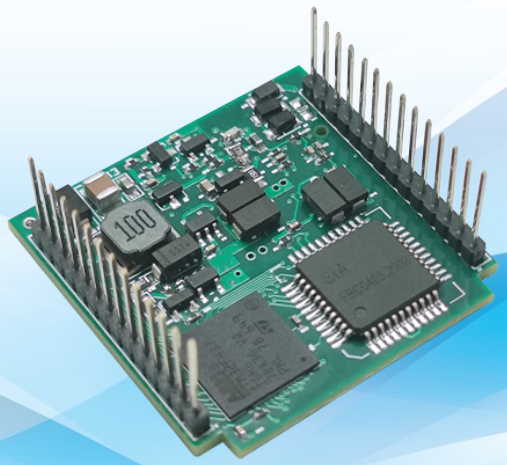
MICROCYBER

MC0307

Modbus to PA Built-in Core Module

User Manual



Warning

1. Please don’t take off/install gateway at random.
2. Please check if the power of the module meets the requirements of different types of power in the User Manual.

Version: V1.2

Disclaimer

The contents of this manual have been checked to confirm the consistency of the described hardware and software. Because the error can not be completely excluded, there is no guarantee of absolute consistency. However, we will regularly check the data in this manual and make necessary corrections in subsequent versions. 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, national science and technology programs for intelligent manufacturing equipment development and it is the national network control system engineering research center construction support unit.

Microcyber Corporation successfully developed the first internationally certified fieldbus protocol master stack, the first nationally certified fieldbus instrument, and the first German TüV certified safety instrument in China. It co-chaired with other units to formulate the first domestic industrial Ethernet protocol standard EPA, the first industrial wireless communication protocol standard WIA-PA, and become the IEC international standard. Microcyber Corporation’s products and technology have won two national second prize for scientific and technological progress, one national scientific and technological invention award, one first prize for scientific and technological progress of the Chinese Academy of Sciences, and one first prize for scientific and technological progress of Liaoning Province. The United States Emerson, Britain Rotork, Britain and other top enterprises have adopted key technologies or components in their products and successfully completed more than 200 large-scale automation projects.

Microcyber is the FF member, the HART member and the Profibus National Organization (PNO) member.

Microcyber passes the Authentication of ISO 9001:2008 Quality System and automotive industry ISO/TS16949 quality system certification. We have laid a solid foundation for the company's entrepreneurship and sustainable development with excellent R & D team, rich experience in automation engineering design and implementation, industry leading products, huge market network and excellent corporate culture.

Carrying employee ideal, creating customer value and promoting enterprise development.

Content

Chapter 1 Overview

MC0307 Modbus to PA built-in core board module is an built-in conversion module of Modbus-RTU protocol and PA protocol developed by Microcyber Corporation. It is one of the Microcyber M series built-in core board modules. This series of built-in core board modules have the same size, the same interface, easy to upgrade, simple configuration, etc. It is the ideal choice for users to quickly develop the field bus equipment. MC0307 Modbus to PA built-in core board module, as the Modbus host, communicates with the device with Modbus-RTU communication function through the TTL interface, and can convert the data in the device to PA device variable output. MC0307 Modbus to PA built-in core board module, as shown in Figure 1.1 below.

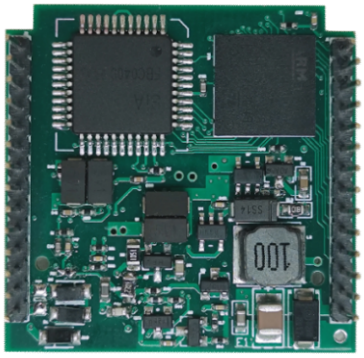


Figure 1.1 MC0307 Modbus to PA built-in core board module

1.1 Module Features

1.1.1 Same Size

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

1.1.2 Same Interface

Microcyber M series embedded core board modules all adopt 2.0 spacing double row 14 pin connectors, which are functionally compatible.

1.1.3 Easy to upgrade

Replace different embedded core board modules of Microcyber M series, and immediately implement devices with different protocols.

1.1.4 Simple configuration

Use the Microcyber special configuration tool for configuration. Easy to operate and easy to use.

1.2 Product Development Process

First step：Hardware design

According to the dimensions of the module and the definition of interface pin, the hardware schematic and PCB of original user products are redesigned. If M series compatibility is considered, refer to the interface pin definition of all M series modules for hardware design.

1.3 Outline Size

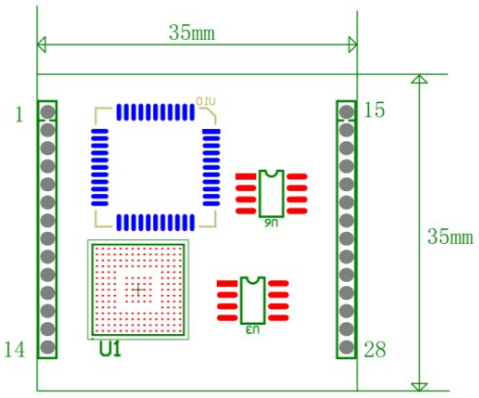


Figure 1.2 Built-in core board module equipment dimensions（unit：mm）

1.4 Module Structure

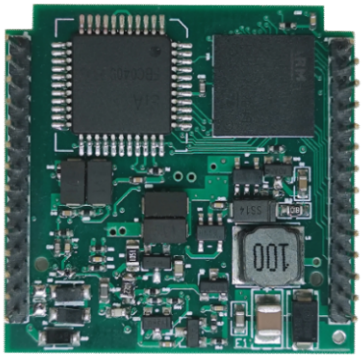


Figure 1.3 Built-in core board module structure

Chapter 2 Installation

2.1 Module External Interface

The terminal distribution and meaning of the MC0307 Modbus to PA built-in core board module are shown in Figure 2.1 below:

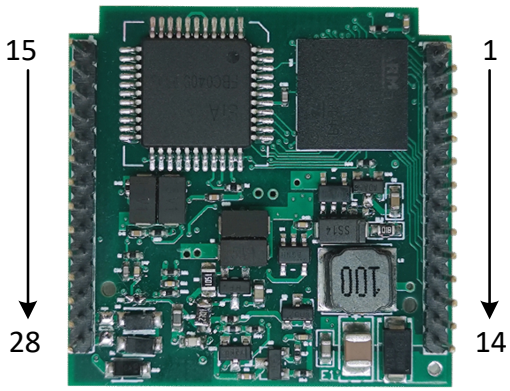


Figure 2.1 MC0307 Modbus to PA built-in core board module communication interface definition

2.2 Introduction of the Module Interface Function

Pin	Name	Description
1	SWDIO	Data input and output(download interface)
2	SWCLK	Clock signal(download interface)
3	NRST	Single-chip microcomputer reset signal (download interface)
4	GND	reference place
5	IO	Reserve GPIO
6	RXD	UART1 input TTL level
7	TXD	UART1 output TTL level
8	IO	Reserve GPIO
9	SPI1_NSS	Reserve SPI interface/GPIO
10	SPI1_MISO	Reserve SPI interface/GPIO
11	SPI1_MOSI	Reserve SPI interface/GPIO
12	SPI1_CLK	Reserve SPI interface/GPIO
13	3.3V_OUT	Power output
14	6.2V_OUT	Power output
15	IO	Reserve GPIO

16	IO	Reserve GPIO
17	WP	Write protection enablement
18	RST	Reset the DIP switch to restore the data of the communication module to the factory state. First power off the PA communication module, turn the DIP switch to the ON position, power on the module, and restore the PA communication module to the factory state, then turn the DIP switch back to the OFF position to ensure that the next power-on will not malfunction.
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	Reserve GPIO
22	SCL	Reserve I2C pins/GPIO
23	SDA	Reserve I2C pins/GPIO
24	IO	Modbus communication status indicator, active low
25	IO	PA communication status indicator light, active low
26	GND	reference place
27	BUS+	Bus power supply is positive
28	BUS-	Bus power supply is negative

Chapter 3 Working Principle

MC0307 Modbus to PA embedded module is only support one-to-one Modbus and Profibus PA communication protocol conversion module. As a PA device, you can communicate with the Modbus device. Through the configuration, the interaction between Modbus device data and PA device data can be realized.

