MICROCYBER

G0306

Modbus to DP gateway

User Manual

Warning

1. It is prohibited to disassemble and assemble the gateway by yourself.
2. Please check whether the power supply voltage of the gateway meets the power supply voltage requirements in the user manual.

Version：**V2.1**

Disclaimer

The contents of this manual have been checked to confirm the consistency of the described hardware and software. Since errors cannot be completely eliminated, absolute agreement cannot be guaranteed. However, we will periodically check the data in this manual and make necessary corrections in subsequent editions. Any suggestions for improvement are welcome.

Microcyber Corporation 2021

Technical data is subject to change

Company Profile

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. The company successfully developed the FF H1 fieldbus protocol stack which is number one to be approved

internationally in China, and the Industrial Ethernet Protocol(HSE) which is number one to be approved in China, and the domestic first fieldbus instrument which has a function of national-level intrinsically safe explosion--proof and safety barrier. Also Microcyber participated in the drafting of the domestic first Ethernet-based industrial automation protocol standards (Ethernet for Plant Automation, EPA). As a result, serial products are composed of configuration, control software, embedded software, control system, instrument chip to the

OEM board, and make Microcyber be an industrial automation products provider in full range, and also further Microcyber‟s leading position in the field of fieldbus technology.

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

Microcyber passes 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.

Microcyber is making progress with China.

Contents

Chapter 1 Overview

Product name: Modbus to DP gateway

Product number：G0306

This gateway realizes the conversion function from Modbus RTU to PROFIBUS DP. Multiple slave devices that comply with the Modbus RTU communication rules can be connected to the PROFIBUS DP network. The Modbus end can be used as a master station or as a slave station. This function can be switched by special function keys.

Figure 1 G0306 Modbus to DP gateway

1.1 Equipment classification

This gateway is divided into two models according to the Modbus interface (consistent with the actual selection)：

Product model	Modbus interface
GW-MODB-DP-RS485	RS485
GW-MODB-DP-RS232	RS232

This gateway is distinguished according to the communication method (function distinction, use the following name to facilitate document description, not the actual selection, see the appendix for the actual selection)：

Product model	Modbus side	PROFIBUS side
G0306-MS	Master station	slave station
G0306-SS	slave station	slave station

Note：

1) The V1.0 version only implements the GW-MODB-DP-RS485 version, and the GW-MODB-DP-RS232

version will be added later, so stay tuned.

2) G0306 Modbus to DP gateway can switch freely between G0306-MS and G0306-SS through special function buttons according to user needs, see 2.2.7 for the switching method.

1.2 Shape structure

1.2.1 Dimensions of the gateway

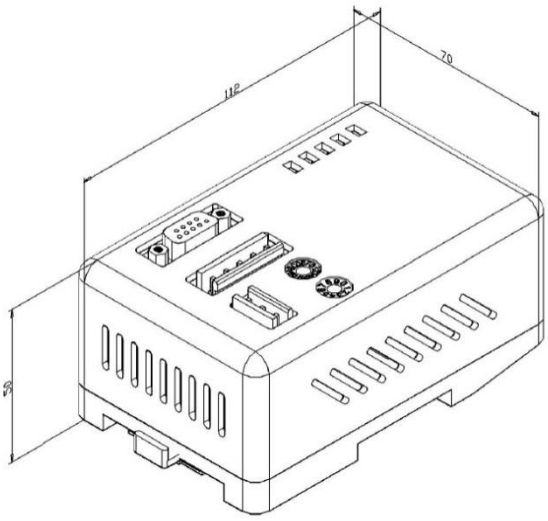


Figure 2 Gateway outline dimension drawing（112*70*50，unit mm）

1.2.2 Gateway structure diagram

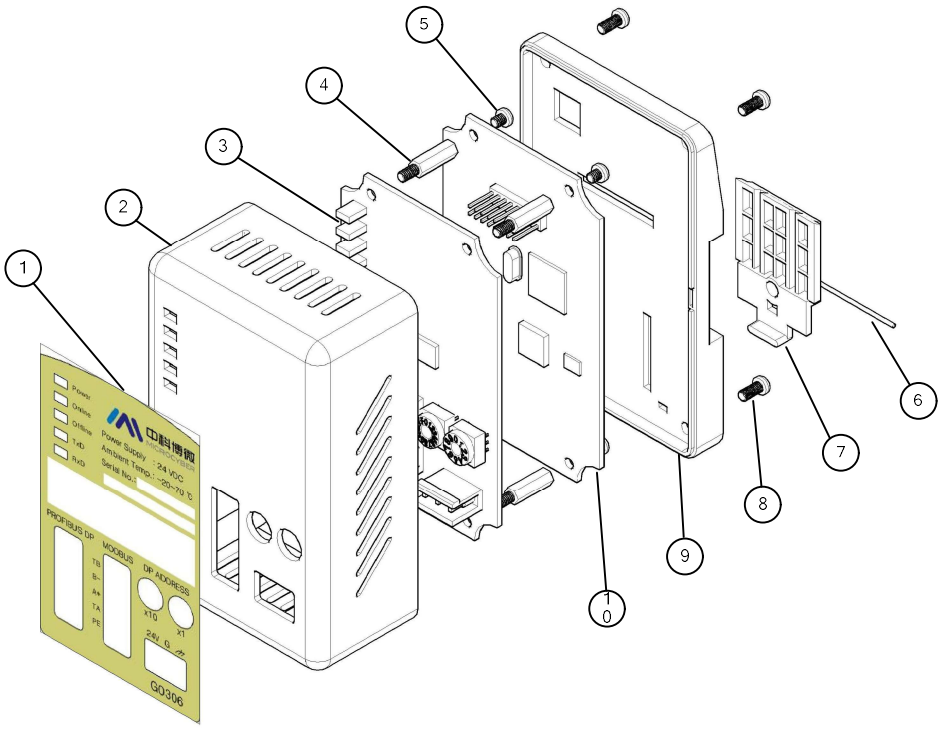


Figure 3 Gateway structure diagram

1	Film	2	Upper cover	3	Interface card	4	Hexagonal prism

5	Wide-edge screw	6	iron wire	7	Clip	8	Wide-edge screw
9	Base	10	Communication card				

Chapter 2 Installation

2.1. DIN Rail installation

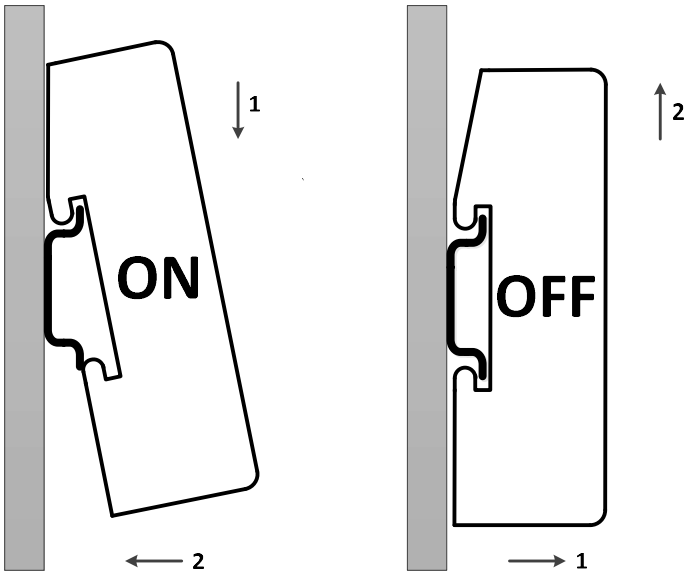
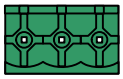


Figure 4 DIN Rail installation diagram

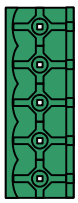
2.2. Gateway interface

2.2.1. Power interface



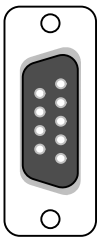
No.	Terminal name	Terminal usage
1	24V	Connect the positive pole of 24V DC power supply
2	G	Connect to negative pole of 24V DC power supply
3	PE	Connect twisted pair shield

2.2.2. Modbus-RS485 interface（Apply to GW-MODB-DP-RS485）



No.	Terminal name	Terminal usage
1	TB	Short-circuit the enable terminal with B-
2	B-	Connect to Modbus bus B
3	A+	Connect to Modbus bus A
4	TA	Short-circuit with A+ to enable terminal
5	PE	Connect twisted pair shield

2.2.3. Modbus-RS232 interface（Apply to GW-MODB-DP-RS232）

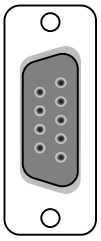


No.	Terminal name	Terminal usage
1	NC	Hang in the air
2	TxD	Connect user device RxD
3	RxD	Connect user device TxD
4	NC	Hang in the air
5	GND	Connect user device GND
6-9	NC	Hang in the air

Note: Users should make their own RS232 cable according to the interface

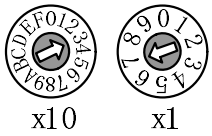
definition.

2.2.4. PROFIBUS DP Bus interface



No.	Terminal name	Terminal usage
1-2	NC	Hang in the air
3	RxD/TxD-P	Receive/send data, line B (red)
4	CNTR-P	Repeater direction control
5	DGND	Data ground (reference voltage to VP)
6	VPa	Power supply +5V (for example, for bus terminator)
7	NC	Hang in the air
8	RxD/TxD-N	Receive/send data, line A (green)
9	NC	Hang in the air

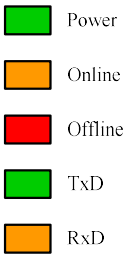
2.2.5. PROFIBUS Address configuration interface



Address dial switch	Description
16-bit rotary dial switch x10	Each scale represents 10, and the range is 0-160；
10-position knob dial switch x1	Each scale represents 1, the range is 0-9；

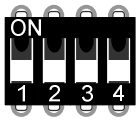
Note: When the address is greater than 125, the fixed address is 125.
As shown in the figure, address=3*10+7*1=37

2.2.6. LED Indicator light



Indicator name	Colour	Indicator usage
Power	Green	Device power indicator
Online	Yellow	PROFIBUS Enter data exchange
Offline	Red	PROFIBUS Data exchange not entered
TxD	Green	Modbus Send indicator
RxD	Yellow	Modbus Receive indicator

2.2.7. Special function interface



No.	Terminal name	Terminal usage
1	MS	Modbus Master-slave switch Off：G0306-MS On：G0306-SS
2	Not Used	Not yet used
3	Not Used	Not yet used
4	Not Used	Not yet used

Chapter 3 Working principle

G0306 Modbus to DP gateway is a gateway device that supports both Modbus RTU protocol and PROFIBUS DP protocol. It is a PROFIBUS DP slave device. As a Modbus master (G0306-MS), it can theoretically connect up to 31 Modbus slave devices to the PROFIBUS DP network (this is limited to the 485 bus environment, and it is recommended not to More than 10 Modbus slave devices, so as not to affect the communication quality), as a Modbus slave (G0306-SS), it can be connected to the Modbus master via the RS485/RS2322 interface.

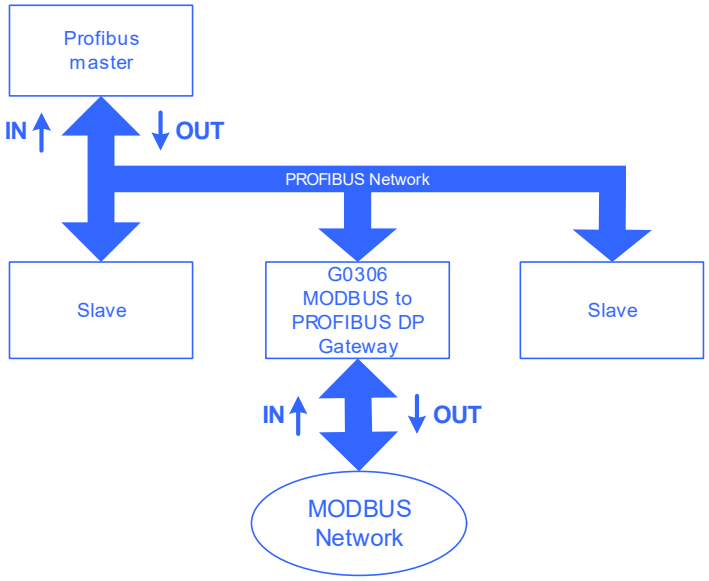


Figure 5 Gateway system connection diagram

3.1. Gateway as Modbus master（G0306-MS）

The working mechanism of G0306-MS is to realize the conversion of Modbus data to DP data by configuring the module. Each module can be configured with 1 Modbus message. G0306-MS has 39 slots and 209 modules. Among them, slots 1 and 2 have fixed functions, and 37 slots are available. Each slot can be configured with 1 module, which is equivalent to a maximum of 37 Modbus messages.

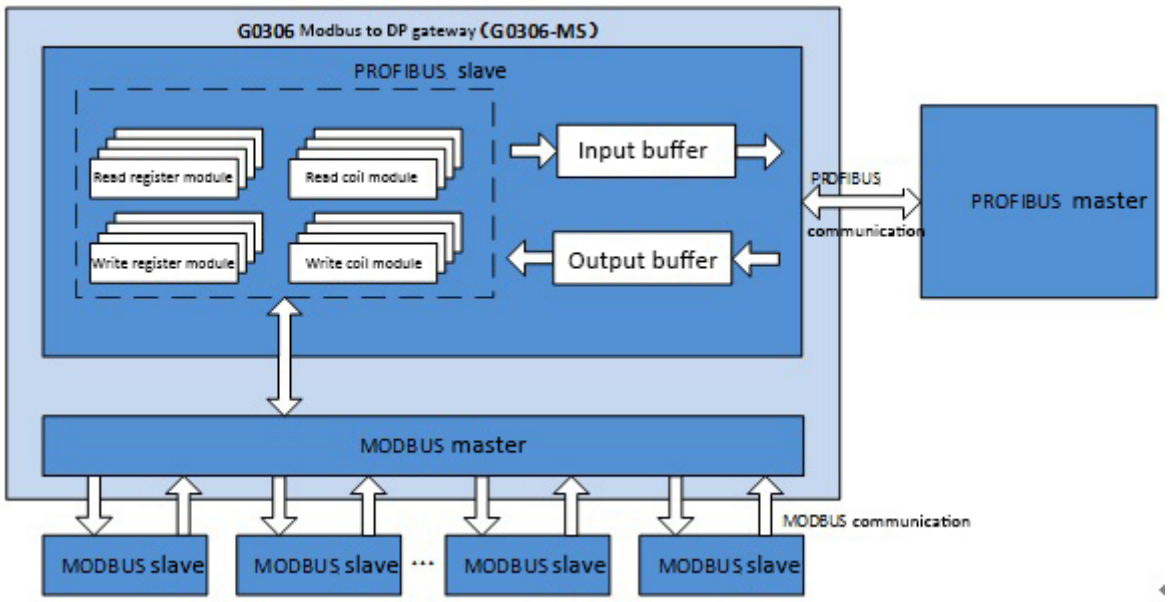


Figure 6 Block diagram of the gateway（G0306-MS）

3.2. Gateway as Modbus slave（G0306-SS）

The working mechanism of G0306-SS is to realize the mapping of the input and output buffers of Profibus to the Modbus storage area by configuring the module, so as to facilitate the Modbus master station to read the desired data in the corresponding Modbus storage area.

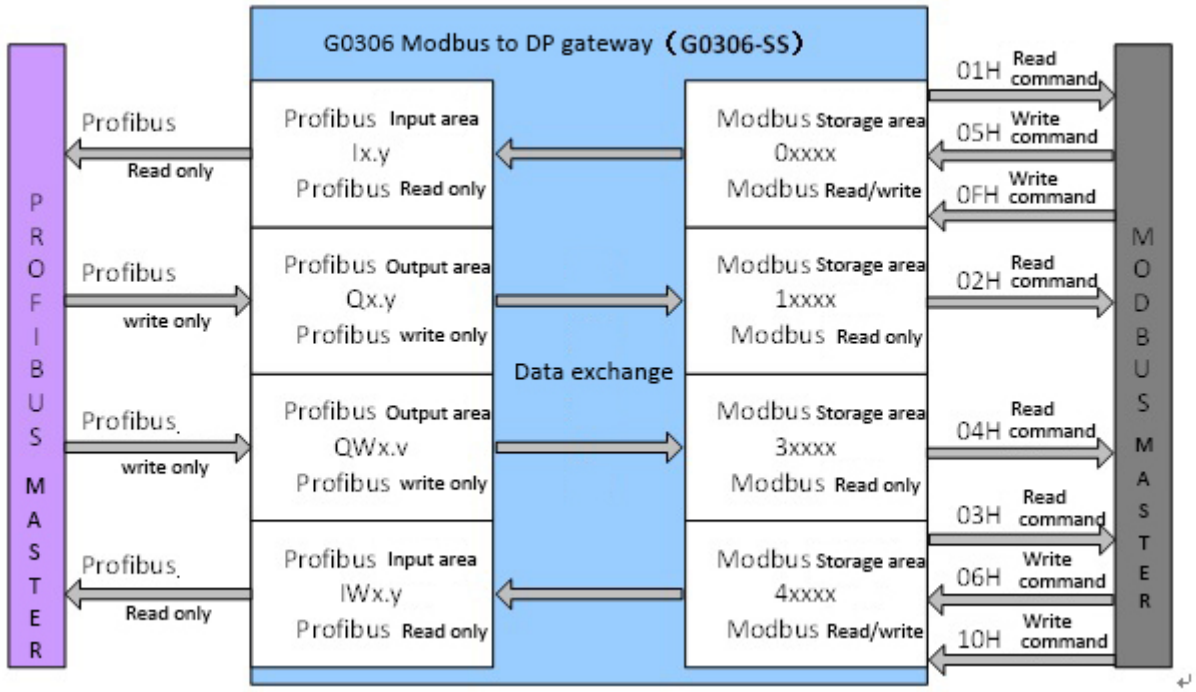


Figure 7 Block diagram of the gateway（G0306-SS）

G0306-SS has 20 slots and 67 modules. Among them, slots 1 and 2 have fixed functions, and 18 slots are available. Each slot can be configured with 1 module, which is equivalent to the establishment of up to 18 Profibus input and output buffers to the Modbus storage area.

