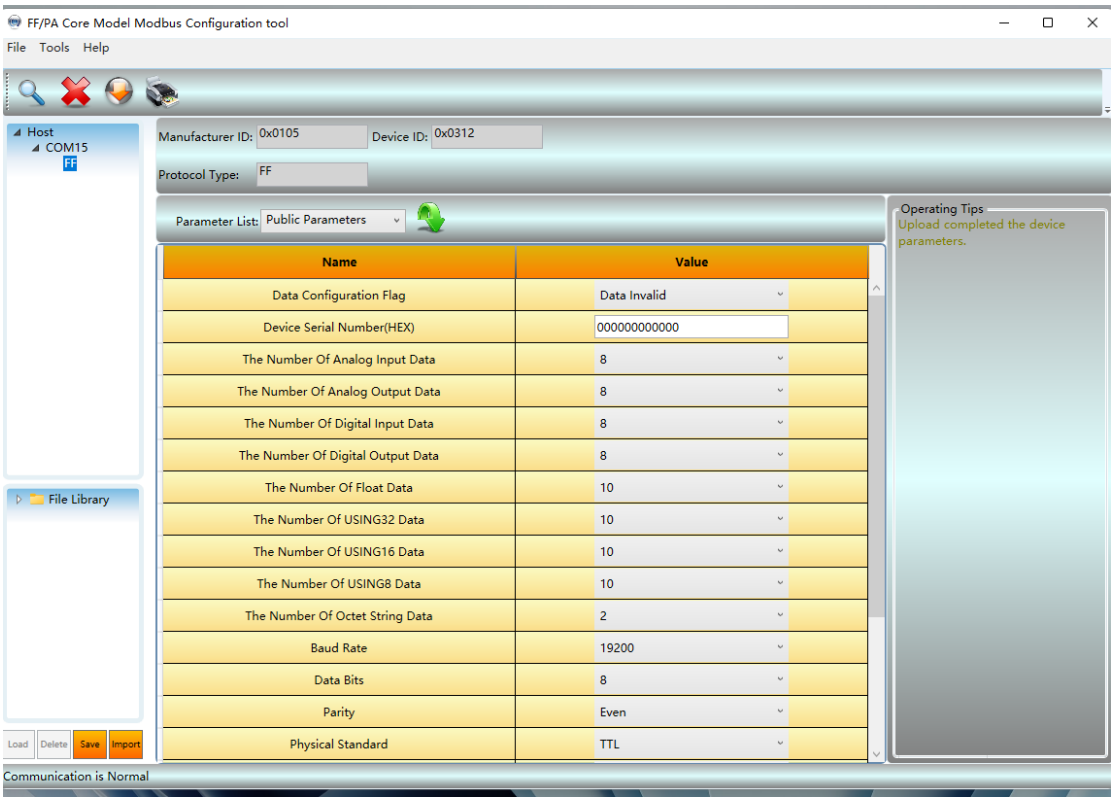


Figure 4.4 Modbus general configuration tool

Data other than public parameters can be configured with data read/write methods, data formats, register addresses and function codes. Users can flexibly configure them as needed. Since different parameter types support different Modbus function codes, corresponding data formats need to be selected after the function codes are selected. Non optional data formats are automatically grayed out by the software, and users are not allowed to select them. As shown in Figure 4.5 below:

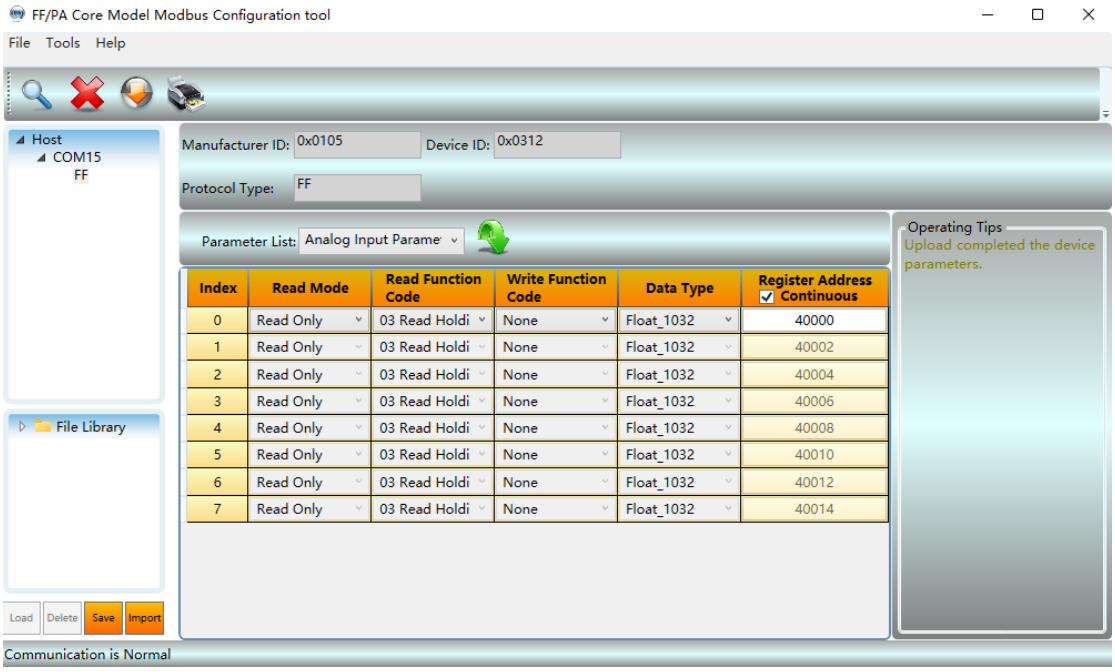


Figure 4.5 Parameter configuration

After the configuration is completed, you can choose to download the parameters separately in the data interface, or click the download icon to download uniformly.

Note: *This manual does not introduce the use method of the Modbus general configuration tool software in detail. Please refer to the software help document for the detailed use method; In the configuration mode, the embedded core module needs to be inserted into the backplane. Set the dial switch of the backplane to the configuration mode.*

4.5 Modbus Slave Configuration Example

The communication parameters of the user's Modbus slave are as follows:

NO.	Parameter	Value
1	**Baud Rate**	9600 bps
2	**Data Bits**	8
3	**Parity**	Even check
4	**Physical Standard**	TTL
5	**Stop Bits**	1
6	**Address**	1
7	**CRC**	Normal
8	**Timeout**	300ms
9	**Number of Retry**	3

The user's Modbus slave supports function code 03 (read holding register) and function code 16 (write multiple registers). The register allocation of device parameters is as follows:

Register address (decimal)	Data format	Register definition
4112	Float Inverse	Floating point representation of instantaneous flow
4114	Float Inverse	Floating point representation of instantaneous velocity

4116	Float Inverse	Floating point representation of flow percentage (battery powered meter reserved)
4118	Float Inverse	Floating point representation of fluid conductivity ratio
4120	Long Inverse	Integer part of forward cumulative value
4122	Float Inverse	Fractional part of positive cumulative value
4124	Long Inverse	Inverse cumulative integer part
4126	Float Inverse	Reverse Cumulative Numeric Fractional Part
4128	Unsigned short	Instantaneous flow unit
4129	Unsigned short	Cumulative total unit

Step 1

Check whether the 8th M of the backplane dial switch is ON. If not, switch to ON status;

Step 2

Open the Modbus general configuration tool software, establish the serial port, and scan the equipment online, as shown in the following figure:

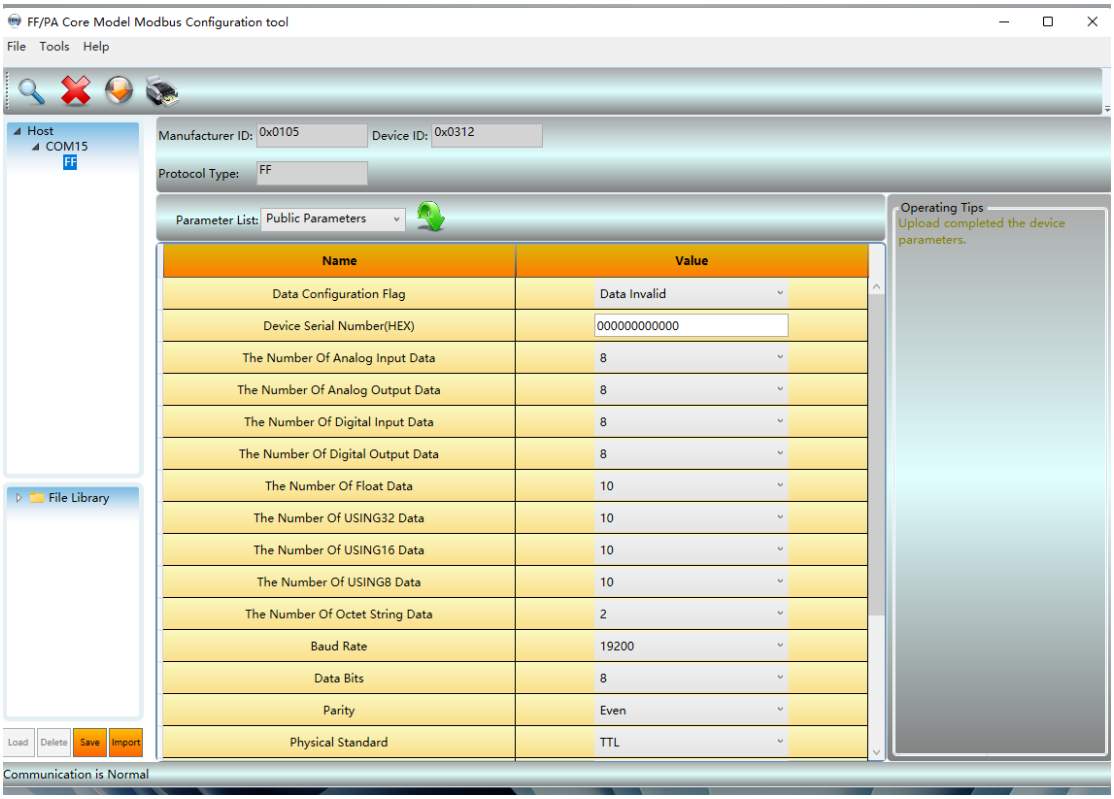


Figure 4.6 Scanning equipment

Step 3

Set each parameter in the common parameter table according to the user's Modbus slave communication parameters, as shown in the figure below

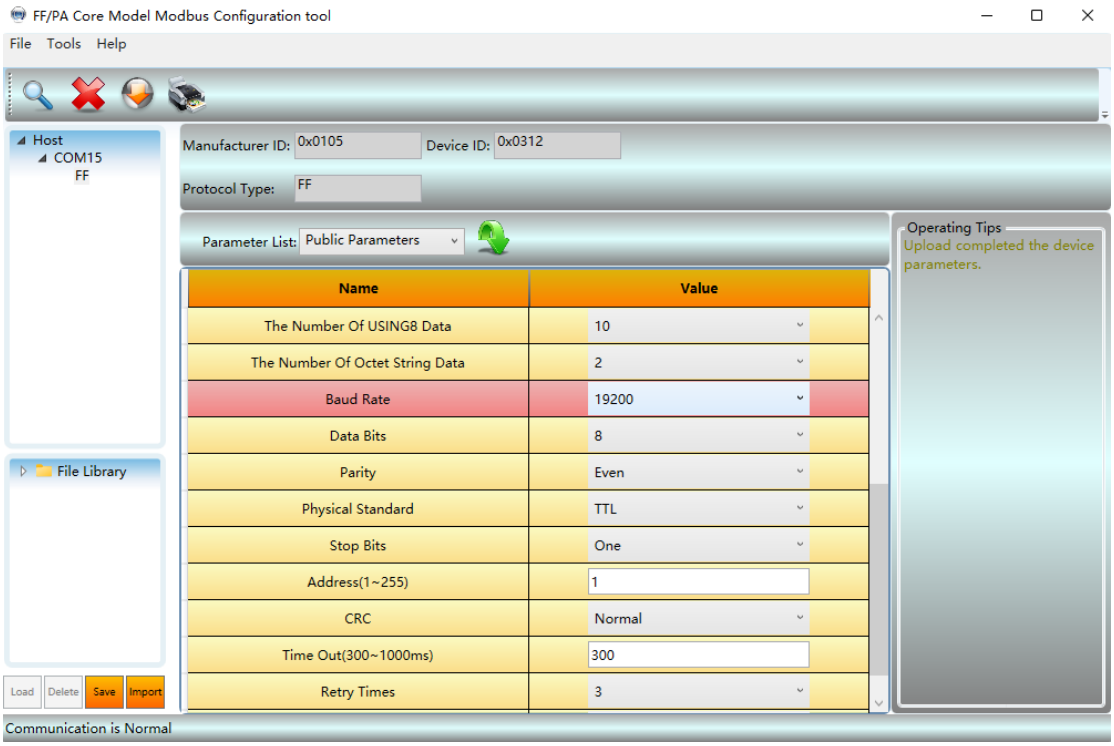


Figure 4.7 Communication parameter configuration

Step 4

Modify the common parameters according to the user equipment parameter list. The number of analog input data is 8, the number of USIGN16 parameters is 2, and the number of other parameters is 0, as shown in the following figure:

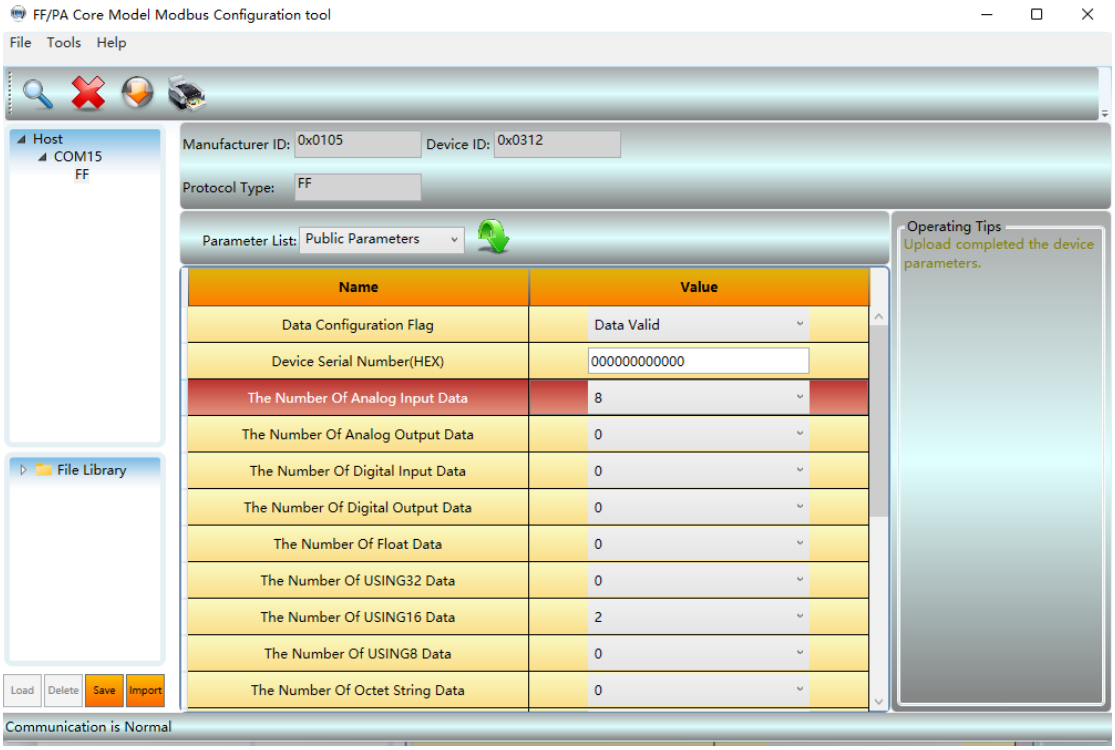


Figure 4.8 Configuration of other common parameters

Step 5

Open the analog input parameter option page, first select the function code used, then select the data type of analog input data, and finally enter the register address, as shown in the following figure:

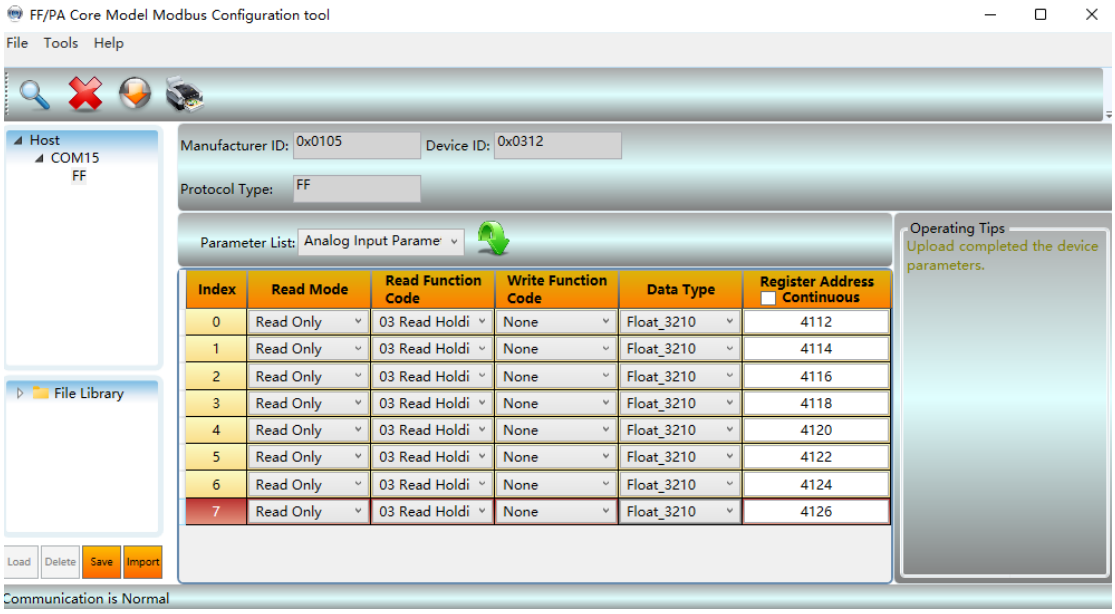


Figure 4.9 Analog input parameter configuration

Note: The register address of the data is the actual address. If the register address of the Modbus slave device given by the customer is the address address, you need to subtract 1 to get the actual address.

Step 6

On the USIGN16 data parameter option page, first select the reading/writing mode of the USIGN16 data, then select the data type and function code used, and finally enter the register address, as shown in the following figure:

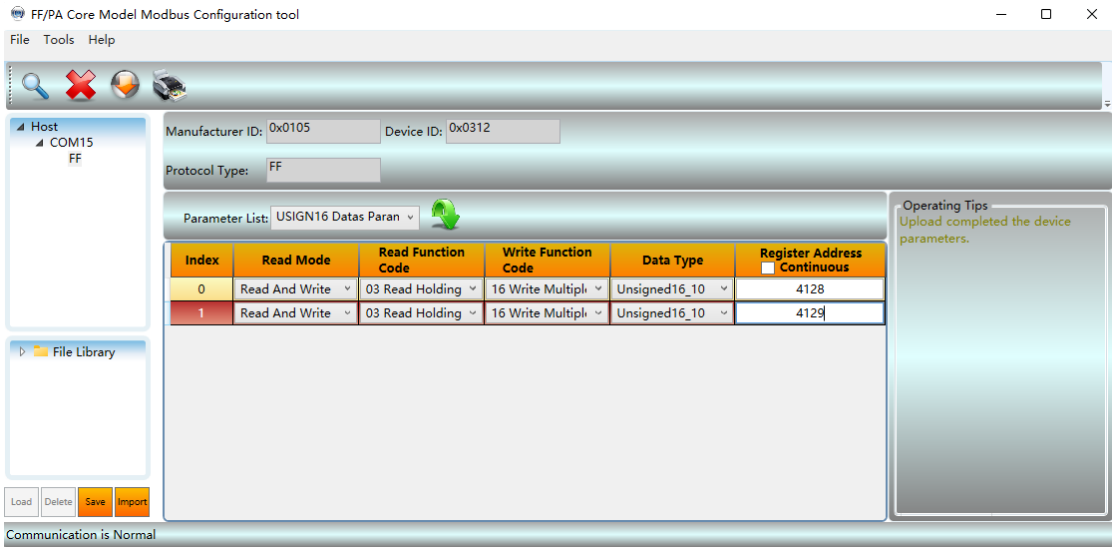


Figure 4.10 USIGN16 data parameter configuration

Step 7

Return to the public parameter option page, modify the data configuration flag to "Data Valid", click the "Batch Download" button above, and write the configuration data to the device, as shown in the following figure:

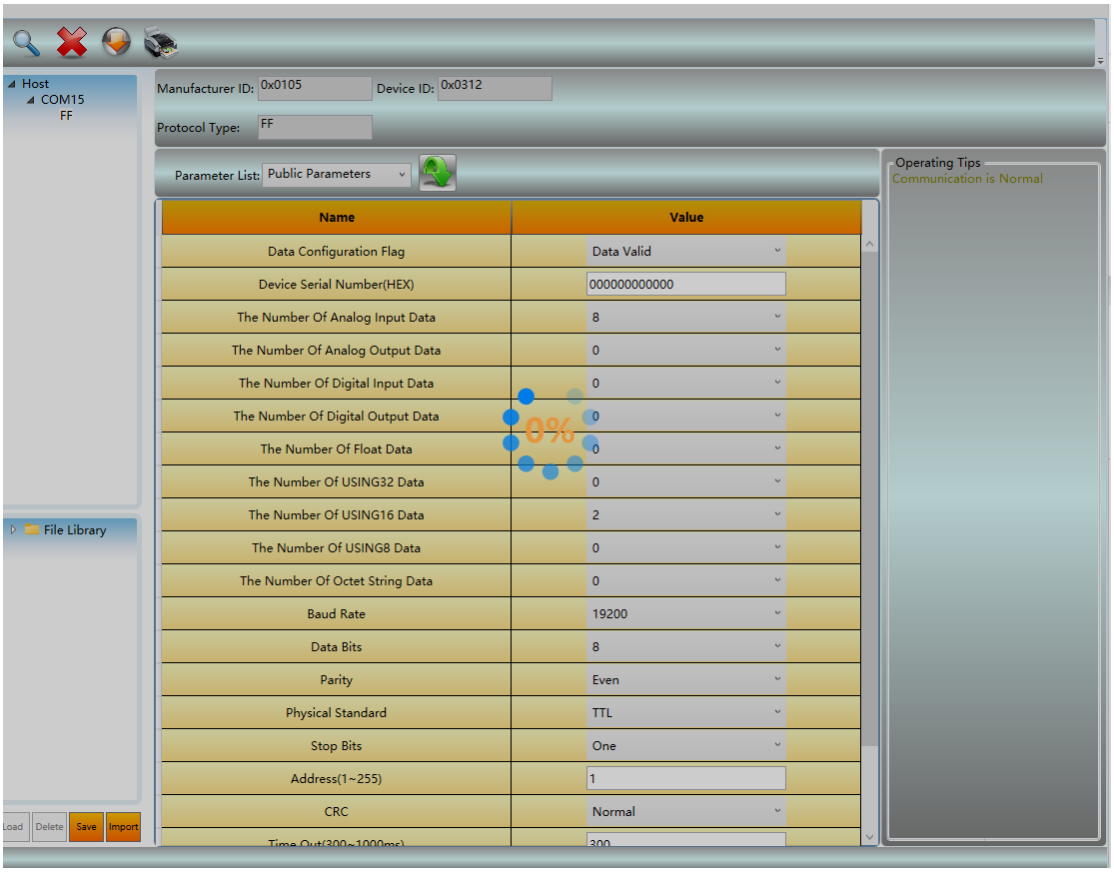


Figure 4.11 Enable data valid and download parameters

Step 8

Switch the 8th position of the bottom plate dial switch SW1 to the OFF state, connect the device to the FF network, use the NI configuration software, and check the transducer block parameters, as shown in the following figure:

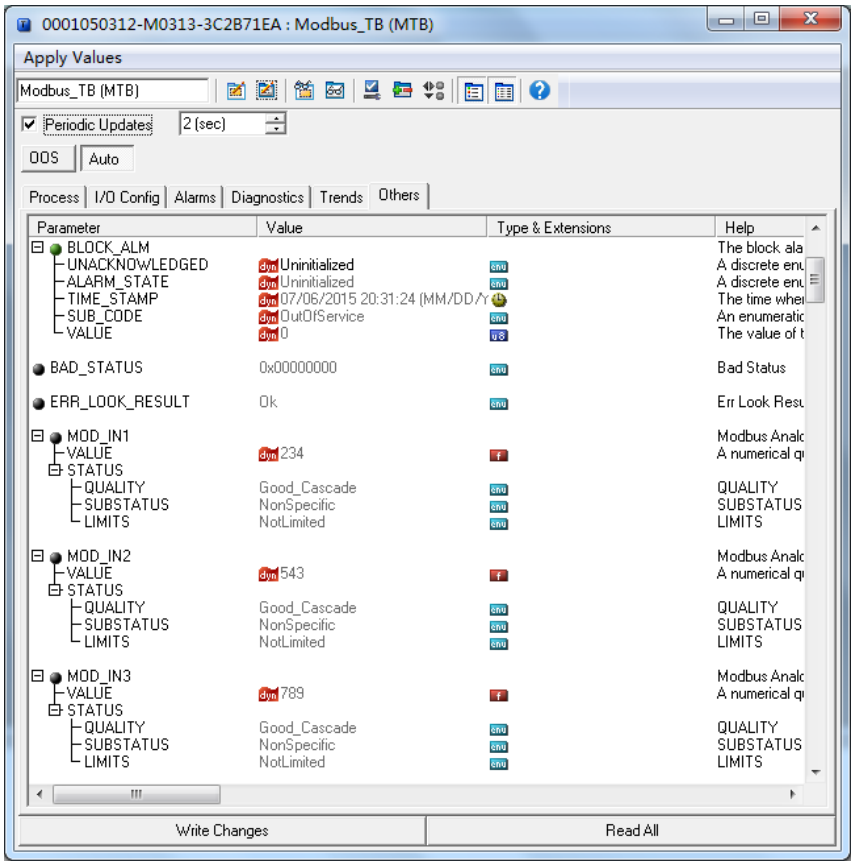


Figure 4.12 Detect Modbus slave data through FF status software

Through the above steps, the FF device can be used normally. The data of the Modbus slave device is transferred to the AI, AO, DI and DO function blocks through the transducer block parameter, which can be configured and used in the FF control system.

If the mode of the function block does not change to Auto, then the data will not be updated. In this case, the processing method for not changing the trans driver block mode to Auto in Section 4.6 should be followed.

4.6 The reason why the transducer block cannot switch to Auto status

There are many reasons why a transducer block cannot switch to an Auto state, the BLOCK_of a transducer block ERR, XD_ ERROR, BAD_ STATUS and ERR_ LOOK_ The RESULT parameter can be used to determine what the problem is.

There are many reasons why the transducer block cannot switch to the Auto state. The BLOCK_ERR, XD_ERROR, BAD_STATUS and ERR_LOOK_RESULT parameters of the transducer block can determine the kind of problem.

Situation 1

After the device is powered on, the XD_ERROR parameter value of the function block is "Configuration error", the ERR_LOOK_RESULT parameter value is "OK", and the BAD_STATUS parameter value is "0x00000000". At this time, it is necessary to check whether the eighth bit of the bottom plate DIP switch SW1 is in the OFF state. If not, please switch to OFF state (normal working mode).

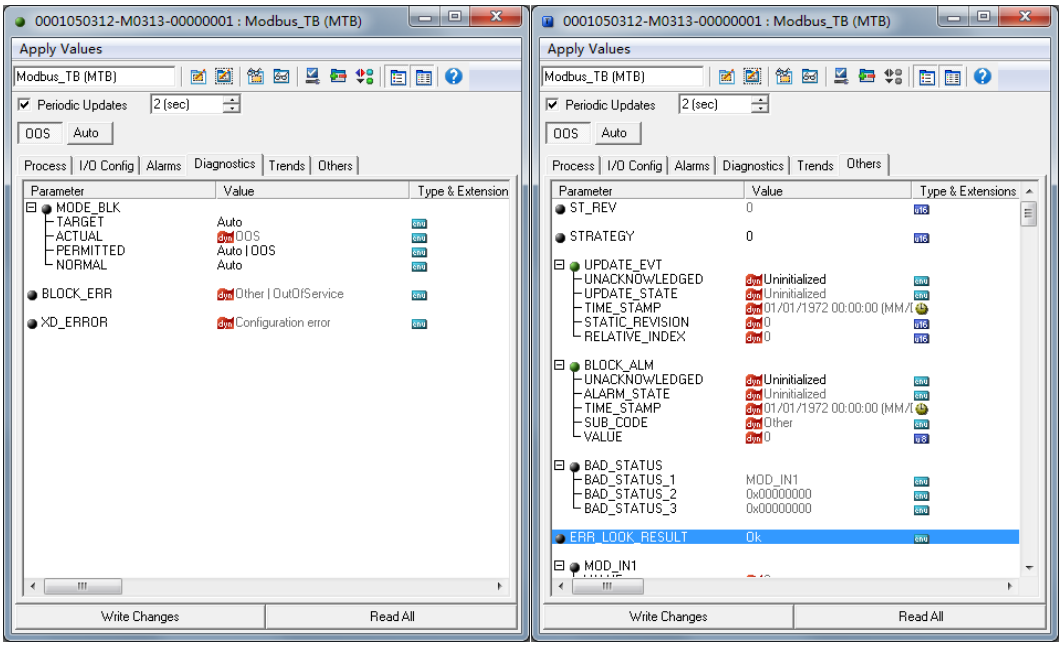


Figure 4.13 Transducer block OOS situation 1

Situation 2

After the device is powered on, the XD_ERROR parameter value of the function block is "Configuration error", the ERR_LOOK_RESULT parameter value is "Comm Failure", and the BAD_STATUS parameter has a value such as "MOD_IN1", indicates that there is a problem with the configuration of communication parameters, and the configuration of communication parameters needs to be checked through the Modbus Universal Configuration Tool.