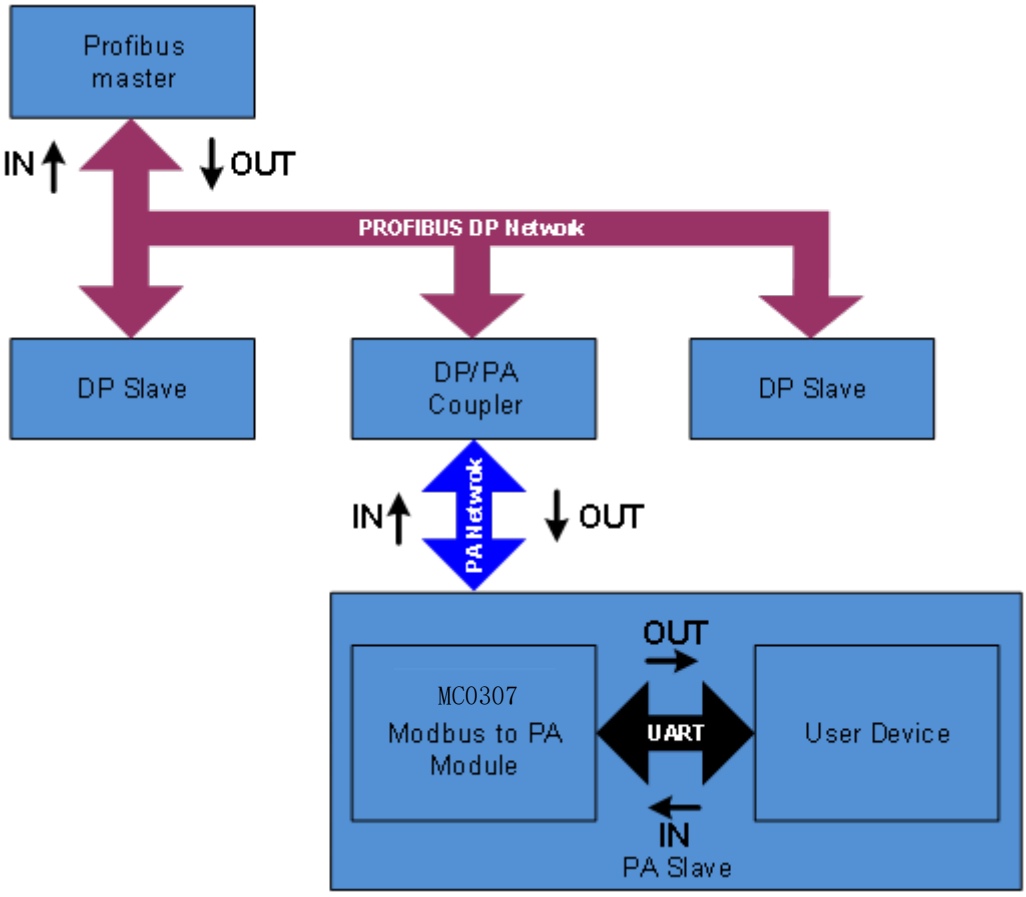


Figure 3.1 System connection diagram

MC0307 Modbus to PA embedded module contains 1 physical block, 1 transformation block, 16 functional blocks, and only supports 1 Modbus slave station.

Among them, the transformation block mainly includes 6 analog input, 2 analog output, 4 discrete input and 4 discrete output, a total of 16 circular parameters, 10 floating point, 1032-bit integer, 1016-bit integer, 108-bit integer and 232-byte strings, a total of 42 non-circular parameters.

The main function of a transform block is to interact with Modbus devices.

3.1 Working Mode

MC0307 can switch between two working modes through the 8th bit M of the DIP switch on the bottom plate: configuration mode and normal working mode. ON is configuration mode, OFF is normal working mode.

3.1.1 Configuration Mode

When MC0307 works in configuration mode, MC0307 as a Modbus slave station, and the Modbus general configuration tool acts as a Modbus master station. Through the Modbus general configuration tool, in addition to

configuring basic information such as manufacturer ID, device ID, and device address source, the main function is to configure the information of the 16 cyclic parameters and 42 acyclic parameters mentioned above, such as which Modbus parameters are used Function code to read and write, what is the register address and so on. After configuration, the information will be downloaded to MC0307 for storage.

3.1.2 Normal Working Mode

When MC0307 works in normal working mode, MC0307 acts as the Modbus master station, and the user equipment as the Modbus slave station. The MC0307 module interacts with the user equipment by sending Modbus commands through the configured working mechanism.

The data collected by the Modbus device is mapped to the parameters of the transformation block of MC0307 by reading and writing the Modbus register, and then provides data support for the Profibus system through the channel access function of the transformation block to the AI, AO, DI and DO function blocks.

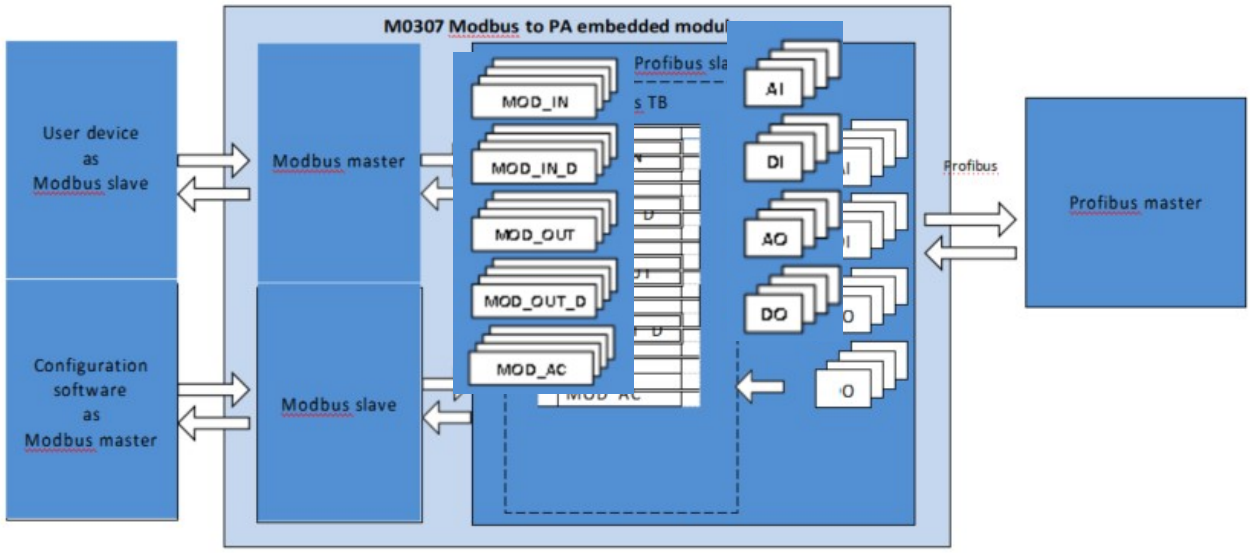
Figure 3.2 Working principle block diagram

The working mechanism of the normal working mode is divided into the following three stages: initialization stage, acyclic stage, and cyclic stage.

- Initialization stage

The main function of the initi[illegible] phase is to test whether the Modbus communication is normal. In this stage,

a



Modbus command will be selected according to the actual configuration, and if a correct response is obtained, it will enter the next stage.

The priority of the selection command is to read the serial port address, read the analog input, and read the discrete input.

That is, if the function of setting the address of the serial port is configured, the command to read the serial port address will be sent at this stage, and the command to read the analog input or discrete input will not be sent; if the function of setting the address of the serial port is not configured, the The number to select the sending command, the priority of analog input is higher than that of discrete input, that is, when the number of analog

input is greater than 0, send and read analog input instead of discrete input. The module also stipulates that at least one of the number of analog input and the number of discrete input is non-zero. If the continuous function is selected, send a continuous read command, otherwise send the read first analog input or discrete input command.

For detailed rules, see the table below.

Table 3.1 Initialization phase command selection rules

Serial port address	Analog input quantity	Analog input register address consecutive	Discrete input quantity	Discrete input Register address consecutive	Send command
√	*	*	*	*	Read serial port address
×	6≥AI > 0	×	*	*	Read the analog quantity input 1
×	6≥AI > 0	√	*	*	Read all analog inputs
×	0	*	4≥DI > 0	×	Read discrete quantity input 1
×	0	*	4≥DI > 0	√	Read all discrete inputs
* **When no effect, can be any state.**					

- Non-cyclic stage

The non-cyclic stage mainly reads all the data once. The order of reading is analog input, discrete input, floating-point data, USIGN32 data, USIGN16 data, USIGN8 data, Octet String data.

Among them, analog input and discrete input can send a one-time read command according to the configuration. For other data, even if the continuous function is configured, each data will be read separately. The continuous function at this time is only for the convenience of user configuration.

That is, the continuous function is divided into two types. The continuous function of the cyclic parameter supports sending one-time read and write commands, while the continuous function of the non-cyclic parameter is only for the convenience of user configuration.

When all data has been read correctly, it enters the loop phase. Otherwise, the acyclic phase will be re-executed until all data has been read successfully.

- Cycle stage

There are two main functions in the cyclic phase, namely cyclic reading of cyclic input and output parameters and writing of non-cyclic parameters.

When the module enters this state, it will send read analog input, write analog output, read discrete input, and write discrete output commands cyclically in sequence. When the acyclic parameter changes, the write acyclic parameter command will be sent.

When there are more than 10 consecutive errors, jump back to the acyclic stage and read all the data again.

If the serial port address is configured, then at this stage, the read serial port address command will be sent in real time after the discrete output command is written.