G0306-SS is the same as other Modbus slave devices, with 4 Modbus storage areas：

Table 1 Modbus storage area

Store ID	Name	Type	ModbusMaster station read/write	Storage unit address range
0xxxx	Coil	Bit	Read/write	00001~01944，Total 1944bit=243byte
1xxxx	Discrete input	Bit	Read only	10001~11944，Total 1944bit=243byte
4xxxx	Holding register	Word	Read/write	40001~40121，Total 121word=242byte
3xxxx	Input register	Word	Read only	30001~30121，Total 121word =242byte

Note: 0xxxx, 1xxxx, 4xxxx, 3xxxx are storage area identifiers. Among them, 0, 1, 4, and 3 respectively identify 4 different storage areas. xxxx is the address of the storage unit in decimal, and after subtracting 1, it is the actual starting address. For example: Discrete input start address 0000 corresponds to address 10001 in the device.

Chapter 4 Gateway configuration

4.1. Topology

4.1.1. Network Topology

G0306 Modbus to DP gateway adopts RS485 transmission technology, and the transmission rate can be selected from 9.6kbit/s to 12Mbit/s. All devices are connected to a bus structure and the same transmission rate is selected. Up to 32 stations (master station or slave station) can be connected in a bus segment. There is an active bus terminator at the beginning and end of each bus segment. Both bus terminal units have a permanent power supply to ensure error-free operation. The bus terminal is usually connected to the device or in the connector (Note: There is no bus terminal on the PROFIBUS side of this gateway. If it is used as a terminal, please use a connector with a terminal). If there are more than 32 stations in the implementation or the network area needs to be expanded, repeaters must be used to connect each bus segment.

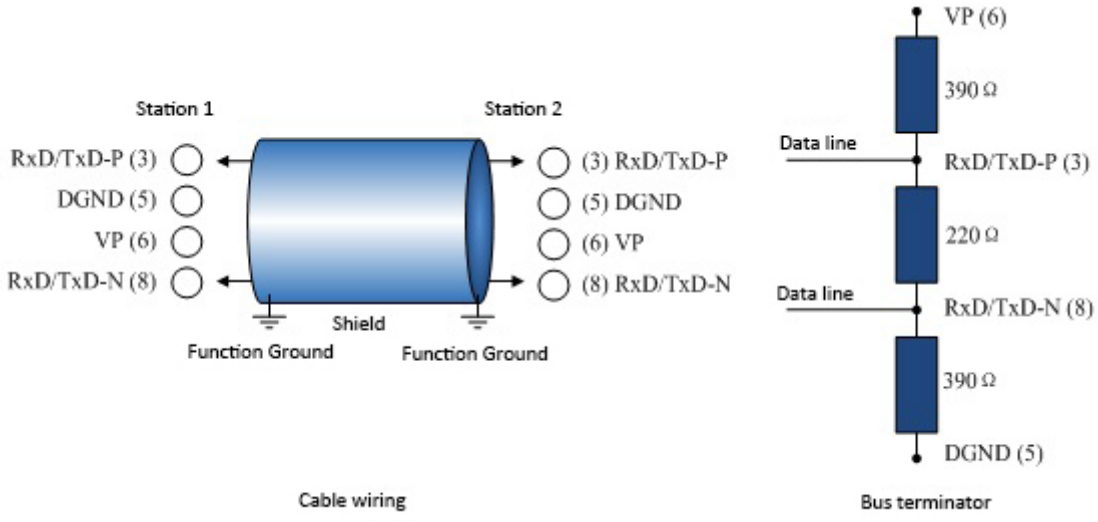


Figure 8 RS485 Transmission technology wiring and bus terminator

4.1.2. Cables and connectors

For different applications, there are different types of cables (type AD) available on the market, which can be used for the connection between equipment and network components (such as segment couplers, linkers and Repeater). When using RS485 transmission technology, PI recommends using cable A.

Table 2 Transmission rate and range of cable type A

Transmission rate（kbit/s）	Range of each bus segment（m）
9.6，19.2，45.45，93.75	1200
187.5	1000
500	400
1500	200
3000, 6000, 12000	100
These values apply to Type A cables with the following characteristics： Impedance：135Ω~165Ω　　Capacitance：≤20pF/m Loop resistance：≤110Ω/km　　Wire diameter：＞0.64mm Conductor cross-sectional area：＞0.34mm^2	

4.2. Set gateway address

Note: This gateway does not support device address setting through the Set_Address service, only hardware address setting is supported.

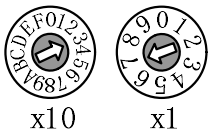
First, please set the address when the power is off.

The PROFIBUS address configuration interface contains two rotary dial switches, and the PROFIBUS address is set by these two rotary dial switches.

The setting method is as follows：

*PROFIBUS bus address=(value of x10 dial switch)*10+(value of x1 dial switch)*

E.g：



PROFIBUS address=3*10+7=37

4.3. Gateway cyclic configuration

The cyclic configuration of Profibus devices is realized through GSD files. The Profibus network master station executes the device initialization process through the GSD file. The master station sends the configured parameters and input and output data configuration to the slave station. The slave station can enter the cyclic data exchange state after the detection is correct.

4.3.1. GSD File description

The GSD file contains software and hardware version, bus baud rate, cyclic data exchange related information, etc. Since the gateway has the difference between MS and SS, our company has applied for two device IDs for this product, which are used for G0306-MS and G0306-SS respectively. The user needs to select the corresponding GSD file according to the actual configuration (see 2.2.7 for details)

1) G0306-MS GSD file description

The GSD file name for G0306-MS is：MCYB0F1A.GSE

This GSD file contains 39 slots, 209 modules, and supports up to 237 user parameters.

- **Module description**

Table 3 GSD module description of G0306-MS

Module No.	Module name	Input data length (bytes)	Output data length (bytes）	Description
1	empty	0	0	Empty module
2	status	1	0	Modbus Communication status module
3	control	0	1	Modbus Communication control module
4	read 8 bits(0xxxx)	1	0	Use function code 1 to read 8 coil data from a Modbus slave station
5	read 8 bits(1xxxx)	1	0	Use function code 2 to read 8 discrete inputs from a Modbus slave
6	read 16 bits(0xxxx)	2	0	Use function code 1 to read 16 coil data from a Modbus slave station
7	read 16 bits(1xxxx)	2	0	Use function code 2 to read 16 discrete inputs from a Modbus slave
8	read 24 bits(0xxxx)	3	0	Use function code 1 to read 24 coil data from a Modbus slave station

Module No.	Module name	Input data length (bytes)	Output data length (bytes）	Description
9	read 24 bits(1xxxx)	3	0	Use function code 2 to read 24 discrete inputs from a Modbus slave
10	read 32 bits(0xxxx)	4	0	Use function code 1 to read 32 coil data from a Modbus slave station
11	read 32 bits(1xxxx)	4	0	Use function code 2 to read 32 discrete inputs from a Modbus slave
12	read 40 bits(0xxxx)	5	0	Use function code 1 to read 40 coil data from a Modbus slave station
13	read 40 bits(1xxxx)	5	0	Use function code 2 to read 40 discrete inputs from a Modbus slave
14	read 48 bits(0xxxx)	6	0	Use function code 1 to read 48 coil data from a Modbus slave station
15	read 48 bits(1xxxx)	6	0	Use function code 2 to read 48 discrete inputs from a Modbus slave
16	read 56 bits(0xxxx)	7	0	Use function code 1 to read 56 coil data from a Modbus slave
17	read 56 bits(1xxxx)	7	0	Use function code 2 to read 56 discrete inputs from a Modbus slave
18	read 64 bits(0xxxx)	8	0	Use function code 1 to read 64 coil data from a Modbus slave station
19	read 64 bits(1xxxx)	8	0	Use function code 2 to read 64 discrete inputs from a Modbus slave
20	read 72 bits(0xxxx)	9	0	Use function code 1 to read 72 coil data from a Modbus slave
21	read 72 bits(1xxxx)	9	0	Use function code 2 to read 72 discrete inputs from a Modbus slave
22	read 80 bits(0xxxx)	10	0	Use function code 1 to read 80 coil data from a Modbus slave station
23	read 80 bits(1xxxx)	10	0	Use function code 2 to read 80 discrete inputs from a Modbus slave
24	read 88 bits(0xxxx)	11	0	Use function code 1 to read 88 coil data from a Modbus slave
25	read 88 bits(1xxxx)	11	0	Use function code 2 to read 88 discrete inputs from a Modbus slave
26	read 96 bits(0xxxx)	12	0	Use function code 1 to read 96 coil data from a Modbus slave station
27	read 96 bits(1xxxx)	12	0	Use function code 2 to read 96 discrete inputs from a Modbus slave
28	read 104 bits(0xxxx)	13	0	Use function code 1 to read 104 coil data from a Modbus slave station
29	read 104 bits(1xxxx)	13	0	Use function code 2 to read 104 discrete inputs from a Modbus slave
30	read 112 bits(0xxxx)	14	0	Use function code 1 to read 112 coil data from a Modbus slave station
31	read 112 bits(1xxxx)	14	0	Use function code 2 to read 112 discrete inputs from a Modbus slave
32	read 120 bits(0xxxx)	15	0	Use function code 1 to read 120 coil data from a Modbus slave
33	read 120 bits(1xxxx)	15	0	Use function code 2 to read 120 discrete inputs from a Modbus slave
34	read 128 bits(0xxxx)	16	0	Use function code 1 to read 128 coil data from a Modbus slave
35	read 128 bits(1xxxx)	16	0	Use function code 2 to read 128 discrete inputs from a Modbus slave
36	read 136 bits(0xxxx)	17	0	Use function code 1 to read 136 coil data from a Modbus slave station
37	read 136 bits(1xxxx)	17	0	Use function code 2 to read 136 discrete inputs from a Modbus slave
38	read 144 bits(0xxxx)	18	0	Use function code 1 to read 144 coil data from a Modbus

Module No.	Module name	Input data length (bytes)	Output data length (bytes）	Description
				slave station
39	read 144 bits(1xxxx)	18	0	Use function code 2 to read 144 discrete inputs from a Modbus slave
40	read 152 bits(0xxxx)	19	0	Use function code 1 to read 152 coil data from a Modbus slave station
41	read 152 bits(1xxxx)	19	0	Use function code 2 to read 152 discrete inputs from a Modbus slave
42	read 160 bits(0xxxx)	20	0	Use function code 1 to read 160 coil data from a Modbus slave
43	read 160 bits(1xxxx)	20	0	Use function code 2 to read 160 discrete inputs from a Modbus slave
44	read 168 bits(0xxxx)	21	0	Use function code 1 to read 168 coil data from a Modbus slave station
45	read 168 bits(1xxxx)	21	0	Use function code 2 to read 168 discrete inputs from a Modbus slave
46	read 176 bits(0xxxx)	22	0	Use function code 1 to read 176 coil data from a Modbus slave station
47	read 176 bits(1xxxx)	22	0	Use function code 2 to read 176 discrete inputs from a Modbus slave
48	read 184 bits(0xxxx)	23	0	Use function code 1 to read 184 coil data from a Modbus slave station
49	read 184 bits(1xxxx)	23	0	Use function code 2 to read 184 discrete inputs from a Modbus slave
50	read 192 bits(0xxxx)	24	0	Use function code 1 to read 192 coil data from a Modbus slave station
51	read 192 bits(1xxxx)	24	0	Use function code 2 to read 192 discrete inputs from a Modbus slave
52	read 200 bits(0xxxx)	25	0	Use function code 1 to read 200 coil data from a Modbus slave station
53	read 200 bits(1xxxx)	25	0	Use function code 2 to read 200 discrete inputs from a Modbus slave
54	read 208 bits(0xxxx)	26	0	Use function code 1 to read 208 coil data from a Modbus slave station
55	read 208 bits(1xxxx)	26	0	Use function code 2 to read 208 discrete inputs from a Modbus slave
56	read 216 bits(0xxxx)	27	0	Use function code 1 to read 216 coil data from a Modbus slave station
57	read 216 bits(1xxxx)	27	0	Use function code 2 to read 216 discrete inputs from a Modbus slave
58	read 224 bits(0xxxx)	28	0	Use function code 1 to read 224 coil data from a Modbus slave station
59	read 224 bits(1xxxx)	28	0	Use function code 2 to read 224 discrete inputs from a Modbus slave
60	read 232 bits(0xxxx)	29	0	Use function code 1 to read 232 coil data from a Modbus slave station
61	read 232 bits(1xxxx)	29	0	Use function code 2 to read 232 discrete inputs from a Modbus slave
62	read 240 bits(0xxxx)	30	0	Use function code 1 to read 240 coil data from a Modbus slave station
63	read 240 bits(1xxxx)	30	0	Use function code 2 to read 240 discrete inputs from a Modbus slave
64	read 248 bits(0xxxx)	31	0	Use function code 1 to read 248 coil data from a Modbus slave
65	read 248 bits(1xxxx)	31	0	Use function code 2 to read 248 discrete inputs from a Modbus slave
66	read 256 bits(0xxxx)	32	0	Use function code 1 to read 256 coil data from a Modbus slave station
67	read 256 bits(1xxxx)	32	0	Use function code 2 to read 256 discrete inputs from a Modbus slave

Module No.	Module name	Input data length (bytes)	Output data length (bytes）	Description
68	read 1 Words(4xxxx)	2	0	Use function code 3 to read 1 register data from a Modbus slave
69	read 1 Words(3xxxx)	2	0	Use function code 4 to read 1 register data from a Modbus slave
70	read 2 Words(4xxxx)	4	0	Use function code 3 to read 2 register data from a Modbus slave
71	read 2 Words(3xxxx)	4	0	Use function code 4 to read 2 register data from a Modbus slave
72	read 3 Words(4xxxx)	6	0	Use function code 3 to read 3 register data from a Modbus slave
73	read 3 Words(3xxxx)	6	0	Use function code 4 to read 3 register data from a Modbus slave
74	read 4 Words(4xxxx)	8	0	Use function code 3 to read 4 register data from a Modbus slave
75	read 4 Words(3xxxx)	8	0	Use function code 4 to read 4 register data from a Modbus slave
76	read 5 Words(4xxxx)	10	0	Use function code 3 to read 5 register data from a Modbus slave
77	read 5 Words(3xxxx)	10	0	Use function code 4 to read 5 register data from a Modbus slave
78	read 6 Words(4xxxx)	12	0	Use function code 3 to read 6 register data from a Modbus slave
79	read 6 Words(3xxxx)	12	0	Use function code 4 to read 6 register data from a Modbus slave
80	read 7 Words(4xxxx)	14	0	Use function code 3 to read 7 register data from a Modbus slave
81	read 7 Words(3xxxx)	14	0	Use function code 4 to read 7 register data from a Modbus slave
82	read 8 Words(4xxxx)	16	0	Use function code 3 to read 8 register data from a Modbus slave
83	read 8 Words(3xxxx)	16	0	Use function code 4 to read 8 register data from a Modbus slave
84	read 9 Words(4xxxx)	18	0	Use function code 3 to read 9 register data from a Modbus slave
85	read 9 Words(3xxxx)	18	0	Use function code 4 to read 9 register data from a Modbus slave
86	read 10 Words(4xxxx)	20	0	Use function code 3 to read 10 register data from a Modbus slave
87	read 10 Words(3xxxx)	20	0	Use function code 4 to read 10 register data from a Modbus slave
88	read 11 Words(4xxxx)	22	0	Use function code 3 to read 11 register data from a Modbus slave
89	read 11 Words(3xxxx)	22	0	Use function code 4 to read 11 register data from a Modbus slave
90	read 12 Words(4xxxx)	24	0	Use function code 3 to read 12 register data from a Modbus slave
91	read 12 Words(3xxxx)	24	0	Use function code 4 to read 12 register data from a Modbus slave
92	read 13 Words(4xxxx)	26	0	Use function code 3 to read 13 register data from a Modbus slave
93	read 13 Words(3xxxx)	26	0	Use function code 4 to read 13 register data from a Modbus slave
94	read 14 Words(4xxxx)	28	0	Use function code 3 to read 14 register data from a Modbus slave
95	read 14 Words(3xxxx)	28	0	Use function code 4 to read 14 register data from a Modbus slave
96	read 15 Words(4xxxx)	30	0	Use function code 3 to read 15 register data from a Modbus slave
97	read 15 Words(3xxxx)	30	0	Use function code 4 to read 15 register data from a