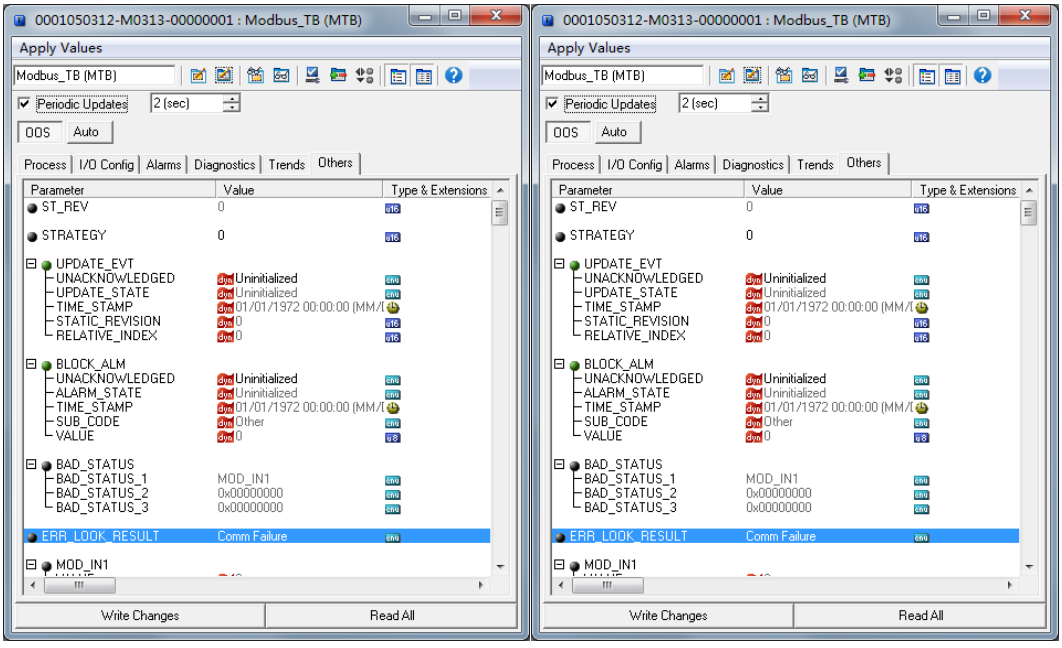


Figure 4.14 Transducer block OOS situation 2

Situation 3

After the device is powered on, the XD_ERROR parameter value of the function block is "Configuration error", the ERR_LOOK_RESULT parameter value is "Function Code Mismatch", and the BAD_STATUS parameter has a value such as "MOD_IN1", indicates that there is a problem with the configuration of the parameter's function code and the

configuration of the parameter's function code needs to be checked through the Modbus Universal Configuration Tool.

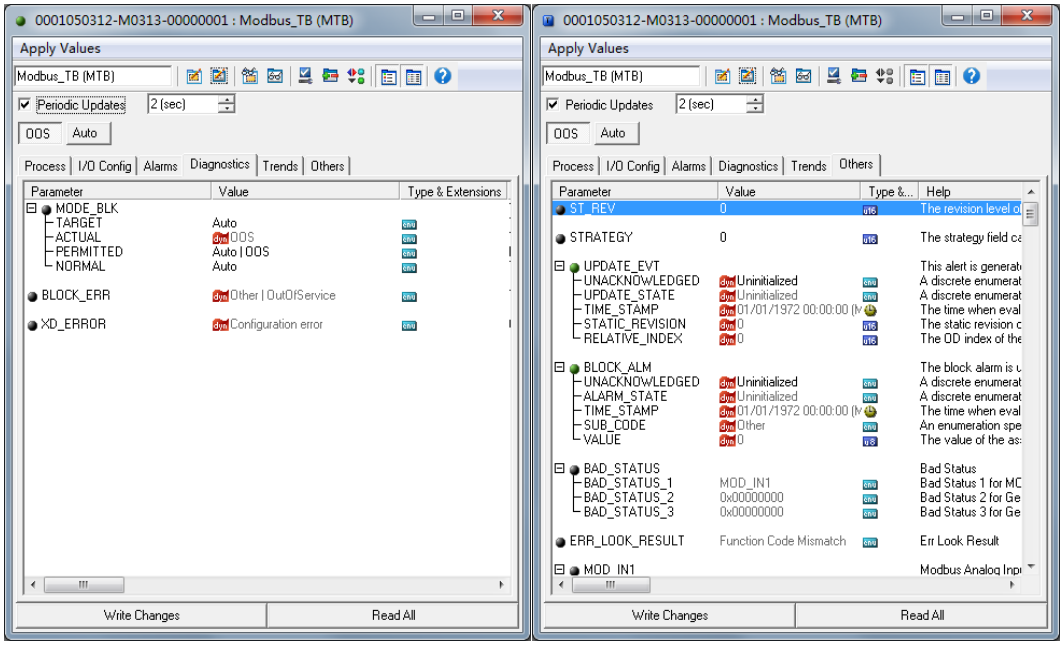


Figure 4.15 Transducer block OOS situation 3

Situation 4

After the device is powered on, the XD_ERROR parameter value of the function block is "Configuration error", the ERR_LOOK_RESULT parameter value is "Data Type Mismatch", and the BAD_STATUS parameter has a value such as "MOD_IN_D1", indicates that there is a problem with the data type configuration of the parameter, and you need to view the data type configuration of the parameter through the Modbus Universal Configuration Tool.

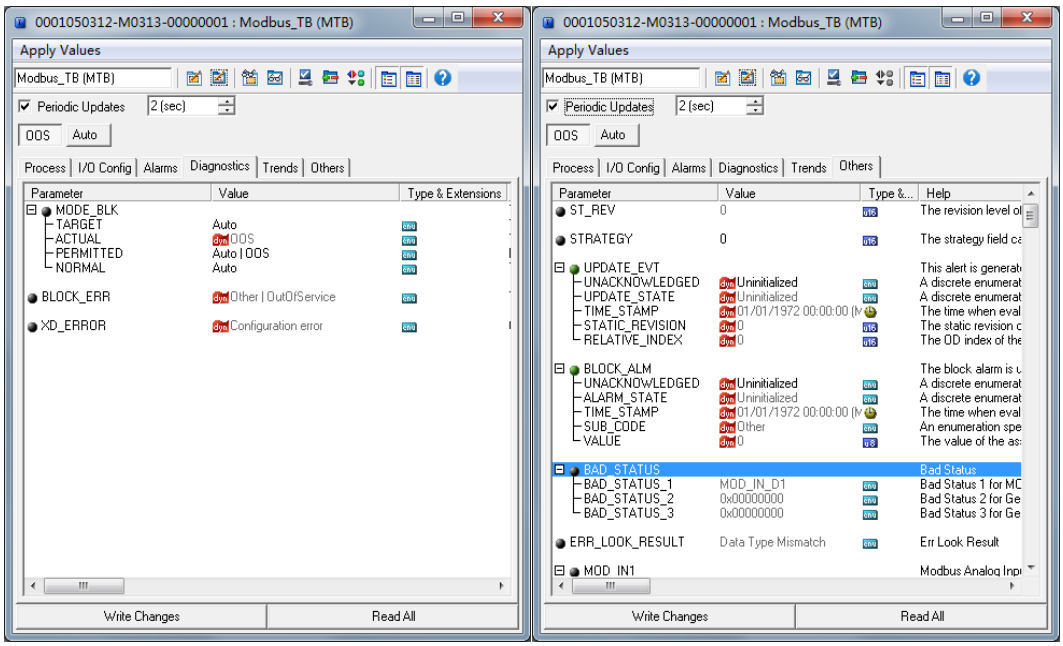


Figure 4.16 Transducer block OOS situation 4

Situation 5

After the device is powered on, the XD_ERROR parameter value of the function block is "Data Integrity Error", the

ERR_LOOK_RESULT parameter value is "Comm Failure", and the BAD_STATUS parameter has a value such as "MOD_IN1", indicates that the device has experienced a communication interruption during normal communication. Please check the connection of the device.