Chapter 4 Module Configuration

4.1 Topology Structure

The PA device supports multiple network topology wiring methods, as shown in Figure 4.1. Figure 4.2 shows the bus connection of the PA equipment. Both ends of the bus need to be connected with terminal matching resistors to ensure the signal quality. The maximum length of the bus is 1900 meters, which can be extended to 10 kilometers using repeaters.

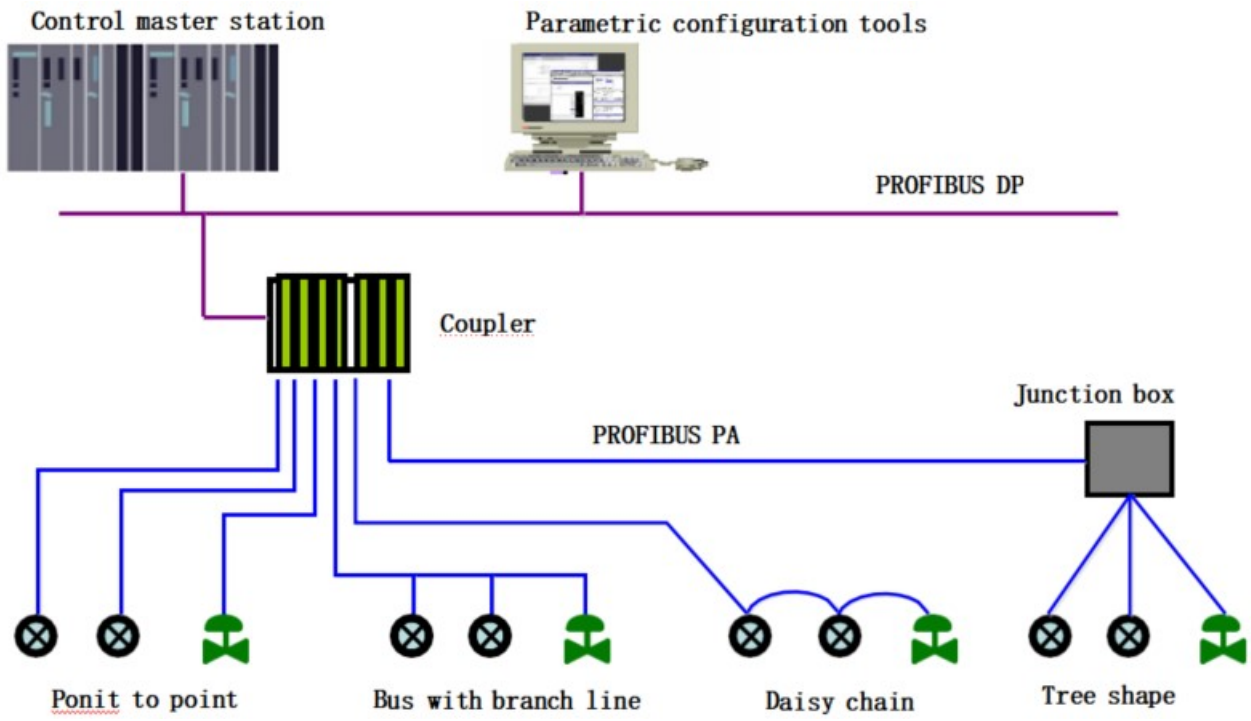


Figure 4.1 PA network topology

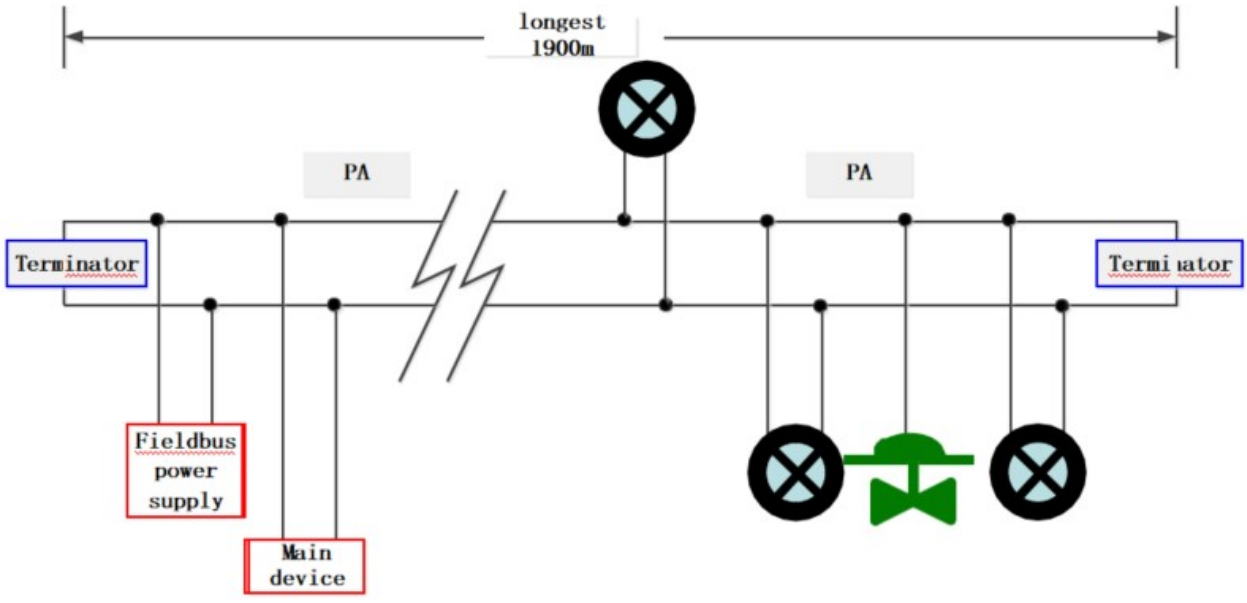


Figure 4.2 PA Bus connection

4.2 Function Block Description

MC0307 Modbus to PA embedded module, including 1 physical block, 6 AI function blocks, 2 AO function blocks, 4 DI function blocks, 4 DO function blocks and 1 User transformation block. Among them, each AI function block has 6 channels, the AO function block has 2 channels, the DI function block has 4 channels, and the DO function block has 4 channels, which point to 6 analog inputs and 2 analog inputs of the User transformation block respectively. Analog output, 4 discrete inputs and 4 discrete outputs.

Table 4.1 Description of the functional blocks

Function Block Name	Function Block Description
Physical Block	Physical Block (PB). It describes the hardware information and identification and diagnosis information specific to the device, including device bit number, software version, hardware version, installation date, etc.
User Transducer Block	Modbus parameters such as 6 analog inputs, 2 analog outputs, 4 discrete inputs and 4 discrete outputs can be read and written through the User conversion block.
Analog Input Block	Analog input function block (AI). Acquires analog process values from Modbus slave devices via internal channels, processes them, and provides the appropriate measured values to the master device via bus communication.
Analog Output Block	Analog Output Block (AO), which is used to pass the data output from the master device to the converter block through the internal channel for Modbus slave devices.
Discrete Input Block	Discrete Input Block (DI), which obtains input data from Modbus slave devices through internal channels and provides it to the master device through bus communication.
Discrete Output Block	Discrete output function block (DO), which passes the discrete output data set by the master device to the converter block through the internal channel, acting on Modbus slave device.

4.3 User Transform Block Parameters

Before configuring the module, let's take a look at the User transformation block. The following table describes the parameters of all User transformation blocks.

Table 4.2 Modbus conversion block parameter attribute table

Index	Parameter Name	Data Type	Valid Range	Default Value	Storage Mode	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	Policy
4	ALERT_KEY	Unsigned8	1 to 255	0	S	Alarm
5	TARGET_MODE	Unsigned8		AUTO	S	Target mode
6	MODE_BLK	DS-37			D	Current mode
7	ALARM_SUM	DS-42			D	Alarm summary
12	BAD_STATUS	Bitstring(4)			D/RO	16 input and output command communication status, a bit set to 1 means that the corresponding command is not responded to, see the description of this parameter in the following table
13	ERR_LOOK_RESULT	Unsigned8(32)	0-255	0xFC	D/RO	16 input-output command