Module No.	Module name	Input data length (bytes)	Output data length (bytes）	Description
				Modbus slave
98	read 16 Words(4xxxx)	32	0	Use function code 3 to read 16 register data from a Modbus slave
99	read 16 Words(3xxxx)	32	0	Use function code 4 to read 16 register data from a Modbus slave
100	read 18 Words(4xxxx)	36	0	Use function code 3 to read 18 register data from a Modbus slave
101	read 18 Words(3xxxx)	36	0	Use function code 4 to read 18 register data from a Modbus slave
102	read 20 Words(4xxxx)	40	0	Use function code 3 to read 20 register data from a Modbus slave
103	read 20 Words(3xxxx)	40	0	Use function code 4 to read 20 register data from a Modbus slave
104	read 22 Words(4xxxx)	44	0	Use function code 3 to read 22 register data from a Modbus slave
105	read 22 Words(3xxxx)	44	0	Use function code 4 to read 22 register data from a Modbus slave
106	read 24 Words(4xxxx)	48	0	Use function code 3 to read 24 register data from a Modbus slave
107	read 24 Words(3xxxx)	48	0	Use function code 4 to read 24 register data from a Modbus slave
108	read 26 Words(4xxxx)	52	0	Use function code 3 to read 26 register data from a Modbus slave
109	read 26 Words(3xxxx)	52	0	Use function code 4 to read 26 register data from a Modbus slave
110	read 28 Words(4xxxx)	56	0	Use function code 3 to read 28 register data from a Modbus slave
111	read 28 Words(3xxxx)	56	0	Use function code 4 to read 28 register data from a Modbus slave
112	read 30 Words(4xxxx)	60	0	Use function code 3 to read 30 register data from a Modbus slave
113	read 30 Words(3xxxx)	60	0	Use function code 4 to read 30 register data from a Modbus slave
114	read 32 Words(4xxxx)	64	0	Use function code 3 to read 32 register data from a Modbus slave
115	read 32 Words(3xxxx)	64	0	Use function code 4 to read 32 register data from a Modbus slave
116	read 34 Words(4xxxx)	68	0	Use function code 3 to read 34 register data from a Modbus slave
117	read 34 Words(3xxxx)	68	0	Use function code 4 to read 34 register data from a Modbus slave
118	read 36 Words(4xxxx)	72	0	Use function code 3 to read 36 register data from a Modbus slave
119	read 36 Words(3xxxx)	72	0	Use function code 4 to read 36 register data from a Modbus slave
120	read 38 Words(4xxxx)	76	0	Use function code 3 to read 38 register data from a Modbus slave
121	read 38 Words(3xxxx)	76	0	Use function code 4 to read 38 register data from a Modbus slave
122	read 40 Words(4xxxx)	80	0	Use function code 3 to read 40 register data from a Modbus slave
123	read 40 Words(3xxxx)	80	0	Use function code 4 to read 40 register data from a Modbus slave
124	read 42 Words(4xxxx)	84	0	Use function code 3 to read 42 register data from a Modbus slave
125	read 42 Words(3xxxx)	84	0	Use function code 4 to read 42 register data from a Modbus slave
126	read 44 Words(4xxxx)	88	0	Use function code 3 to read 44 register data from a Modbus slave

Module No.	Module name	Input data length (bytes)	Output data length (bytes）	Description
127	read 44 Words(3xxxx)	88	0	Use function code 4 to read 44 register data from a Modbus slave
128	read 46 Words(4xxxx)	92	0	Use function code 3 to read 46 register data from a Modbus slave
129	read 46 Words(3xxxx)	92	0	Use function code 4 to read 46 register data from a Modbus slave
130	read 48 Words(4xxxx)	96	0	Use function code 3 to read 48 register data from a Modbus slave
131	read 48 Words(3xxxx)	96	0	Use function code 4 to read 48 register data from a Modbus slave
132	read 50 Words(4xxxx)	100	0	Use function code 3 to read 50 register data from a Modbus slave
133	read 50 Words(3xxxx)	100	0	Use function code 4 to read 50 register data from a Modbus slave
134	read 52 Words(4xxxx)	104	0	Use function code 3 to read 52 register data from a Modbus slave
135	read 52 Words(3xxxx)	104	0	Use function code 4 to read 52 register data from a Modbus slave
136	read 54 Words(4xxxx)	108	0	Use function code 3 to read 54 register data from a Modbus slave
137	read 54 Words(3xxxx)	108	0	Use function code 4 to read 54 register data from a Modbus slave
138	read 56 Words(4xxxx)	112	0	Use function code 3 to read 56 register data from a Modbus slave
139	read 56 Words(3xxxx)	112	0	Use function code 4 to read 56 register data from a Modbus slave
140	read 58 Words(4xxxx)	116	0	Use function code 3 to read 58 register data from a Modbus slave
141	read 58 Words(3xxxx)	116	0	Use function code 4 to read 58 register data from a Modbus slave
142	read 60 Words(4xxxx)	120	0	Use function code 3 to read 60 register data from a Modbus slave
143	read 60 Words(3xxxx)	120	0	Use function code 4 to read 60 register data from a Modbus slave
144	write 8 bits(0xxxx)	0	1	Use function code 15 to write 1~8 coil data to a Modbus slave station
145	write 16 bits(0xxxx)	0	2	Use function code 15 to write 1~16 coil data to a Modbus slave station
146	write 24 bits(0xxxx)	0	3	Use function code 15 to write 1~24 coil data to a Modbus slave station
147	write 32 bits(0xxxx)	0	4	Use function code 15 to write 1~32 coil data to a Modbus slave station
148	write 40 bits(0xxxx)	0	5	Use function code 15 to write 1~40 coil data to a Modbus slave station
149	write 48 bits(0xxxx)	0	6	Use function code 15 to write 1~48 coil data to a Modbus slave station
150	write 56 bits(0xxxx)	0	7	Use function code 15 to write 1~56 coil data to a Modbus slave station
151	write 64 bits(0xxxx)	0	8	Use function code 15 to write 1~64 coil data to a Modbus slave station
152	write 72 bits(0xxxx)	0	9	Use function code 15 to write 1~72 coil data to a Modbus slave station
153	write 80 bits(0xxxx)	0	10	Use function code 15 to write 1~80 coil data to a Modbus slave station
154	write 88 bits(0xxxx)	0	11	Use function code 15 to write 1~88 coil data to a Modbus slave station
155	write 96 bits(0xxxx)	0	12	Use function code 15 to write 1~96 coil data to a Modbus slave station
156	write 104 bits(0xxxx)	0	13	Use function code 15 to write 1~104 coil data to a

Module No.	Module name	Input data length (bytes)	Output data length (bytes）	Description
				Modbus slave station
157	write 112 bits(0xxxx)	0	14	Use function code 15 to write 1~112 coil data to a Modbus slave station
158	write 120 bits(0xxxx)	0	15	Use function code 15 to write 1~120 coil data to a Modbus slave station
159	write 128 bits(0xxxx)	0	16	Use function code 15 to write 1~128 coil data to a Modbus slave station
160	write 136 bits(0xxxx)	0	17	Use function code 15 to write 1~136 coil data to a Modbus slave
161	write 144 bits(0xxxx)	0	18	Use function code 15 to write 1~144 coil data to a Modbus slave station
162	write 152 bits(0xxxx)	0	19	Use function code 15 to write 1~152 coil data to a Modbus slave station
163	write 160 bits(0xxxx)	0	20	Use function code 15 to write 1~160 coil data to a Modbus slave station
164	write 168 bits(0xxxx)	0	21	Use function code 15 to write 1~168 coil data to a Modbus slave station
165	write 176 bits(0xxxx)	0	22	Use function code 15 to write 1~176 coil data to a Modbus slave station
166	write 184 bits(0xxxx)	0	23	Use function code 15 to write 1~184 coil data to a Modbus slave station
167	write 192 bits(0xxxx)	0	24	Use function code 15 to write 1~192 coil data to a Modbus slave station
168	write 200 bits(0xxxx)	0	25	Use function code 15 to write 1~200 coil data to a Modbus slave
169	write 208 bits(0xxxx)	0	26	Use function code 15 to write 1~208 coil data to a Modbus slave station
170	write 216 bits(0xxxx)	0	27	Use function code 15 to write 1~216 coil data to a Modbus slave station
171	write 224 bits(0xxxx)	0	28	Use function code 15 to write 1~224 coil data to a Modbus slave station
172	write 232 bits(0xxxx)	0	29	Use function code 15 to write 1~232 coil data to a Modbus slave station
173	write 240 bits(0xxxx)	0	30	Use function code 15 to write 1~240 coil data to a Modbus slave station
174	write 248 bits(0xxxx)	0	31	Use function code 15 to write 1~248 coil data to a Modbus slave station
175	write 256 bits(0xxxx)	0	32	Use function code 15 to write 1~256 coil data to a Modbus slave
176	write 1 Words(4xxxx)	0	2	Use function code 16 to write 1 register data to a Modbus slave
177	write 2 Words(4xxxx)	0	4	Use function code 16 to write 2 register data to a Modbus slave
178	write 3 Words(4xxxx)	0	6	Use function code 16 to write 3 register data to a Modbus slave
179	write 4 Words(4xxxx)	0	8	Use function code 16 to write 4 register data to a Modbus slave
180	write 5 Words(4xxxx)	0	10	Use function code 16 to write 5 register data to a Modbus slave
181	write 6 Words(4xxxx)	0	12	Use function code 16 to write 6 register data to a Modbus slave
182	write 7 Words(4xxxx)	0	14	Use function code 16 to write 7 register data to a Modbus slave
183	write 8 Words(4xxxx)	0	16	Use function code 16 to write 8 register data to a Modbus slave
184	write 9 Words(4xxxx)	0	18	Use function code 16 to write 9 register data to a Modbus slave
185	write 10 Words(4xxxx)	0	20	Use function code 16 to write 10 register data to a Modbus slave

Module No.	Module name	Input data length (bytes)	Output data length (bytes）	Description
186	write 11 Words(4xxxx)	0	22	Use function code 16 to write 11 register data to a Modbus slave
187	write 12 Words(4xxxx)	0	24	Use function code 16 to write 12 register data to a Modbus slave
188	write 13 Words(4xxxx)	0	26	Use function code 16 to write 13 register data to a Modbus slave
189	write 14 Words(4xxxx)	0	28	Use function code 16 to write 14 register data to a Modbus slave
190	write 15 Words(4xxxx)	0	30	Use function code 16 to write 15 register data to a Modbus slave
191	write 16 Words(4xxxx)	0	32	Use function code 16 to write 16 register data to a Modbus slave
192	write 18 Words(4xxxx)	0	36	Use function code 16 to write 18 register data to a Modbus slave
193	write 20 Words(4xxxx)	0	40	Use function code 16 to write 20 register data to a Modbus slave
194	write 22 Words(4xxxx)	0	44	Use function code 16 to write 22 register data to a Modbus slave
195	write 24 Words(4xxxx)	0	48	Use function code 16 to write 24 register data to a Modbus slave
196	write 26 Words(4xxxx)	0	52	Use function code 16 to write 26 register data to a Modbus slave
197	write 28 Words(4xxxx)	0	56	Use function code 16 to write 28 register data to a Modbus slave
198	write 30 Words(4xxxx)	0	60	Use function code 16 to write 30 register data to a Modbus slave
199	write 32 Words(4xxxx)	0	64	Use function code 16 to write 32 register data to a Modbus slave
200	force single bit (05H Command)	0	1	Use function code 5 to write 1 coil data to a Modbus slave
201	set single word (06H Command)	0	2	Use function code 6 to write 1 register data to a Modbus slave
202	8Bits MODBUS Slaves Status	1	0	Read 8-bit Modbus slave status
203	8Bytes MODBUS Slaves Status	8	0	Read 8-byte Modbus slave status
204	16Bits MODBUS Slaves Status	2	0	Read 16-bit Modbus slave status
205	16Bytes MODBUS Slaves Status	16	0	Read 16-byte Modbus slave status
206	24Bits MODBUS Slaves Status	3	0	Read 24-bit Modbus slave status
207	24Bytes MODBUS Slaves Status	24	0	Read 24-byte Modbus slave status
208	32Bits MODBUS Slaves Status	4	0	Read 32-bit Modbus slave status
209	32Bytes MODBUS Slaves Status	32	0	Read 32-byte Modbus slave status

As shown in the above table, the above 209 modules can be roughly divided into 4 categories：

a) Empty module (module 1)

b) Modbus communication module (module 4-201)

c) Control module (module 3)

d) General status module (module 2) and detailed status module (202-209)

Among them, slot 1 is fixed as the overall state module (module 2), slot 2 is fixed as the control module (module 3), and the remaining 37 slots can be configured as required.

Note: If there is a detailed status module requirement (modules 202-209), the detailed status module can only be placed in the last valid slot (for example: Modbus communication module uses 5 slots, then slot 1 is the overall status module, and slot 2 is the control Module, slots 3-7 are Modbus communication modules, slot 8 is

the last valid slot in this example, if you need a detailed status module, put it in slot 8).

Modbus communication related configuration is realized through user parameters, which are named device user parameters. Except for the first three modules, all other modules have user parameters that can be configured. This part of the parameters is named module user parameters.

- **Equipment user parameters**

This part of the parameters includes the Modbus communication baud rate, check information, data update mode, write mode, master sending interval, interval time, and whether there is Modbus slave status detection and other information.

Table 4 G0306-MS device user parameter table

No.	Parameter name	Description
1	Baud rate	0:300 bps 1:600 bps 2:1200 bps 3:2400 bps 4:4800 bps 5:9600 bps 6:19200 bps 7:38400 bps 8:57600 bps
2	Parity	0: 8Bits, No Parity, 1stop bit 1: 8Bits, Even Parity, 1stop bit 2: 8Bits, Odd Parity, 1stop bit 3: 8Bits, No Parity, 2stop bits
3	MODBUS Slave Monitoring	When this parameter is non-zero, the corresponding detailed status module must be used（202-209） 0: No Monitoring 1: 8Bits Monitoring 2: 8Bytes Monitoring 3: 16Bits Monitoring 4: 16Bytes Monitoring 5: 24Bits Monitoring 6: 24Bytes Monitoring 7: 32Bits Monitoring 8: 32Bytes Monitoring
4	Data update mode	1: All Items End 2: Each Item End
5	Writing mode	0: Write always 1: Write on change
6	Master sending interval	0: Salve Response If the slave does not respond in this way, the response timeout time is fixed at 1 second. 1: Same Interval The sending interval and response timeout time in this way are both the time set by the interval time parameter.
7	Interval time value	1:10ms 2:20ms 3:30ms 4:40ms 5:50ms 6:60ms 8:80ms 10:100ms 12:120ms 15:150ms 20:200ms 25:250ms 30:300ms

		35:350ms 40:400ms 45:450ms 50:500ms 55:550ms 60:600ms 65:650ms 70:700ms 75:750ms 80:800ms 85:850ms 90:900ms 95:950ms 100:1000ms 255: Waiting...

- **Module user parameters**

This part of the parameters is related to the actual Modbus slave device, including the Modbus slave device address, the starting address of the Modbus register to be read and written, and the number of output data to be written.

Table 5 User parameter list of G0306-MS module (4-143, 176-199)

No.	Parameter name	Description
1	MODBUS Slave Address	Modbus slave address, the address range is 0-255.
2	Starting Address	The starting address of the register to read and write data.

Table 6 G0306-MS module (144-175) user parameter list

No.	Parameter name	Description
1	MODBUS Slave Address	Modbus slave address, the address range is 0-255.
2	Starting Address	The starting address of the register to read and write data.
3	Quantity of Outputs	Number of output coils.

Table 7 G0306-MS module 200 user parameter table

No.	Parameter name	Description
1	MODBUS Slave Address	Modbus slave address, the address range is 0-255.
2	Output Address	The output address of the coil to which the data is written.

Table 8 G0306-MS module 201 user parameter table

No.	Parameter name	Description
1	MODBUS Slave Address	Modbus slave address, the address range is 0-255.
2	Register Address	The address of the register where the data is written.

Note: Table 4-7 only lists the user parameters that can be modified by these modules, not all the parameters of these modules. For example: the Modbus function code corresponding to a module, the number of registers for reading and writing data, the number of bytes, etc., which are related to the module and cannot be modified, are not listed in the above table.

Table 9 G0306-MS detailed status module (202-203) user parameter table

Parameter length	Parameter default value	Description
8 bytes	0x01,0x02,0x03,0x04,0x05,0x06,0x07,0x08	Each byte represents a Modbus slave address to be tested.

Table 10 G0306-MS detailed status module (204-205) user parameter table

Parameter length	Parameter default value	Description
16 bytes	0x01,0x02,0x03,0x04,0x05,0x06,0x07,0x08, 0x09,0x0a,0x0b,0x0c,0x0d,0x0e,0x0f,0x10	Each byte represents a Modbus slave address to be tested.

Table 11 G0306-MS detailed status module (206-207) user parameter table

Parameter length	Parameter default value	Description
24 bytes	0x01,0x02,0x03,0x04,0x05,0x06,0x07,0x08, 0x09,0x0a,0x0b,0x0c,0x0d,0x0e,0x0f,0x10, 0x11,0x12,0x13,0x14,0x15,0x16,0x17,0x18	Each byte represents a Modbus slave address to be tested.

Table 12 G0306-MS detailed status module (208-209) user parameter table

Parameter length	Parameter default value	Description
32 bytes	0x01,0x02,0x03,0x04,0x05,0x06,0x07,0x08, 0x09,0x0a,0x0b,0x0c,0x0d,0x0e,0x0f,0x10, 0x11,0x12,0x13,0x14,0x15,0x16,0x17,0x18, 0x19,0x1a,0x1b,0x1c,0x1d,0x1e,0x1f,0x20	Each byte represents a Modbus slave address to be tested.

GSD file description of G0306-SS

The GSD file name for G0306-SS is：MCYB0F19.GSE

This GSD file contains 20 slots, 67 modules, and supports up to 237 user parameters.