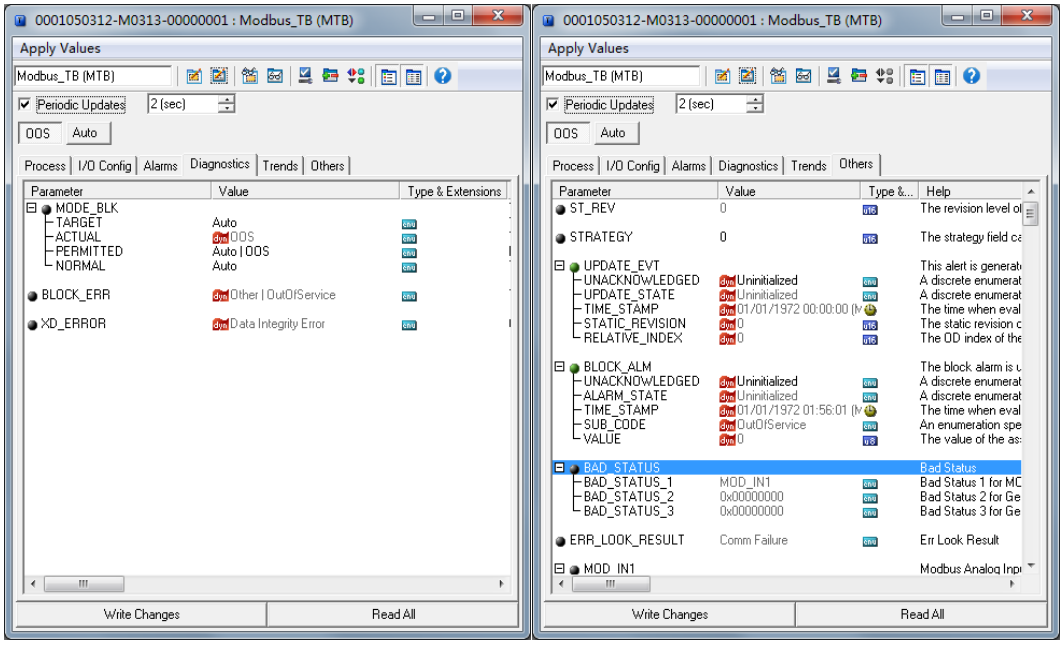


Figure 4.17 Transducer block OOS situation 5

Situation 6

After the device is powered on, the XD_ERROR parameter value of the function block is "Data Integrity Error", the ERR_LOOK_RESULT parameter value is "Illegal Data Address", and the BAD_STATUS parameter has a value such as "MOD_IN1", indicates that the device has read data addresses during normal communication. Please check the device's data register configuration.

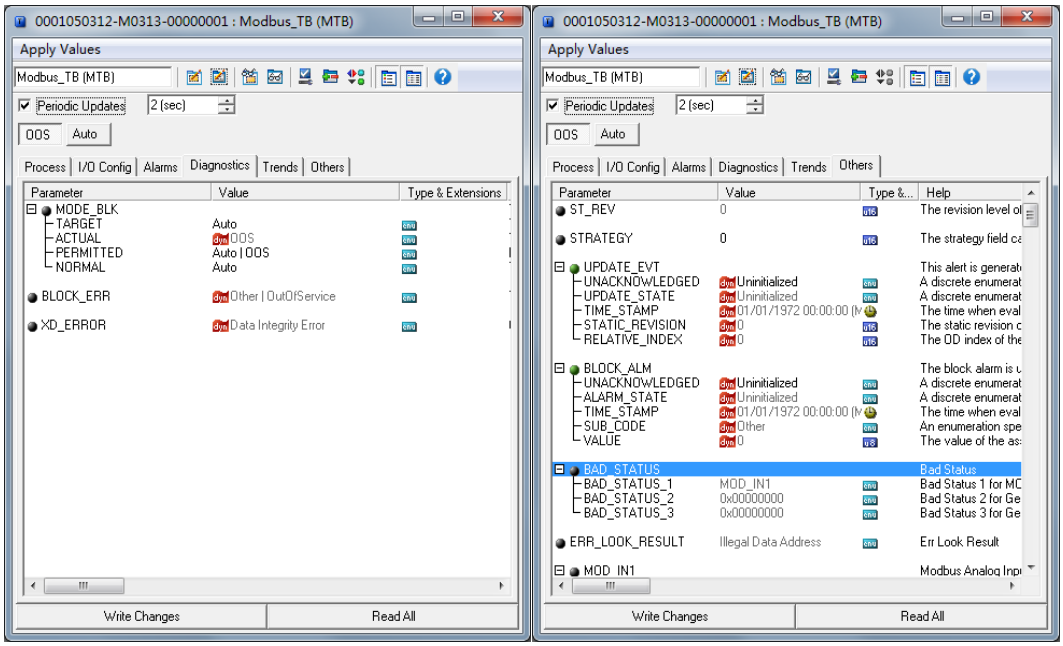


Figure 4.18 Transducer block OOS situation 6

Situation 7

After the device is powered on, the XD_ERROR parameter value of the function block is "Data Integrity Error", the ERR_LOOK_RESULT parameter value is "Illegal Function", and the BAD_STATUS parameter has a value such as "MOD_IN1", indicates that the device has a communication function code matching problem during normal communication. Please check the data function code configuration of the device.

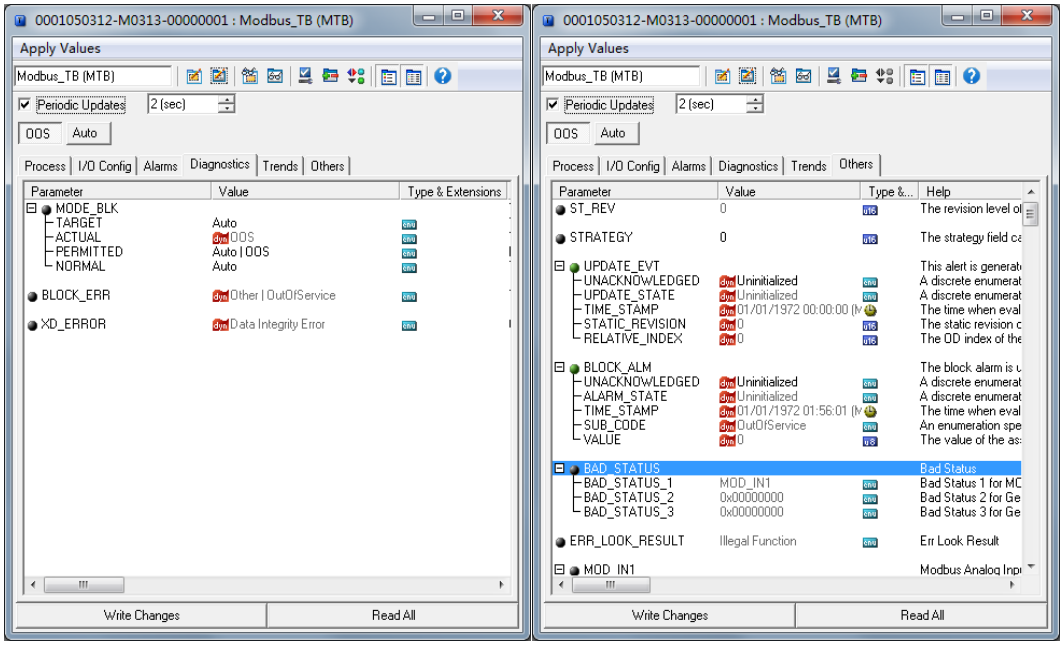


Figure 4.19 Transducer block OOS situation 7

Situation 8

After the device is powered on, the XD_ERROR parameter value of the function block is "Data Integrity Error", the ERR_LOOK_RESULT parameter value is "Unknown Exception Code", and the BAD_STATUS parameter has a value such as "MOD_IN1", indicates that the device has encountered problems during normal communication and the error code returned cannot be parsed. Please check the specific communication configuration of Modbus from the device.