						negative response exception code
14	MOD_IN1	101			D/RO	Analog input 1
15	MOD_IN2	101			D/RO	Analog input 2
16	MOD_IN3	101			D/RO	Analog input 3
17	MOD_IN4	101			D/RO	Analog input 4
22	MOD_IN5	101			D/RO	Analog input 5
23	MOD_IN6	101			D/RO	Analog input 6
24	MOD_OUT1	101			D/RO	Analog output 1
25	MOD_ OUT2	101			D/RO	Analog output 2
30	MOD_IN_D1	102			D/RO	Discrete quantity input 1
31	MOD_IN_D2	102			D/RO	Discrete quantity input 2
32	MOD_IN_D3	102			D/RO	Discrete quantity input 3
33	MOD_IN_D4	102			D/RO	Discrete quantity input 4
38	MOD_OUT_D1	102			D/RO	Discrete quantity output 1
39	MOD_OUT_D2	102			D/RO	Discrete quantity output 2
40	MOD_OUT_D3	102			D/RO	Discrete quantity output 3
41	MOD_OUT_D4	102			D/RO	Discrete quantity output 4
46	GENERIC_FLOAT_1	Float		0	S	Generic floating point variables 1
47	GENERIC_FLOAT_2	Float		0	S	Generic floating point variables 2
48	GENERIC_FLOAT_3	Float		0	S	Generic floating point variables 3
49	GENERIC_FLOAT_4	Float		0	S	Generic floating point variables 4
50	GENERIC_FLOAT_5	Float		0	S	Generic floating point variables 5
51	GENERIC_FLOAT_6	Float		0	S	Generic floating point variables 6
52	GENERIC_FLOAT_7	Float		0	S	Generic floating point variables 7
53	GENERIC_FLOAT_8	Float		0	S	Generic floating point variables 8
54	GENERIC_FLOAT_9	Float		0	S	Generic floating point variables 9
55	GENERIC_FLOAT_10	Float		0	S	Generic floating point variables 10
56	GENERIC_USIGN32_1	Unsigned32		0	S	Generic 32-bit unsigned integer variable 1
57	GENERIC_USIGN32_2	Unsigned32		0	S	Generic 32-bit unsigned integer variable 2
58	GENERIC_USIGN32_3	Unsigned32		0	S	Generic 32-bit unsigned integer variable 3
59	GENERIC_USIGN32_4	Unsigned32		0	S	Generic 32-bit unsigned integer variable 4
60	GENERIC_USIGN32_5	Unsigned32		0	S	Generic 32-bit unsigned integer variable 5
61	GENERIC_USIGN32_6	Unsigned32		0	S	Generic 32-bit unsigned integer variable 6
62	GENERIC_USIGN32_7	Unsigned32		0	S	Generic 32-bit unsigned integer variable 7
63	GENERIC_USIGN32_8	Unsigned32		0	S	Generic 32-bit unsigned integer variable 8
64	GENERIC_USIGN32_9	Unsigned32		0	S	Generic 32-bit unsigned integer variable 9
65	GENERIC_USIGN32_10	Unsigned32		0	S	Generic 32-bit unsigned integer variable 10
66	GENERIC_USIGN16_1	Unsigned16		0	S	Generic 16-bit unsigned integer variable 1
67	GENERIC_USIGN16_2	Unsigned16		0	S	Generic 16-bit unsigned integer variable 2

68	GENERIC_USIGN16_3	Unsigned16		0	S	Generic 16-bit unsigned integer variable 3
69	GENERIC_USIGN16_4	Unsigned16		0	S	Generic 16-bit unsigned integer variable 4
70	GENERIC_USIGN16_5	Unsigned16		0	S	Generic 16-bit unsigned integer variable 5
71	GENERIC_USIGN16_6	Unsigned16		0	S	Generic 16-bit unsigned integer variable 6
72	GENERIC_USIGN16_7	Unsigned16		0	S	Generic 16-bit unsigned integer variable 7
73	GENERIC_USIGN16_8	Unsigned16		0	S	Generic 16-bit unsigned integer variable 8
74	GENERIC_USIGN16_9	Unsigned16		0	S	Generic 16-bit unsigned integer variable 9
75	GENERIC_USIGN16_10	Unsigned16		0	S	Generic 16-bit unsigned integer variable 10
76	GENERIC_USIGN8_1	Unsigned8		0	S	Generic 8-bit unsigned integer variable 1
77	GENERIC_USIGN8_2	Unsigned8		0	S	Generic 8-bit unsigned integer variable 2
78	GENERIC_USIGN8_3	Unsigned8		0	S	Generic 8-bit unsigned integer variable 3
79	GENERIC_USIGN8_4	Unsigned8		0	S	Generic 8-bit unsigned integer variable 4
80	GENERIC_USIGN8_5	Unsigned8		0	S	Generic 8-bit unsigned integer variable 5
81	GENERIC_USIGN8_6	Unsigned8		0	S	Generic 8-bit unsigned integer variable 6
82	GENERIC_USIGN8_7	Unsigned8		0	S	Generic 8-bit unsigned integer variable 7
83	GENERIC_USIGN8_8	Unsigned8		0	S	Generic 8-bit unsigned integer variable 8
84	GENERIC_USIGN8_9	Unsigned8		0	S	Generic 8-bit unsigned integer variable 9
85	GENERIC_USIGN8_10	Unsigned8		0	S	Generic 8-bit unsigned integer variable 10
86	GENERIC_OCTET_1	OctString(32)			S	Generic 32-byte String Variable 1
87	GENERIC_OCTET_2	OctString(32)			S	Generic 32-byte string variable 2
88	FLOAT_ERR_INFO	Unsigned8(10)	0-255	0xFC	S	Generic floating point negative response exception code
89	USIGN32_ERR_INFO	Unsigned8(10)	0-255	0xFC	S	Generic 32-bit unsigned integer negative response exception code
90	USIGN16_ERR_INFO	Unsigned8(10)	0-255	0xFC	S	Generic 16-bit unsigned integer negative response exception code
91	USIGN8_ERR_INFO	Unsigned8(10)	0-255	0xFC	S	Generic 8-bit unsigned integer negative response exception code
92	OCTET _ERR_INFO	Unsigned8(10)	0-255	0xFC	S	Generic 32-byte string negative response exception code

Among them, indexes 1-7 are standard parameters, which will not be described too much in this manual. The remaining parameters are described in detail below.

4.3.1 BAD_STATUS Parameter Description

The BAD_STATUS parameter is used to describe the communication status of the cyclic input and output parameters. If the communication fails, the corresponding bit is set to 1, otherwise it is 0. This parameter can be viewed in the Device->User Configuration->User Error Lookup menu in DD.

Table 4.3 BAD_STATUS parameter description table

Bit	Parameters	Bit	Parameters
0	MOD_IN1	16	MOD_IN_D1
1	MOD_IN2	17	MOD_IN_D2
2	MOD_IN3	18	MOD_IN_D3
3	MOD_IN4	19	MOD_IN_D4
4	MOD_IN5	18	——
5	MOD_IN6	18	——
6	——	22	——
7	——	23	——
8	MOD_OUT1	24	MOD_OUT_D1
9	MOD_ OUT2	25	MOD_ OUT_D2
10	——	26	MOD_ OUT_D3
11	——	27	MOD_ OUT_D4
12	——	28	——
13	——	29	——
14	——	30	——
15	——	31	——

4.3.2 Description of Negative Response Detection Parameters

The negative response detection parameter provides the negative response data query function, and the user can query the response value of each data. Negative response detection parameters include ERR_LOOK_RESULT parameter, FLOAT_ERR_INFO parameter, USIGN32_ERR_INFO parameter, USIGN16_ERR_INFO parameter, USIGN8_ERR_INFO parameter, OCTET _ERR_INFO parameter. View them in the User Error Lookup, User Generic Float, User Generic Usign32, User Generic Usign16, User Generic Usign8, and User Generic Octet menus in the Device->User Configuration menu in DD.

Table 4.4 ERR_LOOK_RESULT parameter description table

Value	Parameter Description
0x00	OK
0x01	Illegal Function
0x02	Illegal Data Address
0x03	Illegal Data Value
0x04	Slave Device Failure
0x05~0xFF	Unknown Exception Code
0xFC	No Communication
0xFD	Data Type Mismatch
0XFE	Function Code Mismatch
0xFF	Communication Failure

4.3.3 User Transformation Block Cycle Input and Output Parameters

User transformation block provides 6 analog input, 2 analog output, 4 discrete input and 4 discrete output parameters. In the case of not modifying the channel of the function block, the function of each input and output parameter is as follows:

Table 4.5 Cycle input and output parameter description table

Index	Parameter Name	Data Type	Description
14	MOD_IN1	101	Analog input, which transmits the value and status from the Modbus slave to AI function block 1
15	MOD_IN2	101	Analog input, transferring the value and status from the Modbus slave to AI function block 2
16	MOD_IN3	101	Analog input, transferring the value and status from the Modbus slave to AI function block 3
17	MOD_IN4	101	Analog input, transferring the value and status from the Modbus slave to AI function block 4
22	MOD_IN5	101	Analog input, which transmits the value and status from the Modbus slave to AI function block 5
23	MOD_IN6	101	Analog input, transferring the value and status from the Modbus slave to AI function block 6
24	MOD_OUT1	101	Analog output, transferring the set value and status from AO function block 1 to Modbus slave
25	MOD_OUT2	101	Analog output, transferring the set value and status from AO function block 2 to Modbus slave
30	MOD_IN_D1	102	Discrete input, transferring the value and status from Modbus slave to DI function block 1
31	MOD_IN_D2	102	Discrete input, transferring the value and status from Modbus slave to DI function block 2
32	MOD_IN_D3	102	Discrete input, transferring the value and status from Modbus slave to DI function block 3
33	MOD_IN_D4	102	Discrete input, transferring the value and status from Modbus slave to DI function block 4
38	MOD_OUT_D1	102	Discrete output, transferring the set value and status from DO function block 1 to Modbus slave
39	MOD_OUT_D2	102	Discrete output, transferring the set value and status from DO function block 2 to Modbus slave
40	MOD_OUT_D3	102	Discrete output, transferring the set value and status from DO function block 3 to Modbus slave
41	MOD_OUT_D4	102	Discrete output, transferring the set value and status from DO function block 4 to Modbus slave

Table 4.6 101 & 102 data type table

Data Type	Data Member	Member Data Type	Description
101	VALUE	Float	Floating point value.
	STATUS	Unsigned8	Contains both mass and state components.
102	VALUE	Unsigned8	Discrete values.
	STATUS	Unsigned8	Contains both mass and state components.