- **Module description**

Table 13 GSD module description of G0306-SS

Module No.	Module name	Input data length (bytes)	Output data length (bytes)	Description
1	empty	0	0	Empty module
2	status	1	0	Modbus Communication status module
3	control	0	1	Modbus Communication status module
4	Input:8 bits(0xxxx)	1	0	Map 1 byte of input data to Modbus 0xxxx storage area in order
5	Input:16 bits(0xxxx)	2	0	Map 2 byte of input data to Modbus 0xxxx storage area in order
6	Input:24 bits(0xxxx)	3	0	Map 3 byte of input data to Modbus 0xxxx storage area in order
7	Input:32 bits(0xxxx)	4	0	Map 4 byte of input data to Modbus 0xxxx storage area in order
8	Input:40 bits(0xxxx)	5	0	Map 5 byte of input data to Modbus 0xxxx storage area in order
9	Input:48 bits(0xxxx)	6	0	Map 6 byte of input data to Modbus 0xxxx storage area in order
10	Input:56 bits(0xxxx)	7	0	Map 7 byte of input data to Modbus 0xxxx storage area in order
11	Input:64 bits(0xxxx)	8	0	Map 8 byte of input data to Modbus 0xxxx storage area in order
12	Input:72 bits(0xxxx)	9	0	Map 9 byte of input data to Modbus 0xxxx storage area in order
13	Input:80 bits(0xxxx)	10	0	Map 10 byte of input data to Modbus 0xxxx storage area in order
14	Input:88 bits(0xxxx)	11	0	Map 11 byte of input data to Modbus 0xxxx storage area in order
15	Input:96 bits(0xxxx)	12	0	Map 12 byte of input data to Modbus 0xxxx storage area in order
16	Input:104 bits(0xxxx)	13	0	Map 13 byte of input data to Modbus 0xxxx storage area in order
17	Input:112 bits(0xxxx)	14	0	Map 14 byte of input data to Modbus 0xxxx storage area in order
18	Input:120 bits(0xxxx)	15	0	Map 15 byte of input data to Modbus 0xxxx storage area in order
19	Input:128 bits(0xxxx)	16	0	Map 1 byte of input data to Modbus 0xxxx storage area in order
20	Input:1 Word(4xxxx)	2	0	Map 2 bytes of input data to Modbus 4xxxx storage area in sequence

Module No.	Module name	Input data length (bytes)	Output data length (bytes)	Description
21	Input:2 Words(4xxxx)	4	0	Map 4 bytes of input data to Modbus 4xxxx storage area in sequence
22	Input:3 Words(4xxxx)	6	0	Map 6 bytes of input data to Modbus 4xxxx storage area in sequence
23	Input:4 Words(4xxxx)	8	0	Map 8 bytes of input data to Modbus 4xxxx storage area in sequence
24	Input:5 Words(4xxxx)	10	0	Map 10 bytes of input data to Modbus 4xxxx storage area in sequence
25	Input:6 Words(4xxxx)	12	0	Map 12 bytes of input data to Modbus 4xxxx storage area in sequence
26	Input:7 Words(4xxxx)	14	0	Map 14 bytes of input data to Modbus 4xxxx storage area in sequence
27	Input:8 Words(4xxxx)	16	0	Map 17 bytes of input data to Modbus 4xxxx storage area in sequence
28	Input:9 Words(4xxxx)	18	0	Map 18 bytes of input data to Modbus 4xxxx storage area in sequence
29	Input:10 Words(4xxxx)	20	0	Map 20 bytes of input data to Modbus 4xxxx storage area in sequence
30	Input:11 Words(4xxxx)	22	0	Map 22 bytes of input data to Modbus 4xxxx storage area in sequence
31	Input:12 Words(4xxxx)	24	0	Map 24 bytes of input data to Modbus 4xxxx storage area in sequence
32	Input:13 Words(4xxxx)	26	0	Map 26 bytes of input data to Modbus 4xxxx storage area in sequence
33	Input:14 Words(4xxxx)	28	0	Map 28 bytes of input data to Modbus 4xxxx storage area in sequence
34	Input:15 Words(4xxxx)	30	0	Map 30 bytes of input data to Modbus 4xxxx storage area in sequence
35	Input:16 Words(4xxxx)	32	0	Map 32 bytes of input data to Modbus 4xxxx storage area in sequence
36	Output:8 bits(1xxxx)	0	1	Map 1 byte of output data to Modbus 1xxxx storage area in order
37	Output:16 bits(1xxxx)	0	2	Map 2 byte of output data to Modbus 1xxxx storage area in order
38	Output:24 bits(1xxxx)	0	3	Map 3 byte of output data to Modbus 1xxxx storage area in order
39	Output:32 bits(1xxxx)	0	4	Map 4 byte of output data to Modbus 1xxxx storage area in order
40	Output:40 bits(1xxxx)	0	5	Map 5 byte of output data to Modbus 1xxxx storage area in order
41	Output:48 bits(1xxxx)	0	6	Map 6 byte of output data to Modbus 1xxxx storage area in order
42	Output:56 bits(1xxxx)	0	7	Map 7 byte of output data to Modbus 1xxxx storage area in order
43	Output:64 bits(1xxxx)	0	8	Map 8 byte of output data to Modbus 1xxxx storage area in order
44	Output:72 bits(1xxxx)	0	9	Map 9 byte of output data to Modbus 1xxxx storage area in order
45	Output:80 bits(1xxxx)	0	10	Map 10 byte of output data to Modbus 1xxxx storage area in order
46	Output:88 bits(1xxxx)	0	11	Map 11 byte of output data to Modbus 1xxxx storage area in order
47	Output:96 bits(1xxxx)	0	12	Map 12 byte of output data to Modbus 1xxxx storage area in order
48	Output:104 bits(1xxxx)	0	13	Map 13 byte of output data to Modbus 1xxxx storage area in order
49	Output:112 bits(1xxxx)	0	14	Map 14 byte of output data to Modbus 1xxxx storage area in order
50	Output:120 bits(1xxxx)	0	15	Map 15 byte of output data to Modbus 1xxxx storage area in

Module No.	Module name	Input data length (bytes)	Output data length (bytes)	Description
				order
51	Output:128 bits(1xxxx)	0	16	Map 16 byte of output data to Modbus 1xxxx storage area in order
52	Output:1 Word(3xxxx)	0	2	Map 2 bytes of output data to the 3xxxx storage area of Modbus in order
53	Output:2 Words(3xxxx)	0	4	Map 4 bytes of output data to the 3xxxx storage area of Modbus in order
54	Output:3 Words(3xxxx)	0	6	Map 6 bytes of output data to the 3xxxx storage area of Modbus in order
55	Output:4 Words(3xxxx)	0	8	Map 8 bytes of output data to the 3xxxx storage area of Modbus in order
56	Output:5 Words(3xxxx)	0	10	Map 10 bytes of output data to the 3xxxx storage area of Modbus in order
57	Output:6 Words(3xxxx)	0	12	Map 12 bytes of output data to the 3xxxx storage area of Modbus in order
58	Output:7 Words(3xxxx)	0	14	Map 14 bytes of output data to the 3xxxx storage area of Modbus in order
59	Output:8 Words(3xxxx)	0	16	Map 16 bytes of output data to the 3xxxx storage area of Modbus in order
60	Output:9 Words(3xxxx)	0	18	Map 18 bytes of output data to the 3xxxx storage area of Modbus in order
61	Output:10 Words(3xxxx)	0	20	Map 20 bytes of output data to the 3xxxx storage area of Modbus in order
62	Output:11 Words(3xxxx)	0	22	Map 22 bytes of output data to the 3xxxx storage area of Modbus in order
63	Output:12 Words(3xxxx)	0	24	Map 24 bytes of output data to the 3xxxx storage area of Modbus in order
64	Output:13 Words(3xxxx)	0	26	Map 26 bytes of output data to the 3xxxx storage area of Modbus in order
65	Output:14 Words(3xxxx)	0	28	Map 28 bytes of output data to the 3xxxx storage area of Modbus in order
66	Output:15 Words(3xxxx)	0	30	Map 30 bytes of output data to the 3xxxx storage area of Modbus in order
67	Output:16 Words(3xxxx)	0	32	Map 32 bytes of output data to the 3xxxx storage area of Modbus in order

As shown in the above table, the above 67 modules can be roughly divided into 4 categories：

a) Empty module (module 1)

b) Overall status module (module 2)

c) Control module (module 3)

d) d) Modbus mapping module (module 4-67), slot 1 is fixed as the overall status module (module 2), slot 2 is fixed as the control module (module 3), and the remaining 18 slots can be configured as required.

Modbus communication related configuration is realized through user parameters, which are named device user parameters. Except for the first three modules, all other modules have user parameters that can be configured. This part of the parameters is named module user parameters.

- **Equipment user parameters**

This part of the parameters includes the Modbus communication baud rate, verification information, and whether there is Modbus slave status detection and other information.

Table 14 G0306-SS device user parameter table

No.	Parameter name	Description
1	Baudrate	3:2400 bps 4:4800 bps

		5:9600 bps 6:19200 bps 7:38400 bps 8:57600 bps
2	Parity	0: 8Bits, No Parity, 1stop bit 1: 8Bits, Even Parity, 1stop bit 2: 8Bits, Odd Parity, 1stop bit 3: 8Bits, No Parity, 2stop bits
3	MODBUS Slave Device Address	This parameter is used to set the current gateway as the address of the MODBUS slave device. Range: 1-247.

- **Module user parameters**

This part of the parameters is related to Modbus storage area and data length. The parameter value is associated with the module and cannot be modified. For example: "Input: 24 bits(0xxxx)" module user parameters are 0x00, 0x03. Indicates that this module is associated with Modbus 0xxxx storage area, and the data length is 3 bytes.

- **Correspondence between Modbus storage area and Profibus input and output data buffer**

Table 15 Correspondence between Modbus storage area and Profibus input and output data buffer

Module	Corresponding Modbus storage area	Maximum length of input and output data	Storage unit address range
Input:8 bits(0xxxx) ~ Input:128 bits(0xxxx)	Coil 0xxxx	243 byte =1944bit	0~1943
Output:8 bits(1xxxx) ~ Output:128 bits(1xxxx)	Discrete input 1xxxx	243 byte =1944bit	0~1943
Input:1 Word(4xxxx) ~ Input:16 Words(4xxxx)	Holding register 4xxxx	242 byte =121word	0~120
Output:1 Word(3xxxx) ~ Output:16 Words(3xxxx)	Input register 3xxxx	242 byte =121word	0~120

4.3.2. Install GSD file

Take Siemens STEP 7 software and G0306-MS as examples, select any project, open the hardware configuration interface, select "Options Install GSD File..." option, and the import GSD file window will open.

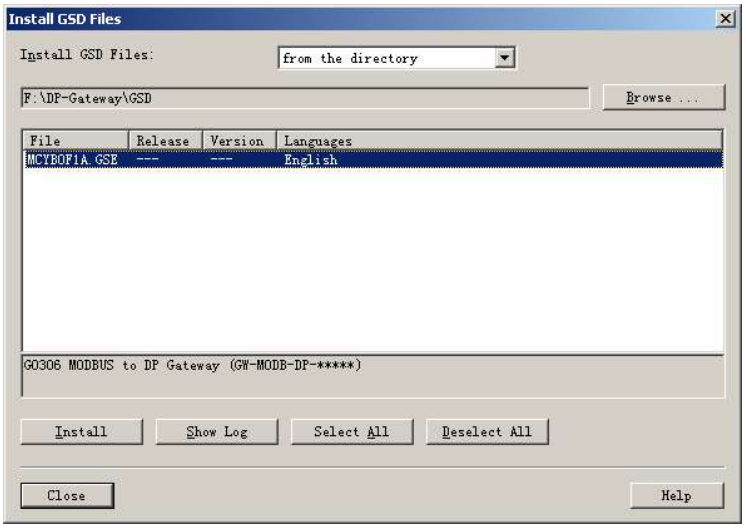


Figure 9 Import GSD file window

Click "Browse..." and select the path where the GSD file is located. All GSD files in the current path will be listed. Select the GSD file to be imported, and then click "Install". Keep clicking "Yes" until Figure 10 appears.

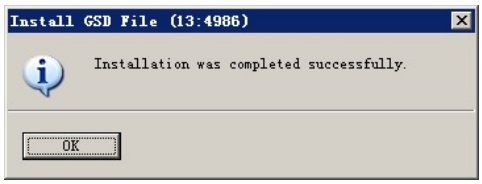


Figure 10 Imported successfully

G0306-SS has the same method of importing GSD files.

4.3.4. Using GSD files

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

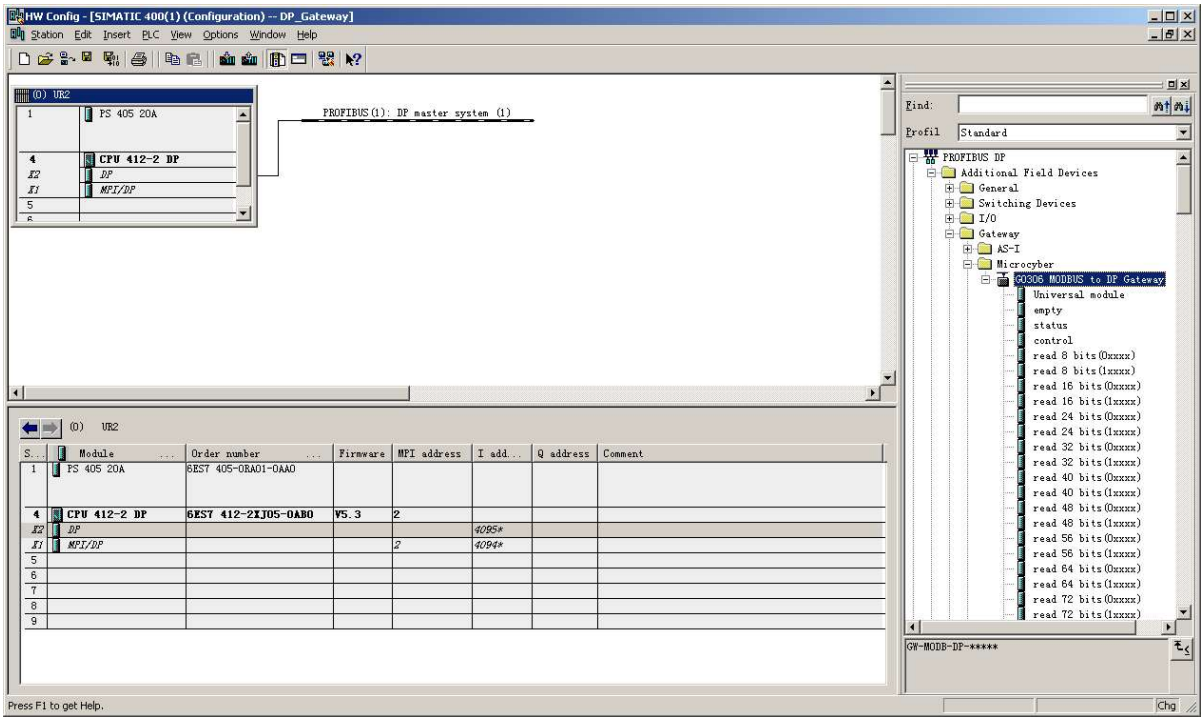


Figure 11 Properly installed equipment

Drag the gateway to the DP bus. The properties window will automatically pop up. Configure the gateway address to the address you need. Here I use address 10.

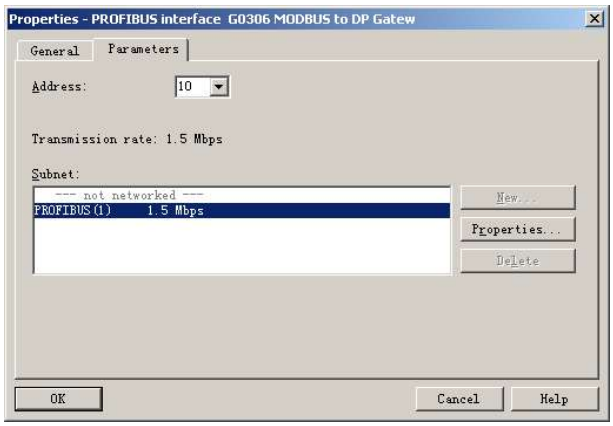


Figure 12 Configure device properties

Click "OK" to finish adding the gateway.

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

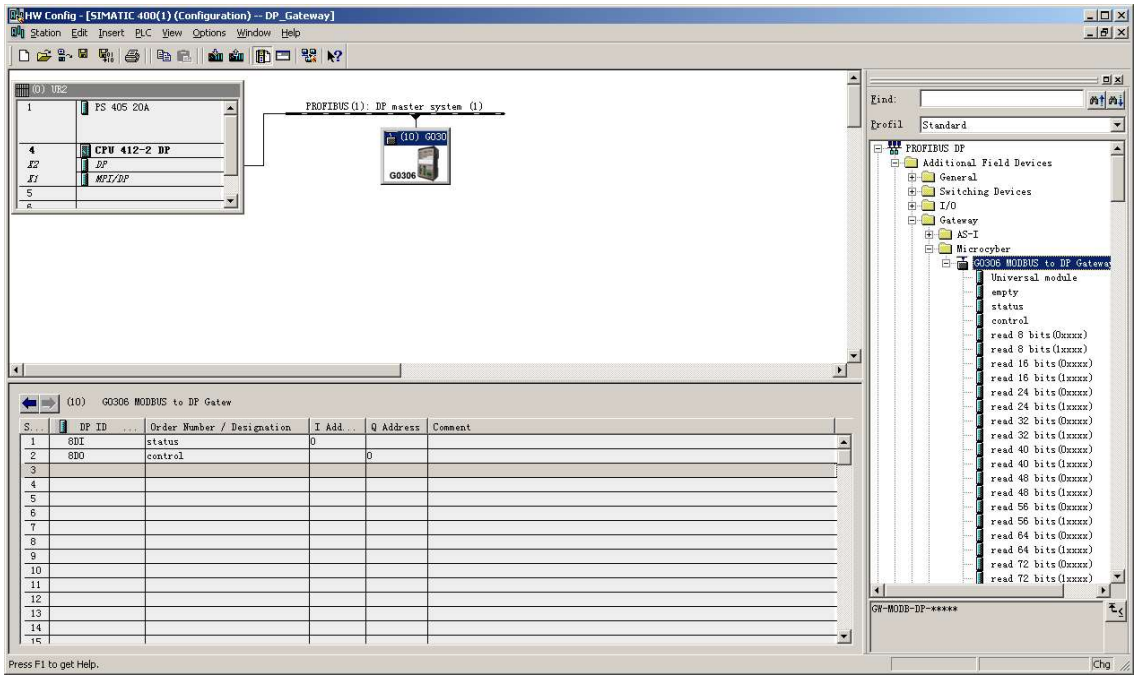


Figure 13 Device configuration configuration

When performing hardware configuration, the user makes corresponding configuration adjustments according to actual needs, so that the input and output data of the gateway and Modbus data form effective configuration information.

The G0306-SS GSD file import method is the same as this.

The following will illustrate how to use the gateway.