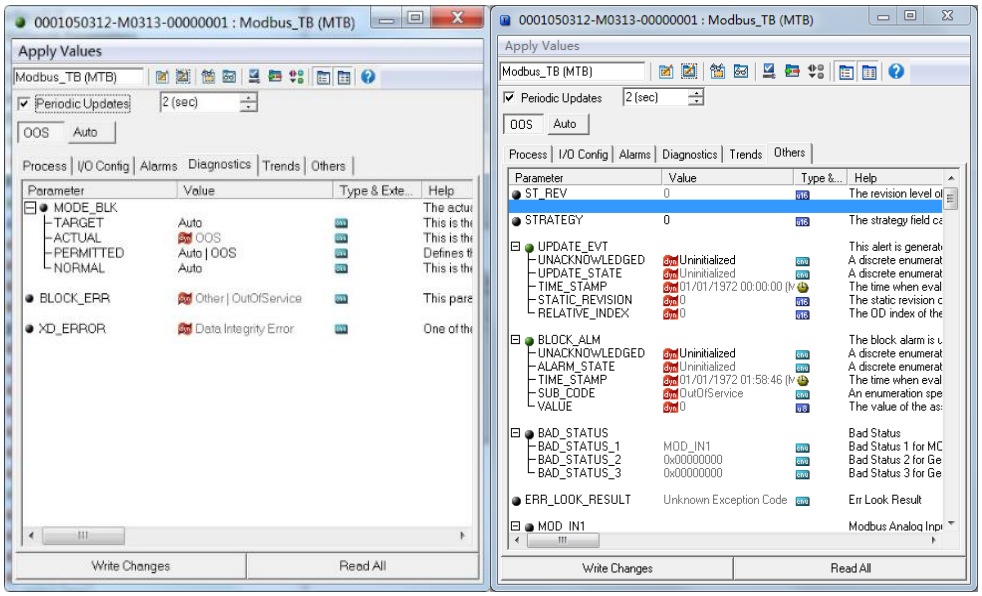


Figure 4.20 Transducer block OOS situation 8

Situation 9

After the device is powered on, the XD_ERROR parameter value of the function block is "Data Integrity Error", the ERR_LOOK_RESULT parameter value is "Slave Device Failure", and the BAD_STATUS parameter value is "MOD_IN1", indicates that the device has had a problem during normal communication, and the error code returned is "Slave Device Failure". Please check the status of Modbus from the device.

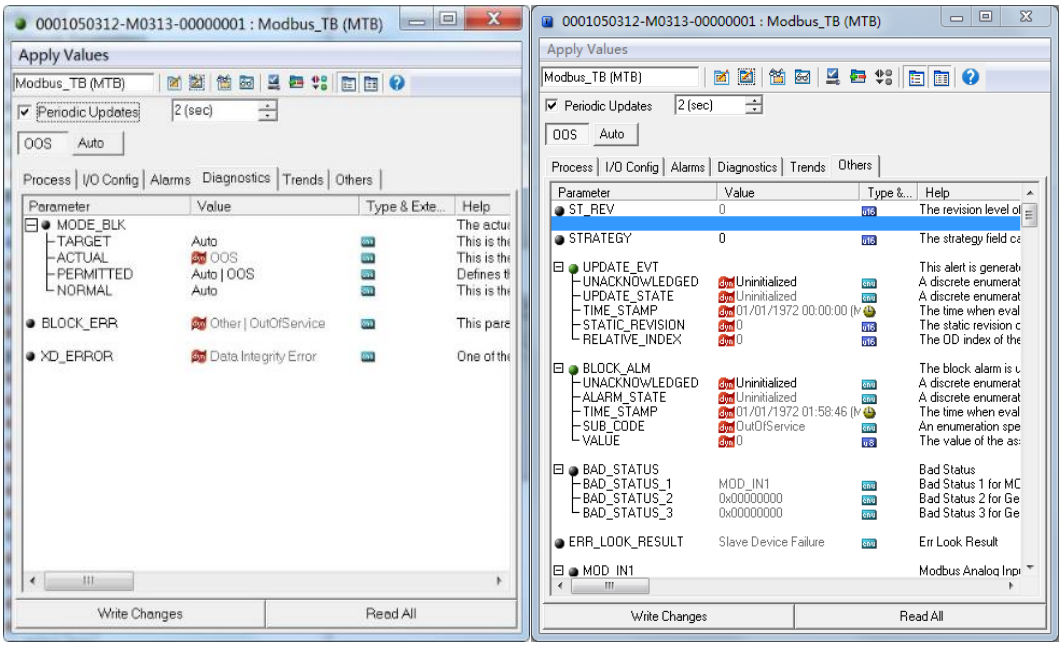


Figure 4.21 Transducer block OOS situation 9

5 Maintenance

- Basic Maintenance

LED Indicator	Colour	Normal state	Abnormal state	Abnormal Reason	Correction Method
FF Communication	Green	Flash	Off	No FF communication	Check the FF host device and FF interface device
				Power supply failure	Check the power supply and connection
				Internal failure	Contact Technical Support
			Bright	No FF communication	Check the FF host device and FF interface device
				Internal failure	Contact Technical Support

- Routine maintenance is limited to equipment cleaning.
- Fault repair: If any fault is found, please return to the factory for repair.

6 Technical Specifications

6.1 Basic Parameters

Measurement object	Modbus RTU slave device
FF Bus Power Supply	9-32VDC
Static Current	≤14mA
Bus protocol	2-wire, FF protocol
Isolation Voltage	Modbus and FF/PA buses are not isolated
Temperature range	-40℃～85℃
Humidity range	5-95% RH
Start-up time	≤5 seconds

6.2 Performance Indicators

EMC	Complies with the immunity requirements for industrial sites in GB/T 18268.1-2010 "Electromagnetic compatibility requirements for electrical equipment for measurement, control and laboratory use - Part 1: General requirements" The FF port test method adopts GB/T 18268.23-2010 "Electromagnetic compatibility requirements for electrical equipment for measurement, control and laboratory use - Part 23: Special requirements for test configuration, working conditions and performance judgment of transmitters with integrated or remote signal conditioning" according to"

6.3 Physical Characteristics

Weight	16 g
Structural materials	Coating: Polyester epoxy resin.

6.4 Default Communication Parameters

Slave Address	1
Baud rate	9600
Data bits	8
Stop Bit	1
Check	EVEN
CRC Check	Low bytes first

6.5 Support Modbus Function Code

1	Read coil status
2	Read discrete input status
3	Read holding register value
4	Read input register value
5	Writing coil
6	Write a single register value
15	Write multiple coils
16	Write multiple register values