These parameters can be viewed in the Device->User Configuration menu in DD.

4.3.4 User Transform Block Acyclic Parameters

In addition to the cyclic input and output parameters, the User transform block also provides five acyclic parameters, as shown in the table below. These parameters can be used to store some auxiliary parameters, which

can be used flexibly according to your needs, such as upper and lower limits of a certain cyclic parameter, unit code, etc. They are read once at each power-up and can be written at any time afterwards. These parameters can be read and written in the Device -> User Configuration menu in DD.

Table 4.7 Description table of acyclic parameters

Index	Parameter Name	Data Type	Description
46~55	GENERIC_FLOAT	Float	10 general-purpose floating-point variables for transferring any Modbus floating-point data to the Profibus PA side
56~65	GENERIC_USIGN32	Unsigned32	10 general purpose 32-bit unsigned integer variables to transfer any Modbus floating point data to the Profibus PA
66~75	GENERIC_USIGN16	Unsigned16	10 general-purpose 16-bit unsigned integer variables for transferring any Modbus floating-point data to the Profibus PA
76~85	GENERIC_USIGN8	Unsigned8	10 general-purpose 8-bit unsigned integer variables to transfer any Modbus floating-point data to the Profibus PA
86~87	GENERIC_OCTET	OctString(32)	2 general-purpose string variables to transfer any Modbus string data to the Profibus PA

4.4 Module Configuration

From the above, we know that the User transformation block in the MC0307 module contains many parameters, and these parameters need to be read from or written to the user board. But what data on the user board can be read and written specifically? Which registers are these data stored on the user board? Therefore, this requires multiple modules to perform initial configuration work.

First, turn the 8th bit M of the DIP switch on the backplane to ON, and the module enters the configuration mode. Connect the module to the computer serial port through the backplane.

Open the Modbus general configuration tool, and add devices by adding serial ports.

After the device is scanned, the basic parameters of the device will be read into the configuration tool. After that, the user can modify the initialization configuration parameters of the module arbitrarily according to their own needs. For the Modbus general configuration tool, see the manual that comes with the tool.

4.4.1 Initialization Parameter Configuration of the Whole Device

The device initialization information of the whole machine includes parameter information closely related to the device such as manufacturer ID and device ID. These parameters are proprietary information of the user's product.

Table 4.8 Initialization parameter table of the whole machine equipment

Parameter Name	Description
Vendor ID	To develop PA devices, you must first become a member of the PI organization before you are allowed to apply for a vendor ID, which is not allowed for non-members. The vendor ID is mainly applied in the device description EDD.
Device ID	Equipment unique identification, need to apply to the PI organization. Members, non-members, the application price is not the same.
Line ID	The identification specified by the line regulation
Device type	String used to describe the type of equipment, length 16 bytes.
Device serial number	Used for the product serial number of the whole equipment, length 16 bytes.
Equipment certification	Used to fill in the equipment certification information, length 32 bytes.

Device installation time		Used to fill in the factory time of the equipment, length 16 bytes.
Serial set address		0：Disable 1：Enable
Bus address configuration information		The following bus address configuration information is valid only when the serial port is set address enabled.
Bus address Attributes	**Read/Write properties**	Read Only
	Function code	03，04
	Data Type	Unsigned8_0，Unsigned8_1
	Register address	The register address where the bus address parameters are stored.

4.4.2 Modbus Communication Parameter Configuration

Modbus communication parameters are the most basic configuration parameters between the module and the user board. Only after configuring these parameters correctly, can the module communicate with the user board correctly.

Table 4.9 Modbus communication parameter list

Parameter Name	Description
Baud rate	0：2400 1：4800 2：9600 3：14400 4：19200
Data bits	0：8 1：7
Calibration method	0：None 1：Even 2：Odd
Interface type	0：TTL 1：RS232 2：RS485
Stop bit	0：One Stop Bit 1：Two Stop Bits
Slave address	This address is the Modbus slave address in normal operation mode, slave address range: 1~255.
CRC check order	CRC check order 0：Normal 1：Swapped
Timeout time	Timeout time range: 300ms~1000ms。
Retry count	Retry times: 1~10。

4.4.3 Cyclic Input and Output Parameter Configuration

As mentioned above, this module contains analog input, analog output, discrete input and discrete output parameters, so this chapter will introduce how these parameters are associated with Modbus slaves. The above four parameters have attributes such as read and write, data type, register address, function code, etc., and can be configured through the Modbus general configuration tool.

- **Read and write attributes**

This attribute describes whether the parameter is read-only, write-only, or readable and writable on the Modbus side.

- **Data Format Properties**

This attribute describes the data type format of the parameter on the Modbus side, its method, etc.

Table 4.10 Data format attribute description table

Index	Name	Data type	Data length	Valid range
1	Float_0123	Single precision floating point	4	
2	Float_1032	Single precision floating	4	

		point		
3	Float_3210	Single precision floating point	4	
4	Float_2301	Single precision floating point	4	
5	Unsigned8_0	Unsigned integer	1	0 - 255
6	Unsigned8_1	Unsigned integer	1	0 - 255
7	Unsigned16_01	Unsigned short integer	2	0 - 65535
8	Unsigned16_10	Unsigned short integer	2	0 - 65535
9	Unsigned32_0123	Unsigned long integer	4	0 - 4294967295
10	Unsigned32_1032	Unsigned long integer	4	0 - 4294967295
11	Unsigned32_3210	Unsigned long integer	4	0 - 4294967295
12	Unsigned32_2301	Unsigned long integer	4	0 - 4294967295
13	Signed8_0	Signed Integer	1	-128 - 127
14	Signed8_1	Signed Integer	1	-128 - 127
15	Signed16_01	Signed Short Integer	2	-32768 - 32767
16	Signed16_10	Signed Short Integer	2	-32768 - 32767
17	Signed32_0123	Signed Long Integer	4	–2,147,483,648 - 2,147,483,647
18	Signed32_1032	Signed Long Integer	4	–2,147,483,648 - 2,147,483,647
19	Signed32_3210	Signed Long Integer	4	–2,147,483,648 - 2,147,483,647
20	Signed32_2301	Signed Long Integer	4	–2,147,483,648 - 2,147,483,647

In the above data format table, the suffix abcd of the data type name DataType_abcd represents the sequence number of the data in the Modbus slave register, 0 represents the lower 8 bits of data in the first registeand 1 represents the upper 8 bits of the first register Data, 2 represents the low 8-bit data in the second register, 3 represents the high 8-bit data in the second register. The module memory adopts little-endian mode, so Unsigned32_0123 means that the data of the Modbus slave registers are assigned to the long integer variables in the module in the original order, while Unsigned32_1032 assigns the data of each register of the Modbus slave after exchanging the high and low bytes to a long variable in the module.

- **Register address attribute**

  This attribute describes the address where the parameter is located in the Modbus memory area.

- **Function code attribute**

  This attribute describes which function code is used to operate the parameter.

Table 4.11 Function code parameter description table

Function Code	Name
1	FC01 Read Coils
2	FC02 Read Discrete Input
3	FC03 Read Holding Register
4	FC04 Read Input Register
5	FC05 Write Single Coils
6	FC06 Write Single Register
16	FC16 Write Multiple Register

Table 4.12 Cyclic input and output parameter configuration table

Data type	Read/Write attribute	Available function code	Available data format	Whether the register address can

				be consecutive
Analog Input	Read Only	03,04	Float_0123, Float_1032, Float_3210, Float_2301, Unsigned32_0123, Unsigned32_1032, Unsigned32_3210, Unsigned32_2301, Unsigned16_01, Unsigned16_10, Signed16_01, Signed16_10, Signed32_0123, Signed32_1032, Signed32_3210, Signed32_2301, Unsigned8_0, Unsigned8_1, Signed8_0, Signed8_1	Yes
Analog output	Write only	06	Unsigned16_01， Unsigned16_10, Signed16_01, Signed16_10, Unsigned8_0, Unsigned8_1, Signed8_0, Signed8_1	NO
		16	Float_0123, Float_1032, Float_3210, Float_2301, Unsigned32_0123, Unsigned32_1032, Unsigned32_3210, Unsigned32_2301, Signed32_0123, Signed32_1032, Signed32_3210, Signed32_2301, Unsigned16_01, Unsigned16_10, Signed16_01, Signed16_10, Unsigned8_0, Unsigned8_1, Signed8_0, Signed8_1	Yes
Discrete input	Read only	01,02	None	Yes
		03,04	Unsigned8_0, Unsigned 8_1	Yes
Discrete output	Write only	05	None	NO
		15	None	Yes
		06	Unsigned8_0， Unsigned8_1	NO
		16	Unsigned8_0， Unsigned8_1	Yes

4.4.4 Non-cyclic Parameter Configuration

As mentioned above, this module contains 5 acyclic parameters. Among them, there are 10 floating-point data, USIGN32 data, USIGN16 data, and USIGN8 data each, and two 32-byte Octet String data. The configuration method of these parameters is exactly the same as that of cycle parameters. It also includes attributes such as read and write, data type, register address, function code, etc., all of which can be configured through the Modbus general configuration tool.