(1) How to use the GSD file of G0306-MS

- **Configure device user parameters**

In the interface shown in Figure 13, right-click on the gateway device and select Object Properties Parameter Assignment

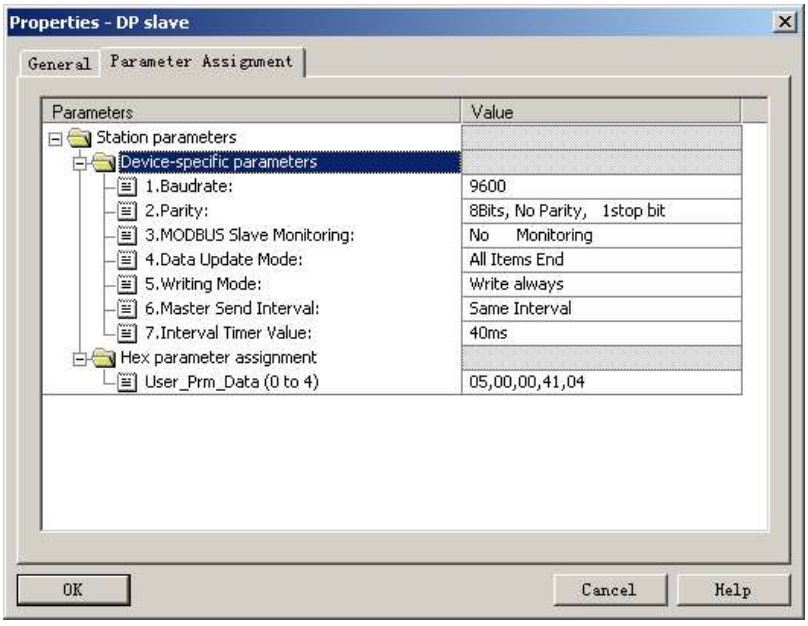


Figure 14 G0306-MSDevice user parameter configuration interface

Here you can modify all the parameters mentioned in Table 4. The user should make modifications according

to the actual situation. For example: The interval time value is related to the selected baud rate, the response time of the actually connected slave device, and the number of connected devices. If the configuration is not appropriate, it may generate a connection packet, resulting in the slave station not responding or responding incorrectly.

- **Introduction to Configure Gateway Module**

As mentioned in Chapter 4.3.1 (1), this gateway GSD contains 39 slots, 209 modules, and supports up to 237 user parameters. The specific meaning of each module option is shown in Table 3.

Among them, slot 1 is fixed as the overall state module (module 2), slot 2 is fixed as the control module (module 3), and the remaining 37 slots can be configured as required.

Take the configuration "read 24 bits (0xxxx)" module (module 8) into slot 3 as an example, select slot 3, in the device list on the right, double-click the "read 24 bits (0xxxx)" module to add the module to the slot 3 in. The module also has module user parameters configurable, the detailed configuration method is shown below.

The gateway forms a Modbus message queue according to the module sequence actually configured by the user. How to use each module will be described in detail below.

- **Overall status module (module 2)**

This module displays the real-time status of each Modbus message according to the Modbus message queue cycle.

Table 16 Overall status module format

B7: Parity check	B6:CRC check	B5: Response timeout	B4-B1: Exception response code	B0: Send/receive
0: Current slave parity check is correct	0: Current slave CRC is correct	0: Current slave response has not timed out	See Appendix A.3 for details	0:Send
1: Current slave parity error	1: Current slave CRC error	1: Current slave response timeout		1:Receive

a) **B0: Send/receive**

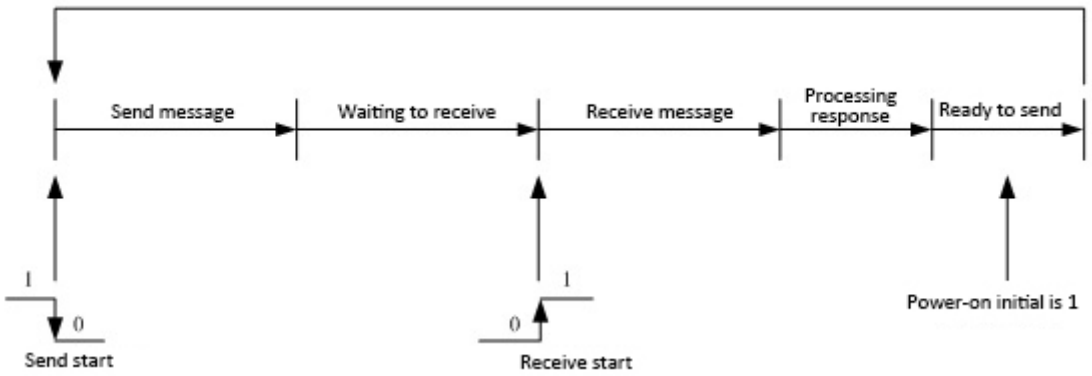


Figure 15 Transmitting and receiving state transition diagram

As shown in the figure above, this bit is in 1 state after the device is powered on. With the normal operation of the Modbus message queue, this bit will change in real time as shown in the figure above. Set to 0 when sending and 1 when receiving.

b) B4-B1: Exception response code

These 4 bits are used to display the abnormal response code of the current Modbus message. The specific meaning of the abnormal response code is detailed in Appendix A.3.

Note: There is also a special usage of these 4 bits, that is, if the slave command configured for a slot is not configured to detect this slave in the detailed status module (202-209), the abnormal response code will be set in the general status module Set to F.

c) B5: Response timeout

This bit indicates that according to the settings of the master station's sending interval parameter and interval time value parameter in the device user parameters, if the user device does not respond after timeout, this bit is set to 1. The Modbus message queue pointer points to the next Modbus message.

d) B6: CRC check

When the gateway receives a MODBUS response message and the CRC check error occurs, this bit is set to 1. At this time, the gateway considers that the MODBUS response data is unreliable, and it is discarded and not exchanged with the corresponding data area of PROFIBUS.

e) B7: Parity check

When the gateway receives data and finds a parity error, this bit is set to 1. At this time, the gateway considers that the MODBUS response data is unreliable, and it is discarded and not exchanged with the corresponding data area of PROFIBUS.

- **Control module (module 3)**

This module is used to schedule Modbus messages. Users can start or stop Modbus scanning through this module, adjust read-only or write-only, skip the current Modbus message and other operations.

Table 17 Control module format

B7: Forced reset	B6: Stop waiting	B5: Error detection	B4-B3: Keep	B2: Write command	B1: Read command	B0: Stop/start
0: Normal execution	0: Normal execution	0: Enable error detection	——	0: Allowed to write	0: Allowed to read	0: Start Modbus scan
1: Reset	1: Skip waiting	1: Prohibit error detection	——	1: No write	1: Prohibit reading	1: Stop Modbus scan

a) B0: **Stop/start**

The default is to start scanning the Modbus message queue. The user can set this bit to 1, that is, stop Modbus message scanning.

b) **B1: Read command**

The default is to allow the gateway to send Modbus read commands. The user can set this bit to 1, that is, prohibit sending Modbus read commands (01H, 02H, 03H, 04H commands).

c) B2: Write command

The default is to allow the gateway to send Modbus write commands. The user can set this bit to 1, that is, prohibit sending Modbus write commands (05H, 06H, 0FH, 10H commands).

d) **B5: Error detection**

The default is to enable error detection. The user can set this bit to 1, which means that error detection is prohibited, and can be used to clear previous error information.

e) **B6: Stop waiting**

When the sending interval parameter of the master station in the device user parameters is set to send at the same time interval and the interval time value is set to wait for a response indefinitely, this bit configuration is valid. The user can set this bit to 1, skip the current waiting, and scan the next Modbus message in the Modbus message queue.

f) B7: Forced reset

Set this bit to forcibly reset the Modbus message queue to scan the first Modbus message.

- **Examples of using modules that read xxx bits (module 4-67)**

These modules can use function code 1 (or 2) to read any number of coil data.

Take the "read 24 bits (0xxxx)" module as an example, this module uses function code 1 to read 24-bit coil data.

a) Add the "read 24 bits(0xxxx)" module, as shown in Figure 16:

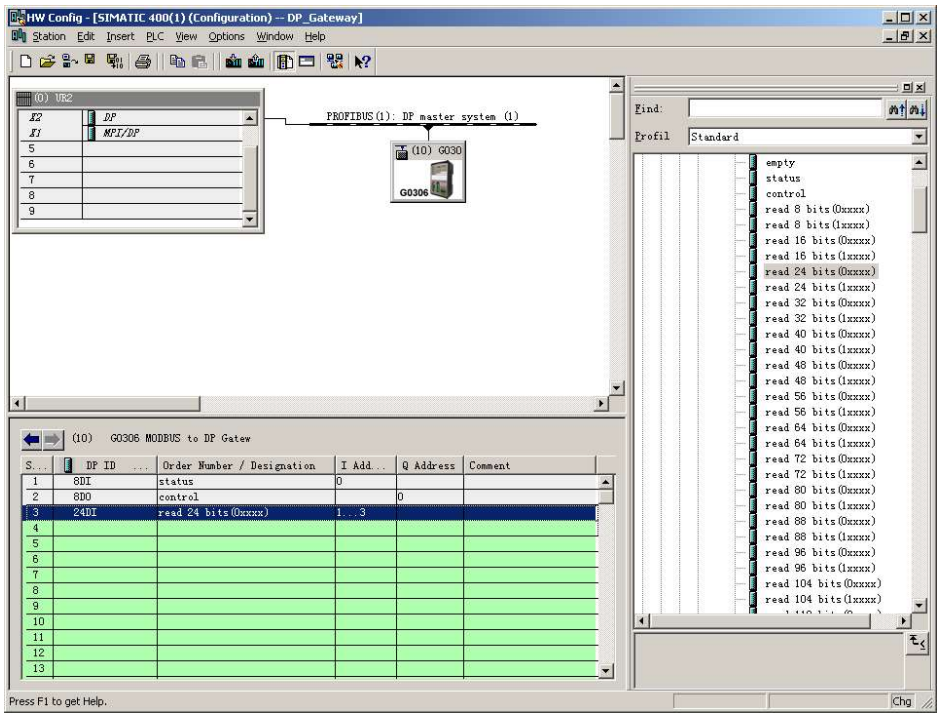


Figure 16 Add“read 24 bits(0xxxx)”module

Take this module into slot 3 as an example, select slot 3 and double-click the "read 24 bits(0xxxx)" module. IB1…3 is the input data address assigned by the Profibus master to the gateway, corresponding to the 24-bit coil (0xxxx) data read by Modbus.

b) Configure the user parameters of the "read 24 bits(0xxxx)" module, as shown in Figure 17：

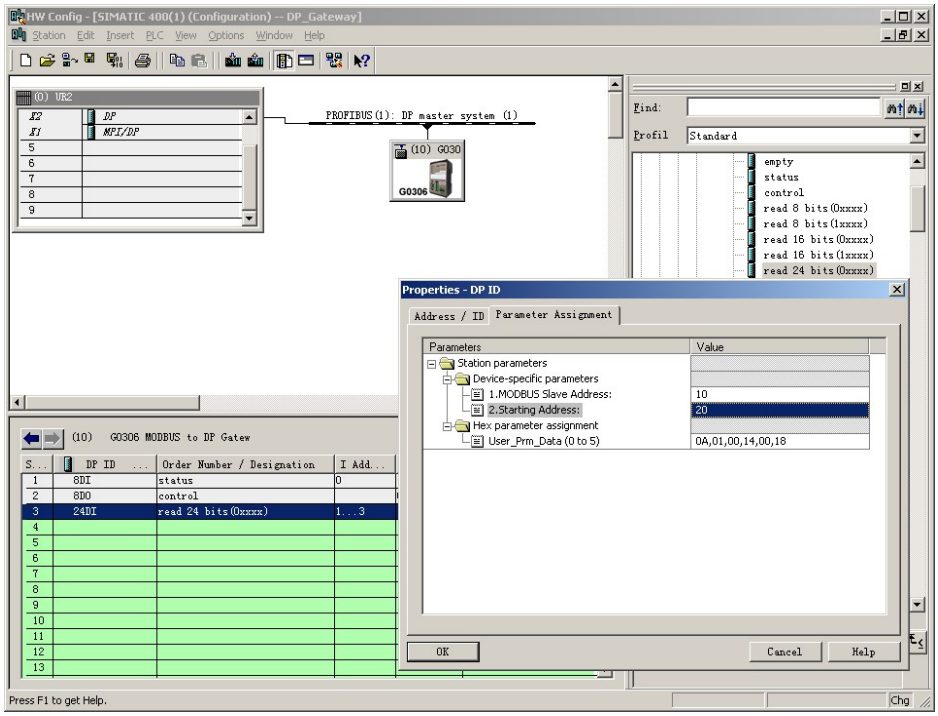


Figure 17 Configure "read 24 bits (0xxxx)" module user parameters

Double-click "24 DI" or "read 24 bits(0xxxx)" or "1…3" in slot 3; select "Parameter Assigement" to complete the parameters of the slave address (MODBUS Slave Address) and the starting address (Starting Address) set up.

Slave address: refers to the address sent by the Modbus communication module to the Modbus slave, corresponding to the first byte of the MODBUS message.

Starting address: Refers to the starting address of 0xxxx to be read. Note: The coil start address 00000 in the message corresponds to the address 00001 in the device, and others are postponed.

“1.MODBUS Slave Address:”⇒ Type MODBUS slave address 10, as shown in Figure 17.

“2.Starting Address:”⇒ Type in the starting address 00021 of the coil 0xxxx to be read, and set the address to 20 "OK", as shown in Figure 17.

c) Correspondence between PROFIBUS address and Modbus address

As shown in Figure 18, IB1..IB3 is the PROFIBUS input data address assigned to this Modbus module by the PROFIBUS master, corresponding to the 24 bits (0xxxx) read in this MODBUS message.

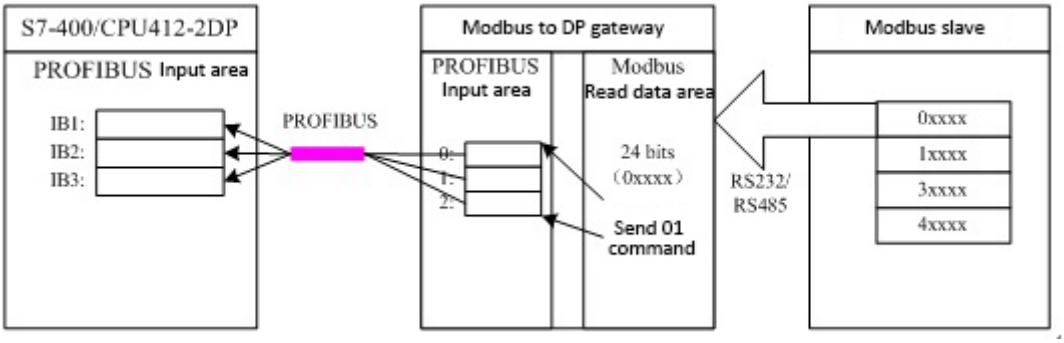


Figure 18 Correspondence between PROFIBUS address and Modbus address

- **Examples of using read xxx word module (module 68-143)**

These modules can use function code 3 (or 4) to read any word of register data.

Take the "read 4 Words(3xxxx)" module as an example, this module uses function code 3 to read 4 words of register data.

a) Add the "read 4 Words(3xxxx)" module, as shown in Figure 19：

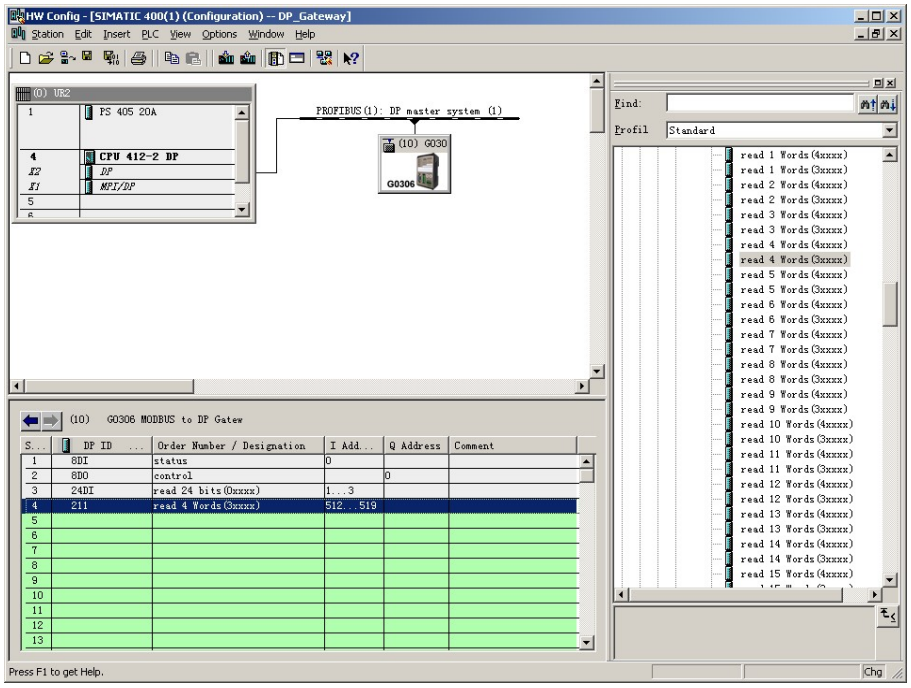


Figure 19 Add“read 4 Words(3xxxx)”module

Take this module into slot 4 as an example, select slot 4 and double-click the "read 4 Words(3xxxx)" module. IB512…519 is the input data address assigned by the Profibus master station to the gateway, corresponding to the 4-word register (3xxxx) data read by Modbus.

b) Configure the user parameters of the "read 4 Words(3xxxx)" module, as shown in Figure 20：

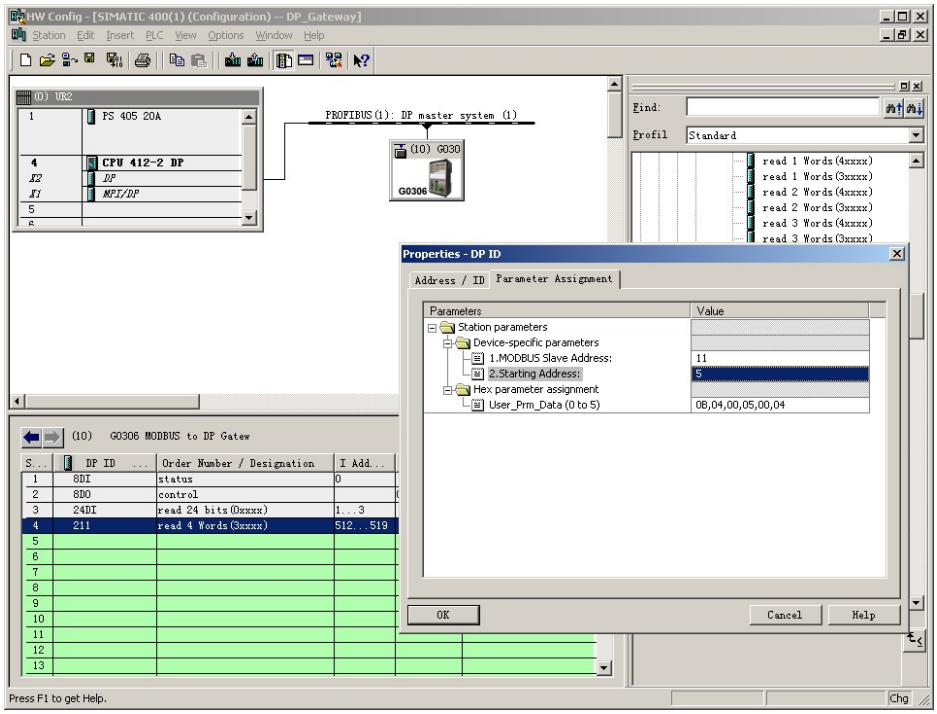


Figure 20 Configure "read 4 Words(3xxxx)" module user parameters

Double-click "211" or "read 4 Words(3xxxx)" or "512…519" in slot 4; select "Parameter Assigement" to complete the parameter setting of the slave address (MODBUS Slave Address) and the starting address (Starting Address) set.