Appendix 1 Modbus Transducer Block Parameter Summary Table

Index	Parameter name	Data type	Valid range	Default value	Storage	Function description
1	ST_REV	Unsigned16		0	S/RO	Static version
2	TAG_DESC	OctString(32)		Spaces	S	Bit number
3	STRATEGY	Unsigned16		0	S	Strategy
4	ALERT_KEY	Unsigned8	1 to 255	0	S	Call the police
5	MODE_BLK	DS-69		O/S	S	Model
6	BLOCK_ERR	Bitstring(2)			D/RO	Mistake
7	UPDATA_EVT	DS-73			D	Static data update event
8	BLOCK_ALM	DS-72			D	Function block alarm
9	TRANSDUCER_TYPE	Unsigned16		65535	N/RO	Transducer block type
10	XD_ERROR	Unsigned8		0	D/RO	Transducer block error description
11	SENSOR_TYPE	Unsigned16		65535	D/RO	Sensor type
12	BAD_STATUS	DS-258			D/RO	It consists of 3 32-bit status sub-parameters. If a certain bit is set to 1, it means that the corresponding command has not received a response. You can observe the parameters by configuring and refer to the error code for details.
13	ERR_LOOK_RESULT	Unsigned8	0-255		D/RO	32 Input and output command negative response exception code index address, for example, 0 means the negative response code of MOD_IN1
14	MOD_IN1	DS-65			D/RO	Analog input 1
15	MOD_IN2	DS-65			D/RO	Analog input 2
16	MOD_IN3	DS-65			D/RO	Analog input 3
17	MOD_IN4	DS-65			D/RO	Analog input 4
18	MOD_IN5	DS-65			D/RO	Analog input 5
19	MOD_IN6	DS-65			D/RO	Analog input 6
20	MOD_IN7	DS-65			D/RO	Analog input 7
21	MOD_IN8	DS-65			D/RO	Analog input 8
22	MOD_OUT1	DS-65			D/RO	Analog output 1
23	MOD_ OUT 2	DS-65			D/RO	Analog output 2
24	MOD_ OUT 3	DS-65			D/RO	Analog output 3
25	MOD_ OUT 4	DS-65			D/RO	Analog output 4
26	MOD_ OUT 5	DS-65			D/RO	Analog output 5
27	MOD_ OUT 6	DS-65			D/RO	Analog output 6
28	MOD_ OUT 7	DS-65			D/RO	Analog output 7
29	MOD_ OUT 8	DS-65			D/RO	Analog output 8
30	MOD_IN_D1	DS-66			D/RO	Discrete input 1
31	MOD_IN_D 2	DS-66			D/RO	Discrete input 2
32	MOD_IN_D 3	DS-66			D/RO	Discrete input 3
33	MOD_IN_D 4	DS-66			D/RO	Discrete input 4
34	MOD_IN_D 5	DS-66			D/RO	Discrete input 5
35	MOD_IN_D 6	DS-66			D/RO	Discrete input 6
36	MOD_IN_D 7	DS-66			D/RO	Discrete input 7
37	MOD_IN_D 8	DS-66			D/RO	Discrete input 8
38	MOD_OUT_D1	DS-66			D/RO	Discrete output 1
39	MOD_OUT_D 2	DS-66			D/RO	Discrete output 1
40	MOD_OUT_D 3	DS-66			D/RO	Discrete output 1
41	MOD_OUT_D 4	DS-66			D/RO	Discrete output 1
42	MOD_OUT_D 5	DS-66			D/RO	Discrete output 1
43	MOD_OUT_D 6	DS-66			D/RO	Discrete output 1
44	MOD_OUT_D 7	DS-66			D/RO	Discrete output 1
45	MOD_OUT_D 8	DS-66			D/RO	Discrete output 1
46	GENERIC_FLOAT_PARAM_1	DS-256			S	Generic Floating Point Parameters 1
47	GENERIC_FLOAT_PARAM_2	DS-65			S	Generic Floating Point Parameters 2
48	GENERIC_FLOAT_PARAM_3	DS-256			S	Generic Floating Point Parameters 3
49	GENERIC_FLOAT_PARAM_4	DS-65			S	Generic Floating Point Parameters 4
50	GENERIC_FLOAT_PARAM_5	DS-256			S	Generic Floating Point Parameters 5
51	GENERIC_FLOAT_PARAM_6	DS-65			S	General purpose floating point parameters 6
52	GENERIC_FLOAT_PARAM_7	DS-256			S	Generic Floating Point Parameters 7
53	GENERIC_FLOAT_PARAM_8	DS-65			S	Generic Floating Point Parameters 8
54	GENERIC_FLOAT_PARAM_9	DS-256			S	Generic Floating Point Parameters 9
55	GENERIC_FLOAT_PARAM_10	DS-66			S	Generic Floating Point Parameters 10
56	GENERIC_USIGN32_PARAM_1	Unsigned32			S	Generic 32-bit unsigned parameter 1
57	GENERIC_USIGN32_PARAM_2	Unsigned32			S	Generic 32-bit unsigned parameter 2

58	GENERIC_USIGN32_PARAM_3	Unsigned32			S	Generic 32-bit unsigned parameter 3
59	GENERIC_USIGN32_PARAM_4	Unsigned32			S	Generic 32-bit unsigned parameter 4
60	GENERIC_USIGN32_PARAM_5	Unsigned32			S	Generic 32-bit unsigned parameter 5
61	GENERIC_USIGN32_PARAM_6	Unsigned32			S	Generic 32-bit unsigned parameter 6
62	GENERIC_USIGN32_PARAM_7	Unsigned32			S	Generic 32-bit unsigned parameter 7
63	GENERIC_USIGN32_PARAM_8	Unsigned32			S	Generic 32-bit unsigned parameter 8
64	GENERIC_USIGN32_PARAM_9	Unsigned32			S	Generic 32-bit unsigned parameter 9
65	GENERIC_USIGN32_PARAM_10	Unsigned32			S	Generic 32-bit unsigned parameter 10
66	GENERIC_USIGN16_PARAM_1	Unsigned16			S	Generic 16-bit unsigned parameter 1
67	GENERIC_USIGN16_PARAM_2	Unsigned16			S	Generic 16-bit unsigned parameter 2
68	GENERIC_USIGN16_PARAM_3	Unsigned16			S	Generic 16-bit unsigned parameter 3
69	GENERIC_USIGN16_PARAM_4	Unsigned16			S	Generic 16-bit unsigned parameter 4
70	GENERIC_USIGN16_PARAM_5	Unsigned16			S	Generic 16-bit unsigned parameter 5
71	GENERIC_USIGN16_PARAM_6	Unsigned16			S	Generic 16-bit unsigned parameter 6
72	GENERIC_USIGN16_PARAM_7	Unsigned16			S	Generic 16-bit unsigned parameter 7
73	GENERIC_USIGN16_PARAM_8	Unsigned16			S	Generic 16-bit unsigned parameter 8
74	GENERIC_USIGN16_PARAM_9	Unsigned16			S	Generic 16-bit unsigned parameter 9
75	GENERIC_USIGN16_PARAM_10	Unsigned16			S	Generic 16-bit unsigned parameter 10
76	GENERIC_USIGN8_PARAM_1	Unsigned8			S	Generic 8-bit unsigned parameter 1
77	GENERIC_USIGN8_PARAM_2	Unsigned8			S	Generic 8-bit unsigned parameter 2
78	GENERIC_USIGN8_PARAM_3	Unsigned8			S	Generic 8-bit unsigned parameter 3
79	GENERIC_USIGN8_PARAM_4	Unsigned8			S	Generic 8-bit unsigned parameter 4
80	GENERIC_USIGN8_PARAM_5	Unsigned8			S	Generic 8-bit unsigned parameter 5
81	GENERIC_USIGN8_PARAM_6	Unsigned8			S	Generic 8-bit unsigned parameter 6
82	GENERIC_USIGN8_PARAM_7	Unsigned8			S	Generic 8-bit unsigned parameter 7
83	GENERIC_USIGN8_PARAM_8	Unsigned8			S	Generic 8-bit unsigned parameter 8
84	GENERIC_USIGN8_PARAM_9	Unsigned8			S	Generic 8-bit unsigned parameter 9
85	GENERIC_USIGN8_PARAM_10	Unsigned8			S	Generic 8-bit unsigned parameter 10
86	GENERIC_STRINGV_PARAM_1	Octet String(32)			S	Generic 32-bit string parameter 1
87	GENERIC_STRINGV_PARAM_2	Octet String(32)			S	Generic 32-bit string parameter 2

Appendix 2 Common Parameter Table

NO.	Parameter name	Description
1	Data configuration flag	Data configuration flag 0xFEDCCDEF: Data is valid 0x00000000: Invalid data
2	Equipment serial number	Equipment serial number SN (6 bytes)
3	Number of analog input data	Number of analog input data (0~8)
4	Number of analog output data	Number of analog output data (0~8)
5	Number of digital input data	Number of digital input data (0~8)
6	Digital output data	Number of digital output data (0~8)
7	Number of floating point data	Number of floating point data (0~10)
8	Number of USIGN32 data	Number of USIGN32 data (0~10)
9	Number of USIGN16 data	Number of USIGN16 data (0~10)
10	Number of USIGN8 data	Number of USIGN8 data (0~10)
11	Number of Octet String data	Number of Octet String data (0~2)
12	Baud rate	0：2400 1：4800 2：9600 3：14400 4：19200
13	Data bits	0：8 1：7
14	Check bit	0：None 1：Even 2：Odd
15	Physical standards	Interface type 0：TTL 1：RS232 2：RS485
16	Stop bit	0：One Stop Bit 1：Two Stop Bits
17	Address	Slave address (1~255), which is the slave address under normal working mode
18	CRC	CRC check sequence 0：Normal 1：Swapped
19	overtime	Timeout (300~1000, unit: ms)
20	Number of retransmissions	Number of retransmissions (1~10)
21	View error parameters	View error parameters (0~73, respectively representing 74 channel data of analog input, analog output, digital input, digital output, floating point data, USIGN32 data, USIGN16 data, USIGN8 data and Octet String data)

Microcyber Corporation
www.microcybers.com
Add：17-8 Wensu Street, Hunnan New District, Shenyang, China 110179
Tel：0086-24-31217278 / 31217280
Fax：0086-24-31217293
Email：sales@microcyber.cn