Table 4.13 Acyclic parameter configuration table

Data Type	Read and Write Properties	Available Function Code	Available Data Format	Whether the register address can be consecutive
Floating point data	Reading and writing	03,04,16	Float_0123, Float_1032, Float_3210, Float_2301	Yes
USIGN32 Data	Reading and writing	03,04,16	Unsigned32_0123, Unsigned32_1032, Unsigned32_3210, Unsigned32_2301	Yes
USIGN16 Data	Reading and writing	03,04,06,16	Unsigned16_01, Unsigned16_10	Yes
USIGN8 Data	Reading and writing	03,04,06,16	Unsigned8_0, Signed8_1	Yes
Octet String Data	Reading and writing	03,04,16	Unsigned16_01, Unsigned16_10	Yes

4.4.5 Generates the GSD File

Through the Modbus general configuration tool, some basic information in the GSD file can be configured, and a user-specific device GSD file can be generated. If the user is not satisfied with the generated GSD file, he can refer to the GSD specification or use a special tool to modify the generated GSD file.

When modifying the GSD file by yourself, pay attention to the following points:

1) The content after ";" means comment text, not the actual GSD file description, users can add comment text according to their needs;

2) The Bitmap_Device image has a format requirement, and the file uses the Windows Bitmap format (.bmp), with a length of 70*width of 40 pixels and 16 bits. For compatibility reasons, you can also use Device Independent Bitmap format files (.dib);

Slave_Family is used to specify the slave station type of this product. This parameter is fixed at 12 for PA equipment. You can add @ after 12 to increase the device directory. For example: 12@Microcyber@Module.

4.5 Equipment Usage

After the module configuration is completed, switch back to the normal working mode, and then the module can be embedded into the user's product to form a Profibus PA slave device.

4.5.1 Set the Slave Station Address

When using the whole device, you need to pay attention to the setting method of the device address. This module supports software setting address. Address setting by software includes address setting by bus and address by serial port.

1) Set the address through the bus

When configuring the module, set the "serial port setting address" parameter to prohibit. At this point, the slave station address can be set through the bus command.

2) Set the address through the serial port

When configuring the module, set the "serial port setting address" parameter to enable. At this time, the address of the slave station device comes from the Modbus register where the "bus address register" parameter is located. The user can modify the address of the slave station by modifying this register.

4.5.2 Equipment Cycle Configuration

- GSD file description

A PA device generally supports at least two GSD files: a manufacturer GSD file and a profile GSD file. The GSD file generated above is the manufacturer's GSD file. The 16 function blocks contained in the manufacturer's GSD file generated by MC0307 can all perform cyclic data exchange services with a class 1 master station. Users need to configure the modules of these function blocks.

Table 4.14 GSD Module

Function Block	Module Name	Module Number	Configuration Data
Empty Module	EMPTY_MODULE	1	0x00
AI function block	Analog Input (AI)	2	0x42,0x84,0x08,0x05
AO function block	SP	3	0x82,0x84,0x08,0x05
	SP+READBACK+POS_D	4	0xC6, 0x84, 0x86, 0x08, 0x05, 0x08, 0x05, 0x05, 0x05
	SP+CHECKBACK	5	0xC3, 0x84, 0x82, 0x08, 0x05, 0x0A

	SP+READBACK+POS_D+CHECKBACK	6	0xC7, 0x84, 0x89, 0x08, 0x05, 0x08, 0x05, 0x05, 0x05, 0x0A
	RC_IN+RC_OUT	7	0xC4, 0x84, 0x84, 0x08, 0x05, 0x08, 0x05
	RC_IN+RC_OUT+CHECKBACK	8	0xC5, 0x84, 0x87, 0x08, 0x05, 0x08, 0x05, 0x0A
	SP+RC_IN+RB+RC_OUT+POS_D+CB	9	0xCB, 0x89, 0x8E, 0x08, 0x05, 0x08, 0x05, 0x08, 0x05, 0x08, 0x05, 0x05, 0x05, 0x0A
DI function block	OUT_D	10	0x91
DO function block	SP_D	11	0xA1
	SP_D+RB_D	12	0xC1, 0x81, 0x81, 0x83
	SP_D+CB_D	13	0xC1, 0x81, 0x82, 0x92
	SP_D+RB_D+CB_D	14	0xC1, 0x81, 0x84, 0x93
	RC_IN_D+RC_OUT_D	15	0xC1, 0x81, 0x81, 0x8C
	RC_IN_D+RC_OUT_D+CB_D	16	0xC1, 0x81, 0x84, 0x9C
	SP_D+RC_IN_D+RB_D+RC_OUT_D+CB_D	17	0xC1, 0x83, 0x86, 0x9F

Note: RB = READBACK, CB = CHECKBACK, RC_OUT = RCAS_OUT, RC_IN = RCAS_IN

Each function block occupies a slot, and each slot can have multiple module options.

Table 4.15 GSD input and output data configuration table

Slot	Function Block	Default Module	Optional Module
1	AI function block 1	2	1,2
2	AI function block 2	2	1,2
3	AI function block 3	2	1,2
4	AI function block 4	2	1,2
5	AI function block 5	2	1,2
6	AI function block 6	2	1,2
7	AO function block 1	3	1,3,4,5,6,7,8,9
8	AO function block 2	3	1,3,4,5,6,7,8,9
9	DI function block 1	10	1,10
10	DI function block 2	10	1,10
11	DI function block 3	10	1,10
12	DI function block 4	10	1,10
13	DO function block 1	11	1,11,12,13,14,15,16,17
14	DO function block 2	11	1,11,12,13,14,15,16,17
15	DO function block 3	11	1,11,12,13,14,15,16,17
16	DO function block 4	11	1,11,12,13,14,15,16,17

- Install the GSD file

Take Siemens STEP 7 software as an example, select any project, open the hardware configuration interface, select the "Options→ Install GSD File..." option, and the window for importing GSD files will open.

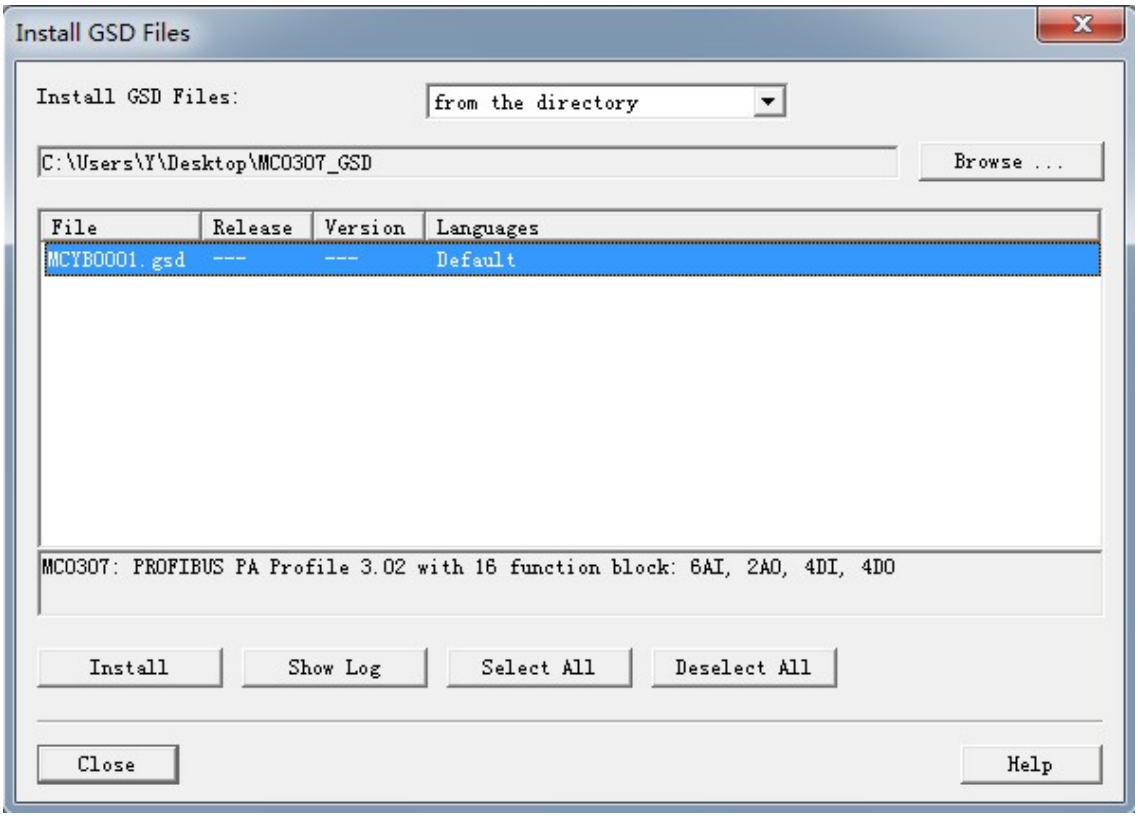


Figure 4.3 Import GSD file window

Click "Browse...", select the path where the GSD file is located, and all GSD files under the current path will be listed. Select the GSD file to be imported, and click "Install". Keep clicking "Yes" until Figure 4.4 appears.