Slave address: refers to the address sent by the Modbus communication module to the Modbus slave, corresponding to the first byte of the MODBUS message.

Start address: Refers to the 3xxxx start address to be read. Note: The start address of the register 30000 in the message corresponds to the address 30001 in the device, and others are postponed.

"1.MODBUS Slave Address:"⇒ Type in the MODBUS slave address 11, as shown in Figure 20

"2.Starting Address:"⇒ Type in the starting address 30006 of the register 3xxxx to be read, and set the address to 5 "OK", as shown in Figure 20.

c) Correspondence between PROFIBUS address and Modbus address

As shown in Figure 21, IB512...IB519 is the PROFIBUS input data address assigned to this Modbus module by the PROFIBUS master, corresponding to the 4 Words (3xxxx) read in this MODBUS message.

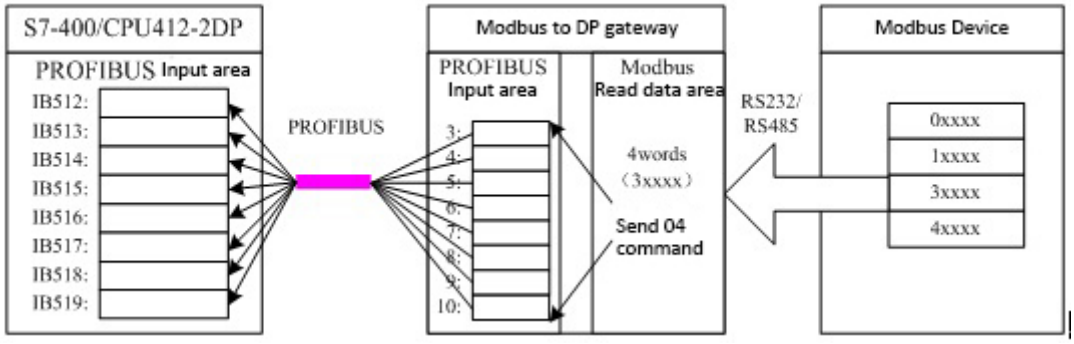


Figure 21 Correspondence between PROFIBUS address and Modbus address

- **Examples of using write xxx bit modules (modules 144-175)**

These modules can use function code 15 to write data to and from any number of coils.

Take the "write 16 bits(0xxxx)" module as an example, this module uses function code 15 to write data to and from the 16-bit coil.

a) Add the "write 16 bits(0xxxx)" module, as shown in Figure 22：

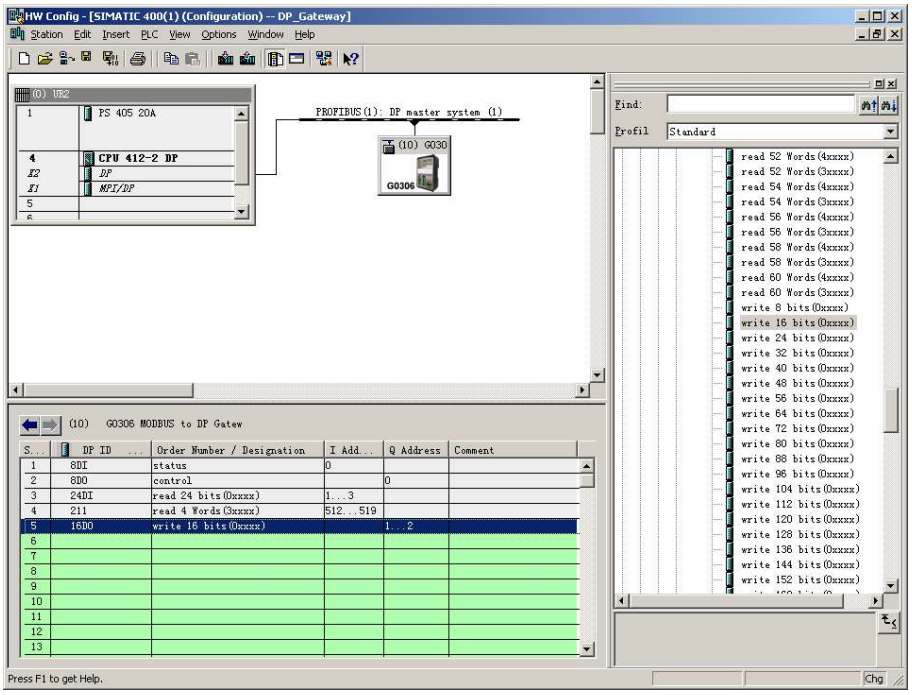


Figure 22 Add“write 16 bits(0xxxx)”module

Take this module into slot 5 as an example, select slot 5 and double-click the "write 16 bits(0xxxx)" module. QB1...2 is the output data address assigned by the Profibus master station to the gateway, corresponding to the 16-bit (0xxxx) data written by Modbus.

b) Configure the user parameters of the "write 16 bits(0xxxx)" module, as shown in Figure 23：

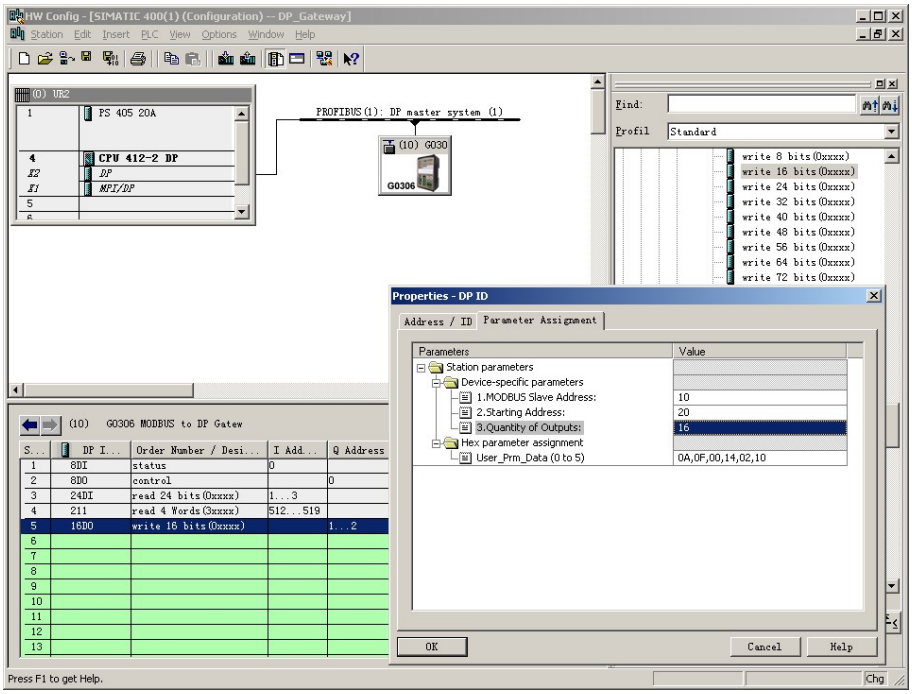


Figure 23 Configure "write 16 bits(0xxxx)" module user parameters

Double-click "16DO" or "write 16 bits(0xxxx)" or "1...2" in slot 5; select "Parameter Assigement" to complete the slave address (MODBUS Slave Address), starting address (Starting Address) and the number of coils (Quantity of Outputs) parameter setting.

Slave address: refers to the address sent by the Modbus communication module to the Modbus slave, corresponding to the first byte of the MODBUS message.

Start address: refers to the 0xxxx start address to be written. Note: The coil start address 00000 in the message corresponds to the address 00001 in the device, and others are postponed.

Number of coils: refers to the number of bits in 0xxxx to be written into this Modbus messag

"1.MODBUS Slave Address:"⇒ Type in Modbus slave address 10, as shown in Figure 23

"2.Starting Address:" ⇒ Type in the starting address 00021 of the coil 0xxxx to be written, and set the address to 20, as shown in Figure 23

"3. Quantity of Outputs:" ⇒ Enter the number of coils to be written 16 ⇒"OK"， as shown in Figure 23

C) Correspondence between PROFIBUS address and Modbus address

As shown in Figure 24, QB1 and QB2 are the PROFIBUS output addresses assigned to this Modbus module by the PROFIBUS master, a total of 2 bytes, corresponding to the 16 continuous coils written into the Modbus device by the Modbus module, the Modbus module will be the PROFIBUS master The 2 bytes (16 bits) of QB1 and QB2 are written into the 0xxxx data area of the Modbus device. The starting address is 00020 in this example; that is, QB1 and QB2 of PRODIBUS are written into 00021~00036 of the Modbus device.

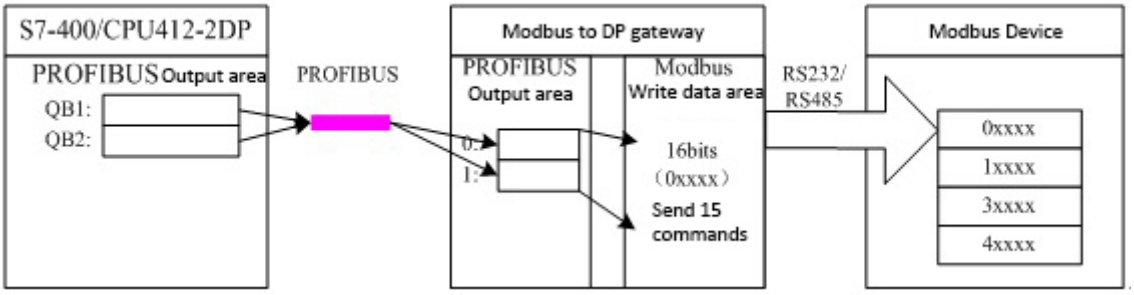


Figure 24 Correspondence between PROFIBUS address and Modbus address

- **Examples of using write xxx word module (module 176-199)**

These modules can use function code 16 to write data to and from any register.

Take the "write 4 Words(4xxxx)" module as an example, this module uses function code 16 to write data to and from 4 registers.

a) Add“write 4 Words(4xxxx)”Module，as shown in Figure 25：

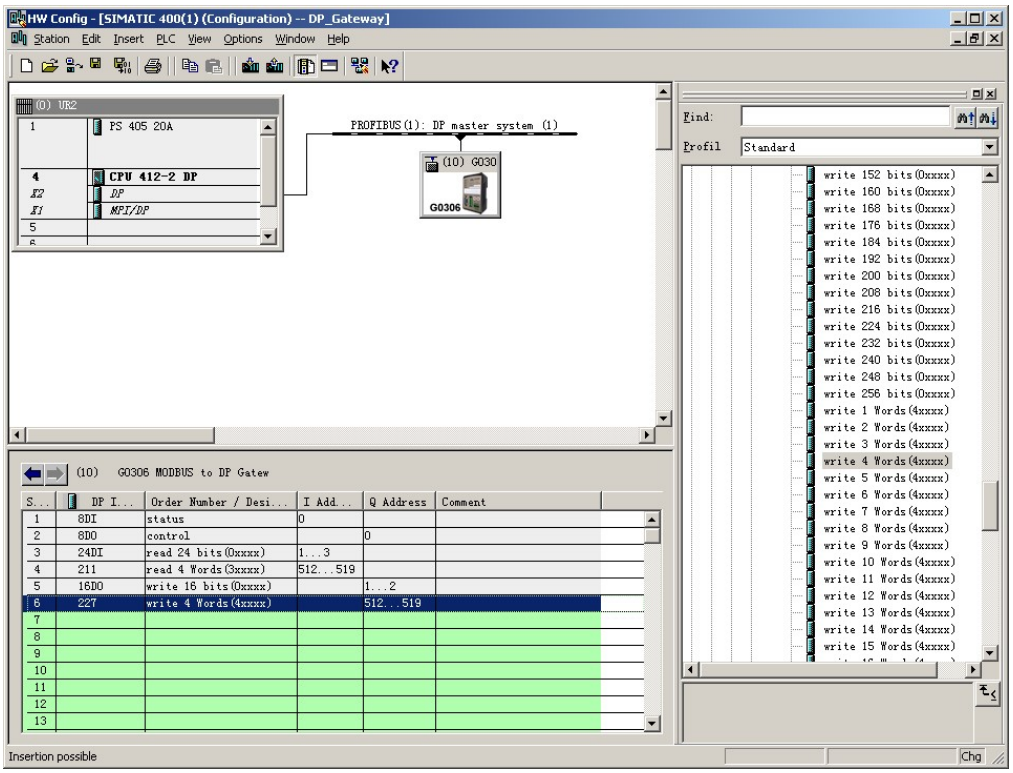


Figure 25 Add“write 16 bits(0xxxx)”module

Take this module into slot 6 as an example, select slot 6 and double-click the "write 4 Words(4xxxx)" module. QB512…519 is the output data address assigned by the Profibus master station to the gateway, corresponding to the 4-word register (4xxxx) data written by Modbus.

b) Configure the user parameters of the "write 4 Words(4xxxx)" module, as shown in Figure 26：

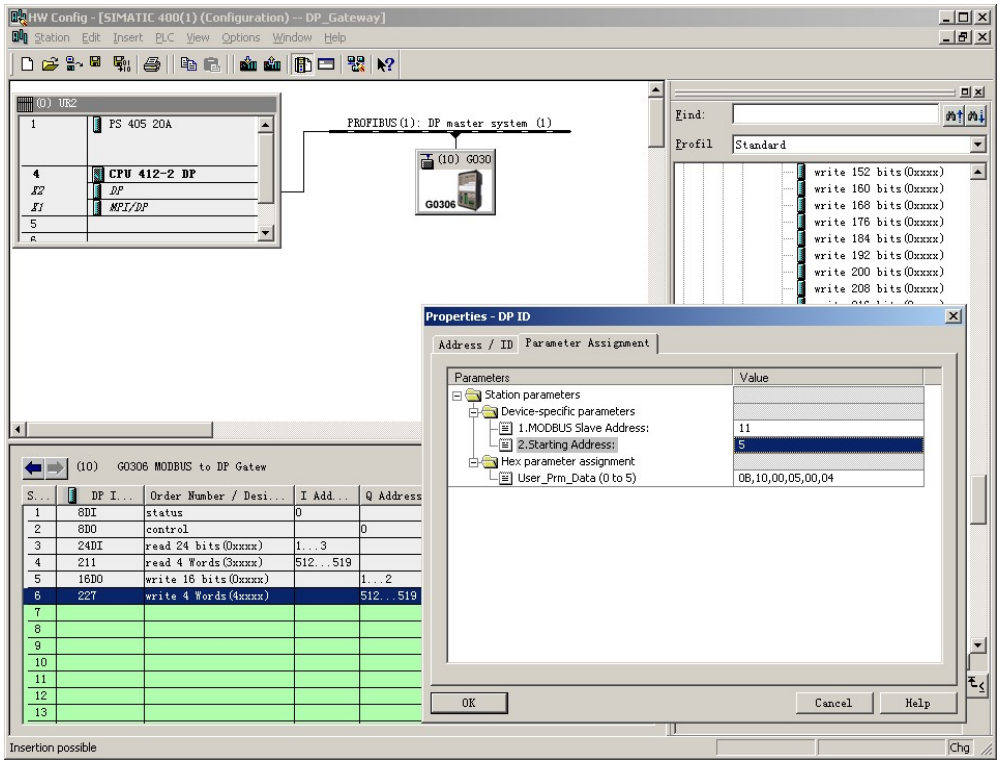


Figure 26 Configure user parameters of the "write 4 Words(4xxxx)" module

Double-click "227" or "write 4 Words(4xxxx)" or "512…519" in slot 6; select "Parameter Assigement" to complete the parameter setting of the slave address (MODBUS Slave Address) and the starting address (Starting Address) set.

Slave address: refers to the address sent by the Modbus communication module to the Modbus slave, corresponding to the first byte of the Modbus message.

Start address: refers to the 4xxxx start address to be written. Note: The starting address of the register 40000 in the message corresponds to the address 40001 in the device, and others are postponed.

“1.MODBUS Slave Address:”⇒ Type in Modbus slave address 11, as shown in Figure 26.

“2.Starting Address:”⇒ Type in the starting address 0006 of register 0xxxx to be written, and set the address to 5⇒“OK”， Figure 26。

c) Correspondence between PROFIBUS address and Modbus address

As shown in Figure 27, QB512..QB519 is the PROFIBUS output data address assigned to this Modbus module by the PROFIBUS master, a total of 8 bytes, corresponding to the 4 Words (4xxxx) written in this Modbus message to the Modbus device.

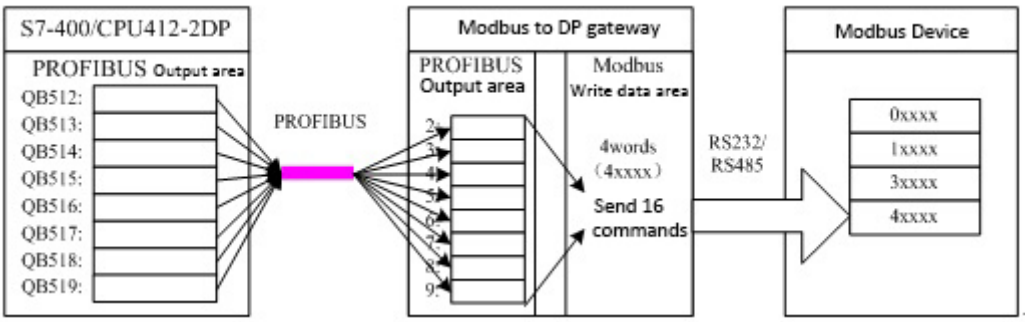


Figure 27 Correspondence between PROFIBUS address and Modbus address

- **Example of using write single coil module (module 200)**

"force single bit (05H Command)" The module uses function code 5 to write data to and from a coil

a) Add"force single bit (05H Command)"nodule，as shown in Figure 28：

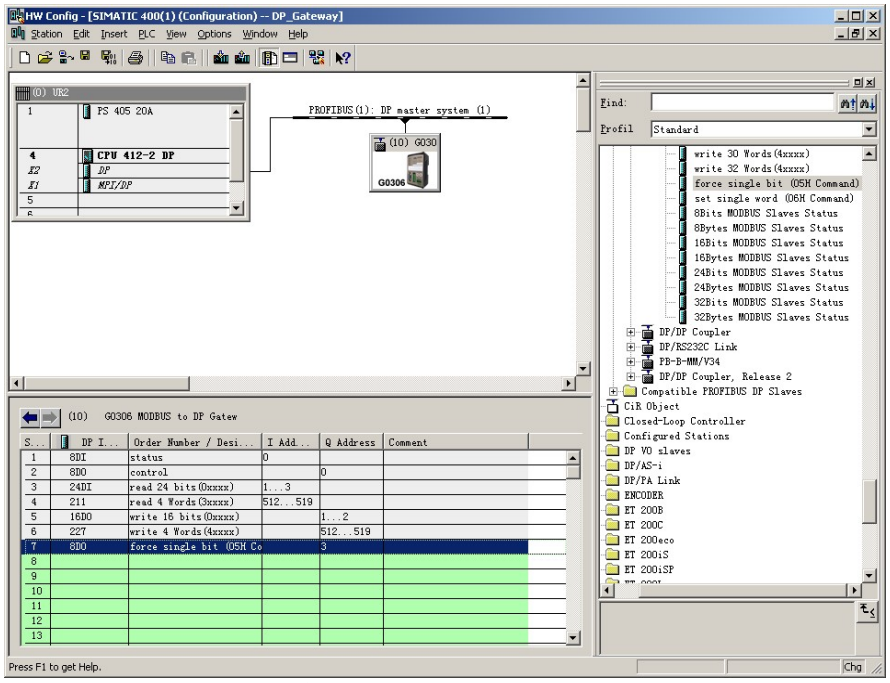


Figure 28 Add"force single bit (05H Command)"module

Take this module into slot 7 as an example, select slot 7 and double-click the "force single bit (05H Command)" module. QB3 is the output data address assigned by the Profibus master station to the gateway, corresponding to the 1-bit (0xxxx) data written by Modbus.

b) Configure the user parameters of the "force single bit (05H Command)" module, as shown in Figure 29：

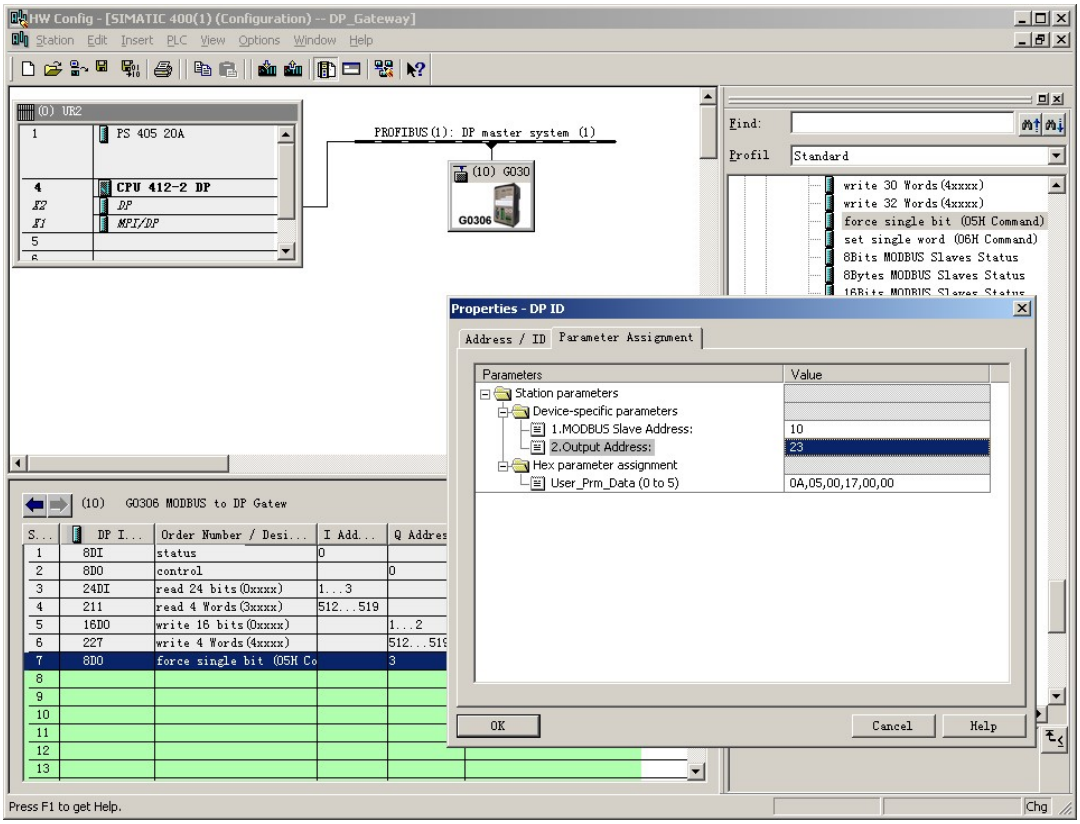


Figure 29 Configure "force single bit (05H Command)" module user parameters

Double-click "8DO" or "force single bit (05H Command)" or "3" in slot 7; select "Parameter Assigement" to complete the parameter setting of the slave address (MODBUS Slave Address) and output address (Output Address).

Slave address: refers to the address sent by the Modbus communication module to the Modbus slave, corresponding to the first byte of the Modbus message.

Output address: refers to the 0xxxx output address to be written. Note: The coil output address 00000 in the message corresponds to the address 00001 in the device, and others are postponed。

“1.MODBUS Slave Address:”⇒ Type the Modbus slave address 10， as shown in Figure 29.

“2.Output Address:”⇒ Type in the output address 00024 of the coil 0xxxx to be written, and set the address to 23⇒“OK”， as shown in Figure 29

c) Correspondence between PROFIBUS address and Modbus address

As shown in Figure 30, QB3 is the PROFIBUS output data address assigned to this Modbus module by the PROFIBUS master, which is 1 byte, corresponding to the single coil (0xxxx) written to the Modbus device in this Modbus message. The Modbus module sends Modbus command No. 05 according to the value of QB3 to set the single coil 0xxxx of the Modbus device to 1 or 0. The output address of the single coil in this example is 00023, if QB3=0, the single coil 00024 is set to 0; if QB3≠0, the single coil 00024 is set to 1.

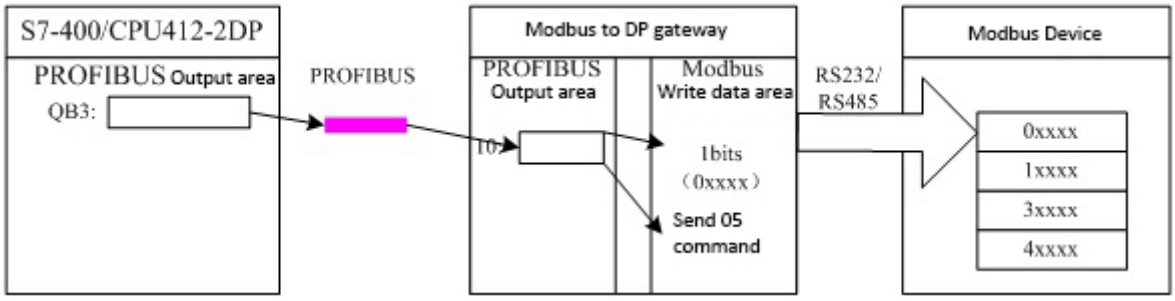


Figure 30 Correspondence between PROFIBUS address and Modbus address

- **Example of using write single register module (module 201)**

"set single word (06H Command)" The module uses function code 6 to write data to and from a register.

a) Add "set single word (06H Command)" module, as shown in Figure 31:

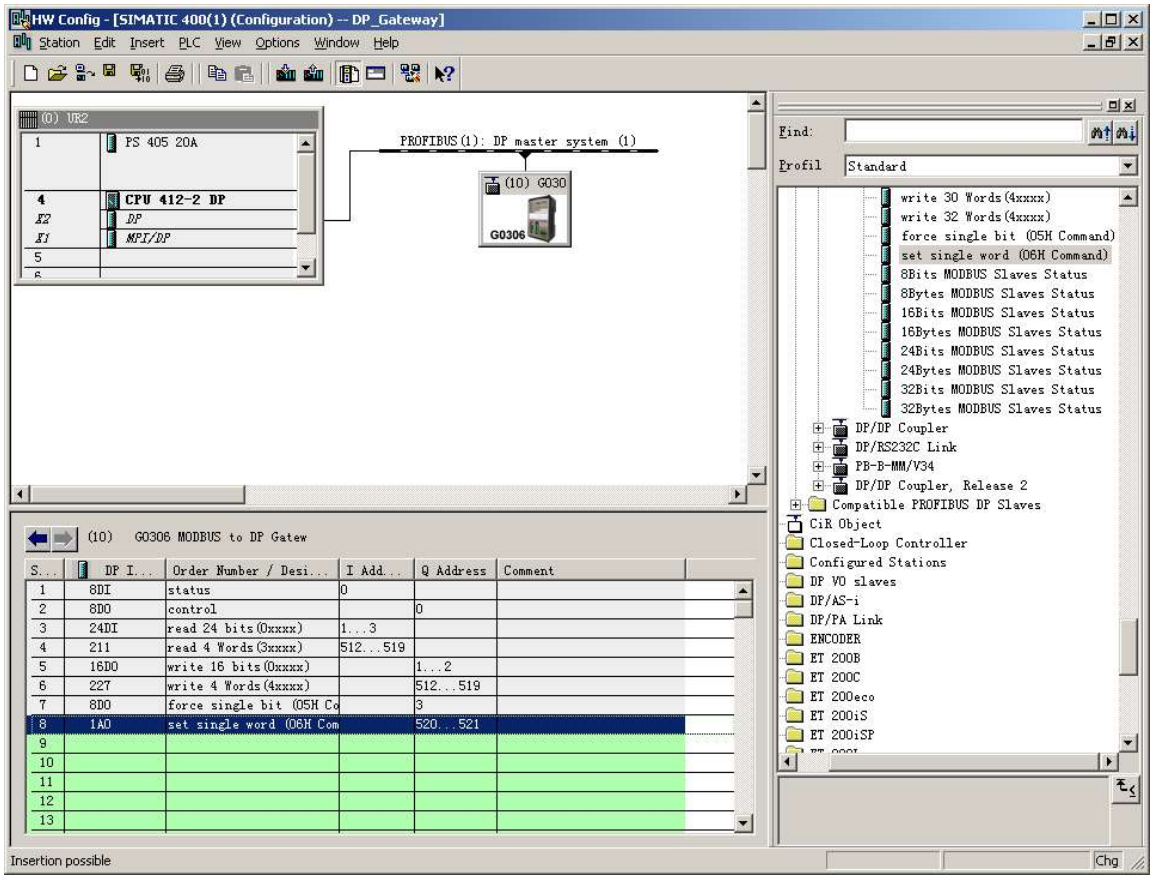


Figure 31 Add"set single word (06H Command)"module

Take this module into slot 8 as an example, select slot 8 and double-click the "set single word (06H Command)" module. QB520...QB521 is the output data address assigned by the Profibus master station to the gateway, corresponding to the 1-word register (4xxxx) data written by Modbus.

b) Configure the user parameters of the "set single word (06H Command)" module, as shown in Figure 32：

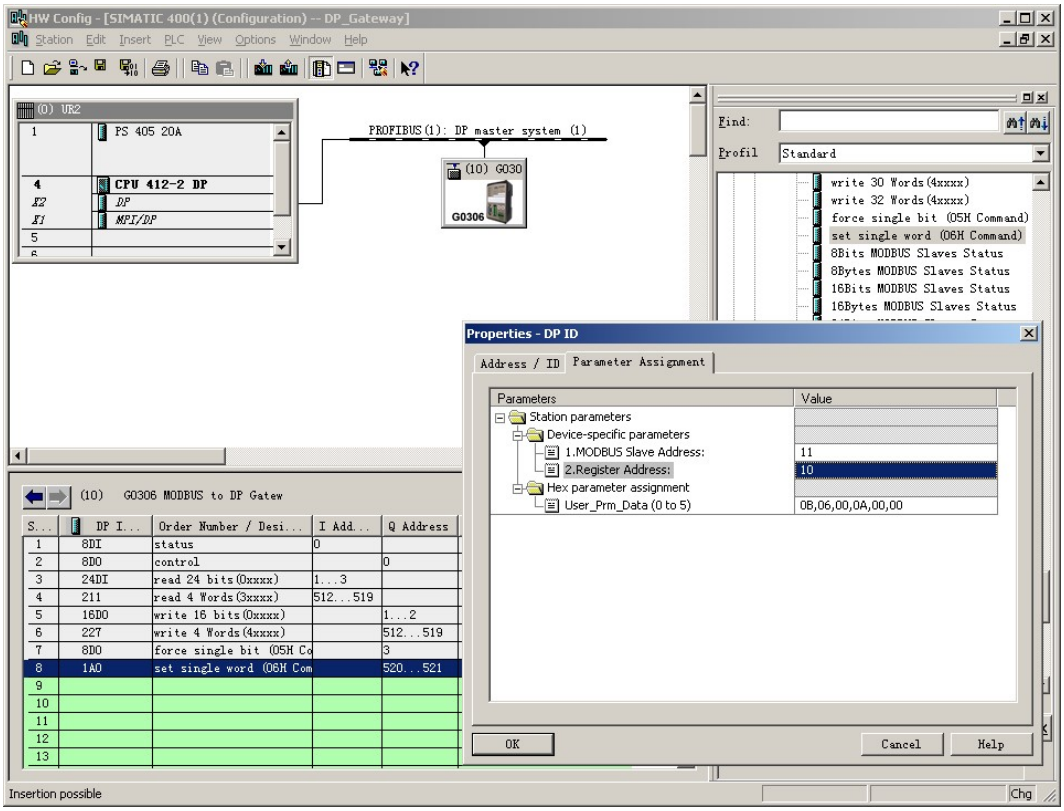


Figure 32 Configure "set single word (06H Command)" module user parameters

Double-click "1AO" or "set single word (06H Command)" or "520…521" in slot 8; select "Parameter Assigement" to complete the parameter setting of the slave address (MODBUS Slave Address) and output address (Register Address) set.

Slave address: refers to the address sent by the Modbus communication module to the Modbus slave, corresponding to the first byte of the Modbus message.

Register address: refers to the 4xxxx register address to be written. Note: The coil output address 40000 in the message corresponds to the address 40001 in the device, and others are postponed.

“1.MODBUS Slave Address:”⇒ Type in Modbus slave address 11, as shown in Figure 32

“2. Register Address:”⇒ Type in the address 40011 to be written to the register 4xxxx, set the address to 10⇒“OK”， Figure 32

c) Correspondence between PROFIBUS address and Modbus address

As shown in Figure 33, QB520...521 is the PROFIBUS output data address assigned to this Modbus module by the PROFIBUS master station, a total of 2 bytes, corresponding to the 1-word register (4xxxx) written to the Modbus device in this Modbus message.