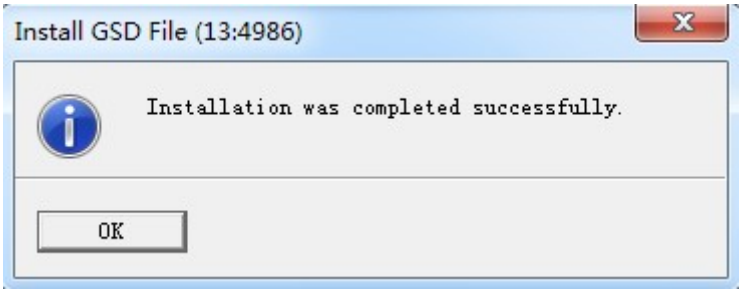


Figure 4.4 Import successfully

- Use GSD files

After installing the GSD file, the module will appear in the tree list on the right side of the hardware configuration interface.

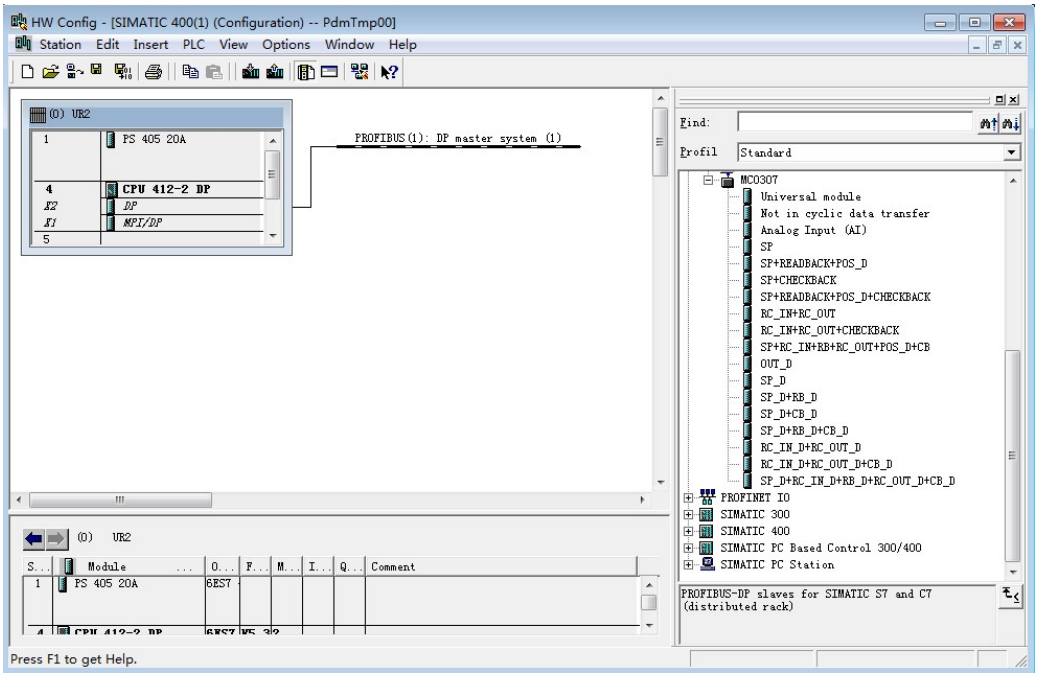
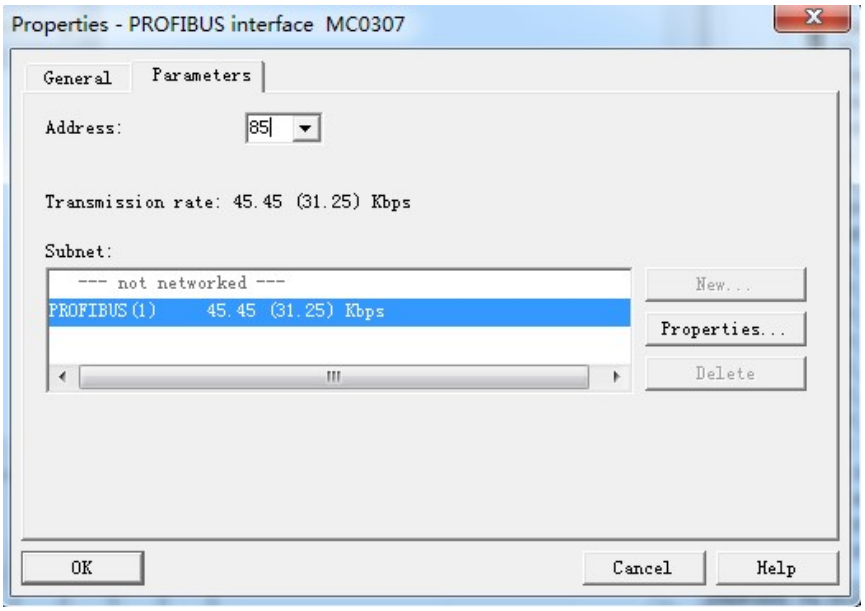


Figure 4.5 Correctly installed equipment

Drag and drop the module onto the DP bus. The properties window will pop up automatically. Configure the module address as your desired address. Here I use address 85.



Click "OK" to complete adding the module.

Select the module in the configuration diagram, and the configuration configuration of the device will appear at the bottom left of the view, as shown in the figure below:

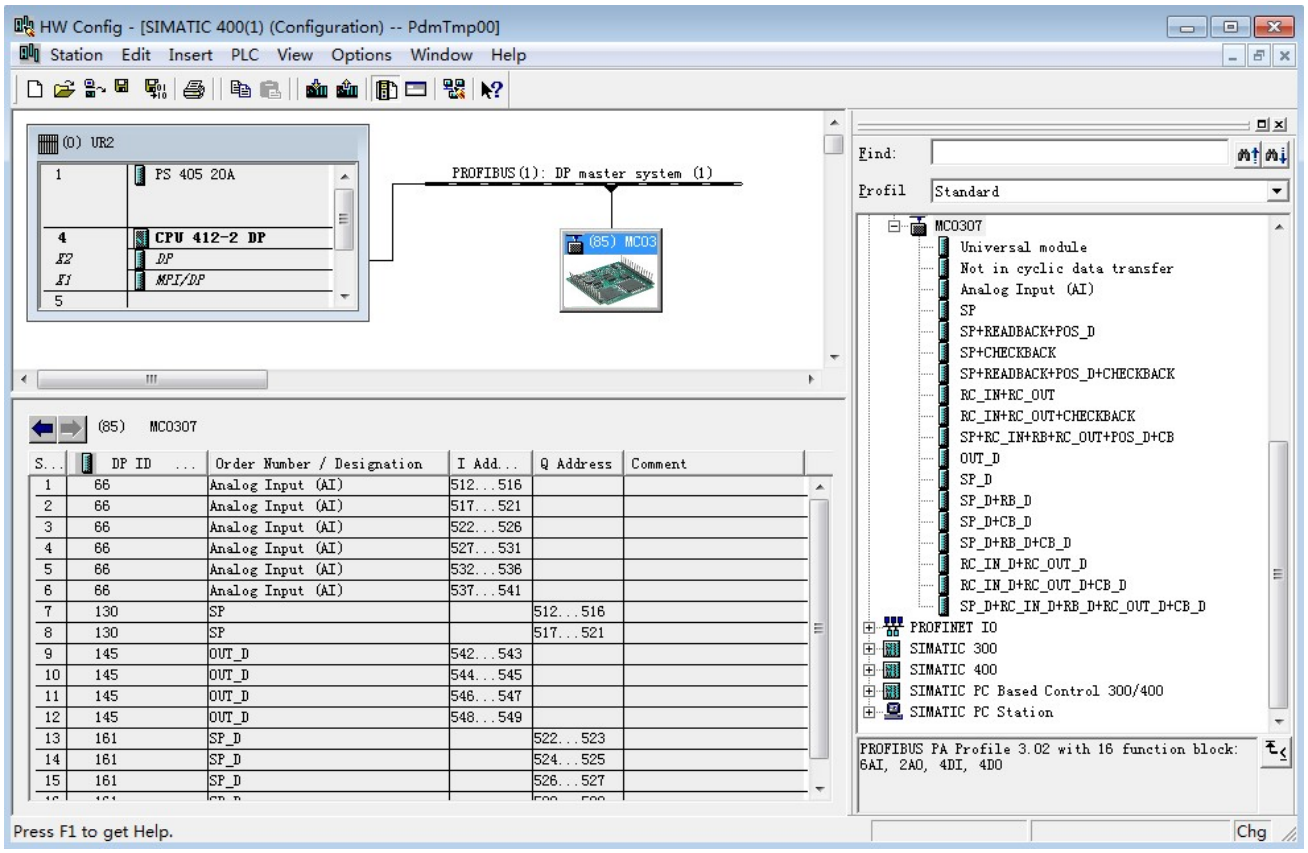


Figure 4.7 Device configuration configuration

When performing hardware configuration, the user makes corresponding configuration adjustments according to actual needs, thereby forming configuration information of the input and output data of the module. See 4.5.2 for the specific meaning of each module option.

- Profile GSD file

  In addition to the manufacturer's GSD file, users can also use the GSD file defined by the line: pa139760.gsd.

  But note that since the modules placed in each slot of the device have been specified (see Table 4.15), it can work normally only when the configuration is correct.

Chapter 5 GSD file, ID number, and Product Certification Testing

5.1 GSD file, ID number, and the Introduction of the Product Certification Test

5.1.1 GSD File (Electronic Data Sheet)

Each PROFIBUS slave station or a class of master station has a device description file called a GSD file. This file is used to describe the features of the PROFIBUS device.

The GSD file contains all the defined parameters of the device, including:

- ✓ Supported Baud rate;
- ✓ Supported length of information;
- ✓ Quantity of input / output data;
- ✓ The meaning of the diagnostic information;
- ✓ Optional module types and so on.

GSD files are text files that can be edited with notepad type software.

Regardless of the system environment used, you need to configure the device according to the GSD file.

The international PROFIBUS Organization PI provided the GSD file editing software: GSD-Editor. The software can check the GSD files edited by the user according to the Profibus technical standard format. The software's "help" is rich in content and is a quick way to learn about GSD file technology. But you must become a PI organization member to download and obtain it.

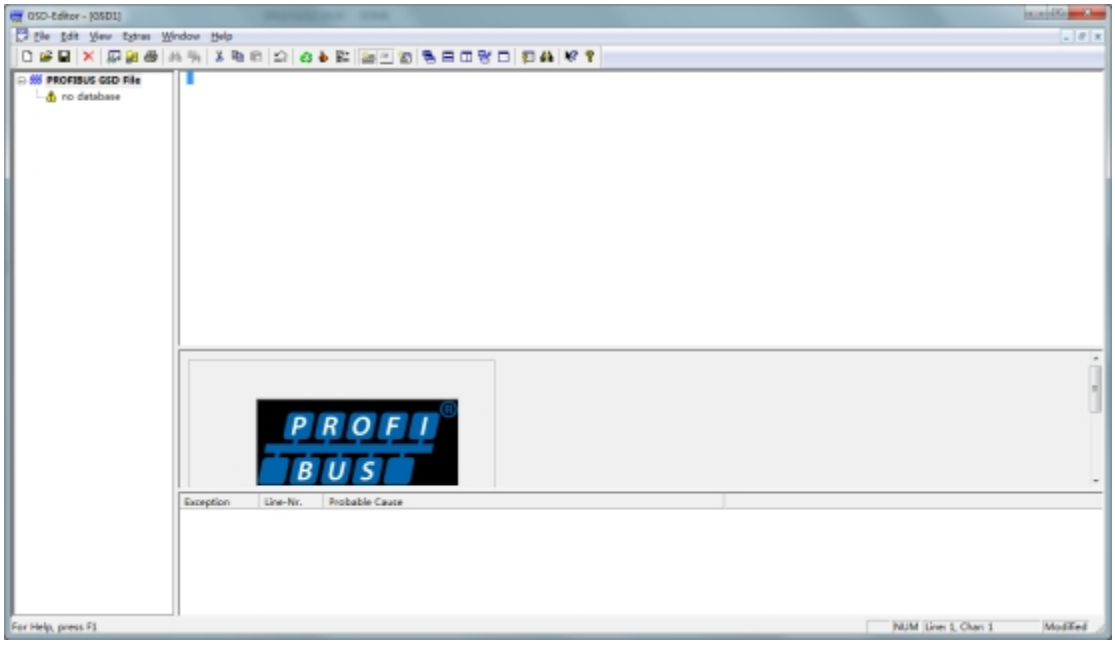


Figure 5.1 The GSD-Editor opens an empty file

5.1.2 ID Number (Ident Number)

Every PROFIBUS device should have a unique ID number. Users can apply to the international PROFIBUS Organization PI for the product ID number by entrusting the "China PROFIBUS Organization CPA".

Each member manufacturer can also apply for the manufacturer ID number, and non-members are not allowed to apply.

CPA tel: 010-63405107 Contact: Wang Jing

For CPA contact information, please check the website http://www.pi-china.org/

5.1.3 Product certification test

PROFIBUS Product certification testing is not mandatory. However, if the product passes the certification test, it can give the design institute and other end users stronger confidence, and facilitate the product to participate in the project bidding and market development.

In China, the PROFIBUS organization CPA can test the product, and after passing the test, you can entrust the CPA to apply for the certification certificate to the international PROFIBUS organization PI.

CPA tel: 010-63322089 Contact: Liu Dan

For CPA contact information, please check the website http://www.pi-china.org/

5.2 GSD File and ID Number of the User Product

Since the module is sold as an OEM, users have their own intellectual property rights and brand for the PROFIBUS devices developed by this module. Therefore, the user product can not use the ID number and GSD file name of the module.

The user can replace the company name, product model, series number with the user product information based on the GSD file of this module, and can form the GSD file of the user's own product.

GSD files are generally named according to the following rules, consisting of an 8-bit string, 4 higher for the manufacturer and 4 lower for the ID number. For example, in the MCYB0001.gsd file, MCYB is short for Microcyber, and 0001 is the ID number of this product. The 4-bit vendor name abbreviation is usually defined by the user when applying for the ID number.

The ID number of the module configuration must match the ID number in the GSD file to connect.

5.3 Device Description File

Profibus PA There are two kinds of device mainstream device description files: EDD file and device DTM.

Since both documents are complicated, no description of the equipment description document is made in this manual. This module provides an EDD file template, which users can modify according to their own needs. device File, to implement the most basic EDD function.

Modifying the number of parameters may lead to EDD failure to read some parameters, which is required by matching the parameters in the EDD file with the configured parameters modification.

If there have any EDD document or equipment DTM needs, please contact the company.

Chapter 6 Maintenance

- Simple Maintenance

LED	Color	Normal	Abnormal	Reason	Solution
PA communication	Blue	blink	off	No PA communication	Check the PA host equipment and the interface equipment
				Power failure	Check the power supply and connection
				Internal failure	Contact technical support
			on	No PA communication	Check the PA host equipment and the interface equipment
				Internal failure	Contact technical support

- Daily maintenance means cleaning device only.
- Fault maintenance: Please return to the factory if there's fault.

Chapter 7 Technical Specification

7.1 Basic Parameters

Measuring object	Modbus RTU slave unit
PA bus power supply	9～32VDC
Quiescent current	≤14mA
Bus protocol	2-wire，PA protocol
Isolation voltage	Modbus and FF/PA buses are not isolated
Temperature range	-40℃～85℃
Humidity range	5～95%RH
Start time	≤5S
Refresh time	0.2S

7.2 Performance Index

Electromagnetic compatibility	Compliance with the immunity requirements of industrial sites in GB / T 18268.1-2010 Electromagnetic compatibility requirements for Measurement, Control and Laboratory-Part 1: General Requirements GB / T 18268.23-2010. Electromagnetic compatibility requirements for electrical equipment for measurement, Control and laboratory use-Part 23: Test configuration, working conditions and performance criteria for integrated or remote signal conditioning transmitters

7.3 Physical Characteristics

Weight	16 g
Construction materials	Coating: polyester epoxy resin.

7.4 Default Communication Parameters

Slave address	1
Baud rate	9600
Data bit	8
Stop bit	1
Verification	EVEN
CRC verification	Low byte in front

7.5 Support for the Modbus Function Code

1	Read the coil status
2	Read the discrete input states
3	Read the hold-register value
4	Read the input register value
5	Write the coil
6	Write the single register values
15	Write multiple coils
16	Write the multiple register values

MICROCYBER CORPORATION

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