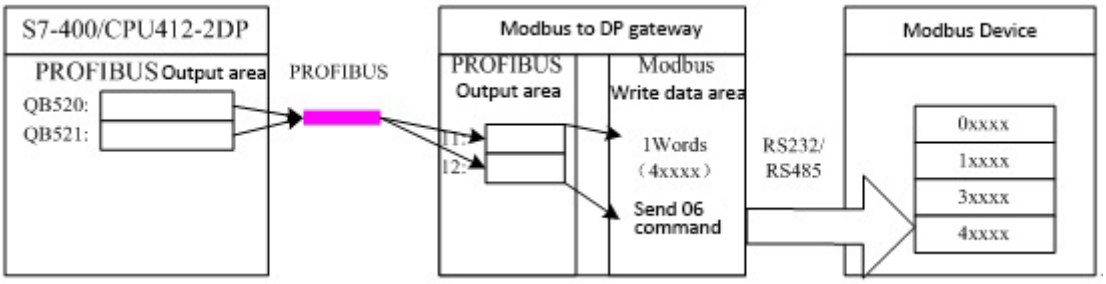


Figure 33 Correspondence between PROFIBUS address and Modbus address

- **Examples of using xxx-bit Modbus slave monitoring modules (modules 202, 204, 206, 208）**

a) Configure "3.MODBUS Slave Monitoring" device user parameters, as shown in Figure 34：

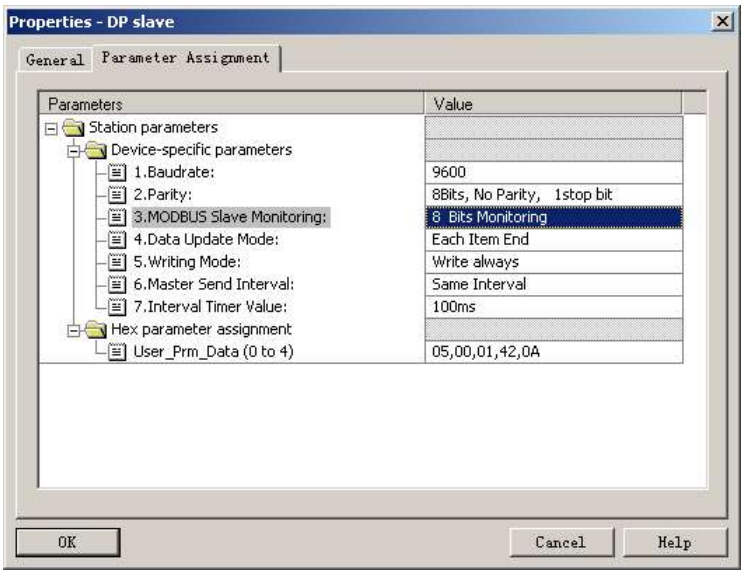


Figure 34 Configure "3.MODBUS Slave Monitoring" device user parameters

As shown in Figure 34, if you want to use the xxx-bit Modbus slave monitoring module, you must configure the corresponding Modbus slave monitoring parameters in the device user parameters. This example configures "8 Bits Monitoring", the corresponding "8Bits MODBUS Slaves Status" module should be used.

b) Add“8Bits MODBUS Slaves Status”module， as shown in Figure 35：

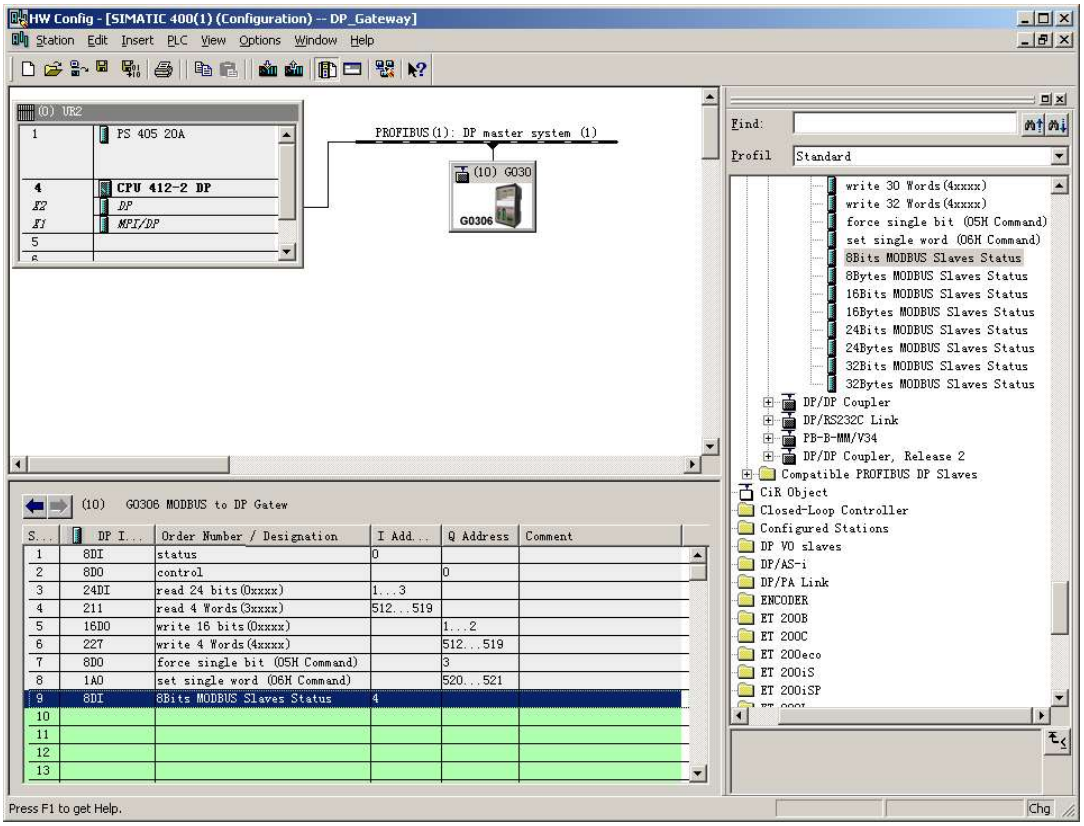


Figure 35 Add“8Bits MODBUS Slaves Status”module

As described above, this type of module must be placed after all Modbus communication modules, so according to the previous example, slots 1-8 already have corresponding modules, then the "8Bits MODBUS Slaves Status" module will be filled into slot 9. Select slot 9 and double-click the "8Bits MODBUS Slaves Status" module. IB4 is the input data address assigned by the Profibus master station to the gateway, a total of 1 byte, each bit corresponds to the Modbus device status to be monitored.

c) Configure the user parameters of the "8Bits MODBUS Slaves Status" module, as shown in Figure 36：

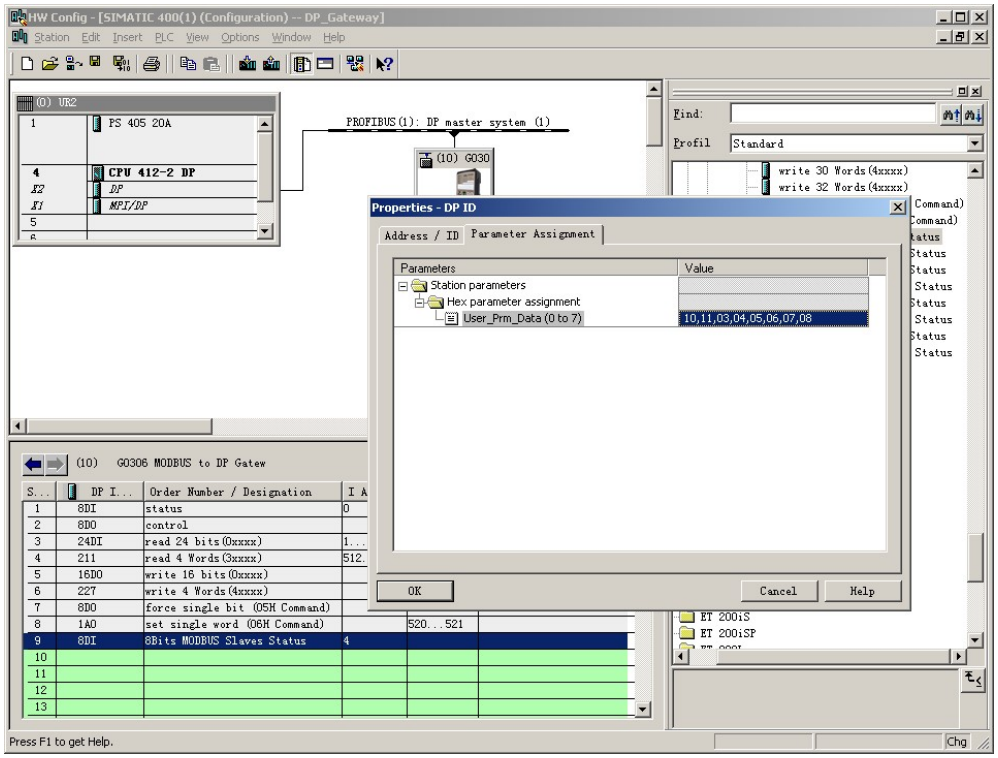


Figure 36 Configure the slave address to be monitored

As shown in Figure 36, each device corresponds to one byte of module user parameters. In the previous example, a total of two slave stations 10 and 11 are configured, so these two addresses are filled in the module user parameters.

After the program runs, you can monitor the status of slave stations 10 and 11 in the PROFIBUS address IB4, as shown in the following table：

Table 18 Modbus slave status monitored in IB4

B7	B6	B5	B4	B3	B2	B1	B0
Not used in this example	Not used in this example	Not used in this example	Not used in this example	Not used in this example	Not used in this example	Address 11 Communication status	Address 10 Communication status
--	--	--	--	--	--	0：There is a response	0：There is a response
--	--	--	--	--	--	1：No response	1：No response

B0=0，Indicates that the gateway sends a command to the Modbus slave station with address 10 according to the current configuration, and the gateway can receive the correct response message sent by the slave station.

B0=1， Indicates that the gateway sends commands to the Modbus slave with address 10 according to the current configuration, but the slave does not respond after timeout or does not connect to the slave with address 10 at all.

The meanings of other bits are similar, except that the monitored slave address is different.

Note: If the number of slaves is less than the detected number, then the order of the slave address configuration is from low to high and the number of configured slaves must be the same as the actual number of slaves (for example, there are 2 slaves in this example, but the configured module 8 slave stations can be monitored, then the addresses of these two slave stations can only be configured in B0 and B1. The order of the

addresses can be reversed, for example, B1 monitors address 10, B0 monitors address 11)

- **Examples of using xxx byte Modbus slave monitoring module（Module203、205、207、209）**

a) Configure "3.MODBUS Slave Monitoring" device user parameters, as shown in Figure 37：

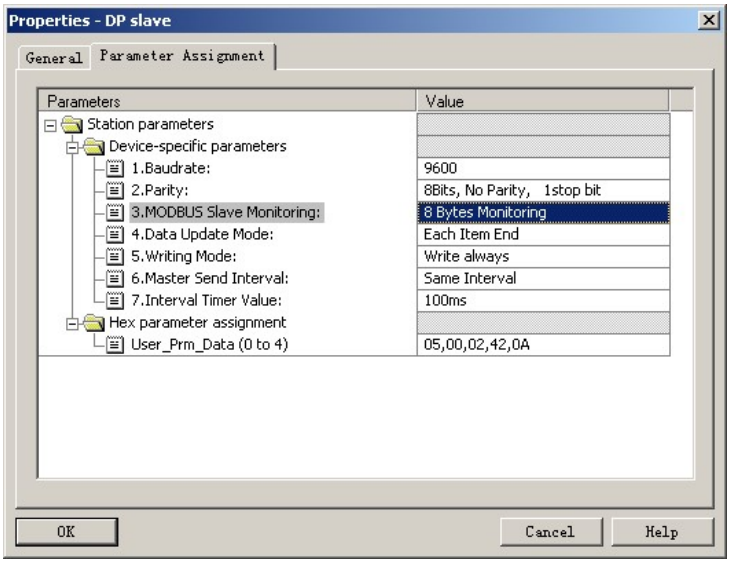


Figure 37 Configure "3.MODBUS Slave Monitoring" device user parameters

As shown in Figure 37, if you want to use the xxx byte Modbus slave monitoring module, you must configure the corresponding Modbus slave monitoring parameters in the device user parameters. This example is based on the previous 6 examples. If the user parameters of the device are configured to "8 Bytes Monitoring", the corresponding "8Bytes MODBUS Slaves Status" module should be used.

b) Add "8Bytes MODBUS Slaves Status" module, as shown in Figure 38：

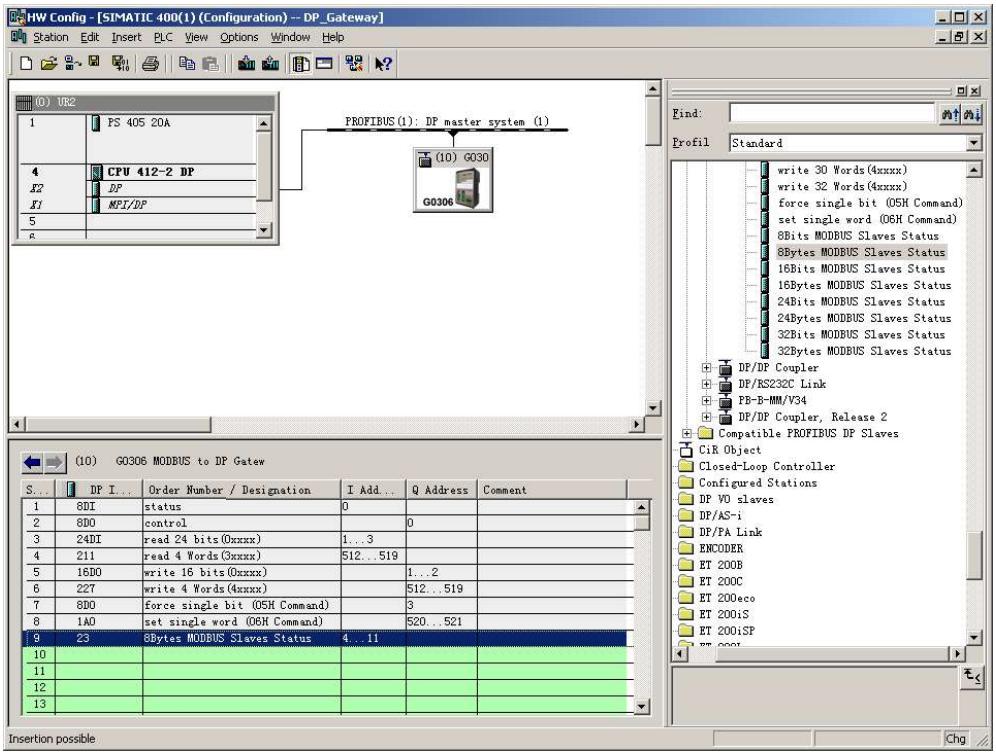


Figure 38 Add"8Bytes MODBUS Slaves Status"module

As described above, this type of module must be placed after all Modbus communication modules, so

according to the previous example, slots 1-8 already have corresponding modules, the "8Bytes MODBUS Slaves Status" module will be filled into slot 9. Select slot 9 and double-click the "8Bytes MODBUS Slaves Status" module. IB4…11 is the input data address assigned by the Profibus master station to the gateway, a total of 8 bytes, each byte corresponds to the Modbus device status to be monitored.

c) Configure the user parameters of the "8Bytes MODBUS Slaves Status" module, as shown in Figure 39：

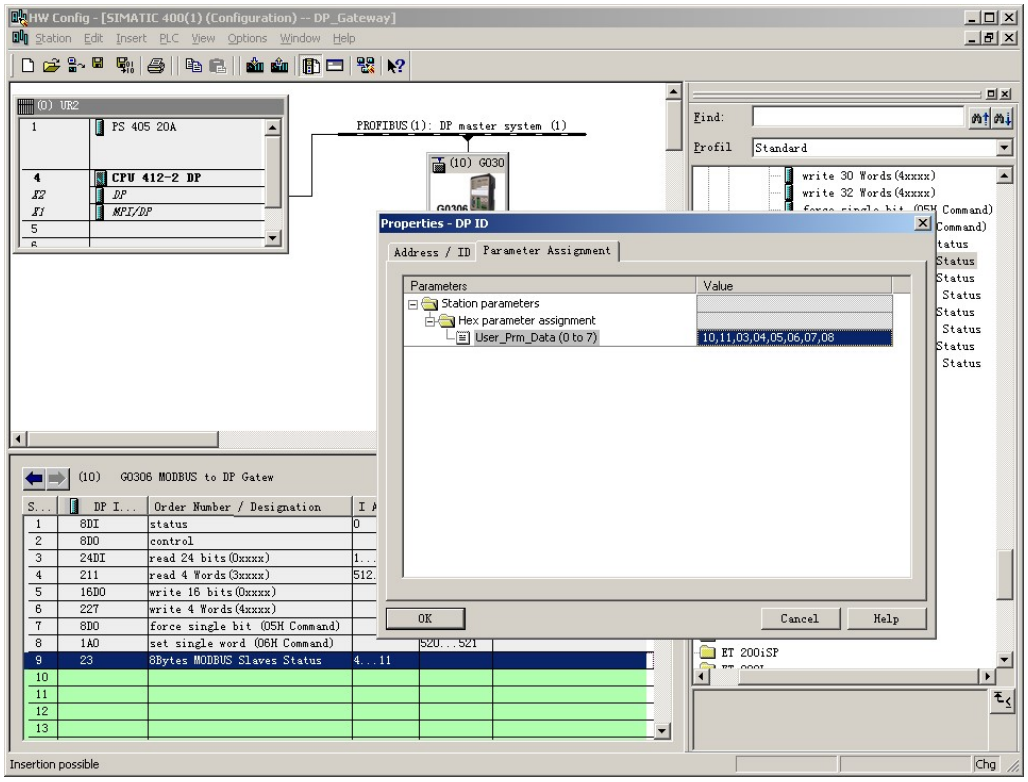


Figure 39 Configure the slave address to be monitored

As shown in Figure 39, each device corresponds to one byte of module user parameters. In the previous example, a total of two slave stations 10 and 11 are configured, so these two addresses are filled in the module user parameters.

After the program runs, the status of 8 slaves can be monitored in the PROFIBUS address IB4...11. Since there are only two slaves in this example, IB4 is the status of slave 10 and IB5 is the status of slave 11.

Table 19 A slave station can be detected status table

B7: Parity check	B6:CRCcheck	B5:Keep	B4-B1: Exception response code	B0: Response timeout
0: Current slave parity check is correct	0: The current slave CRC is correct		See Appendix A.3 for details	0: The current slave response has not timed out
1: Current slave parity error	1: Current slave CRC error			1: Current slave response timeout

This part is different from the total status module, each byte is fixed to indicate a slave station, and the status of each slave station can be checked in detail.

Note: If the number of slaves is less than the detection number, the order of the slave address configuration is from low byte to high byte and the configured number of slaves must be the same as the actual number of slaves (for example, there are 2 slaves in this example, But the configured module can monitor 8 slaves, so the addresses of these two slaves can only be configured in the first two bytes. The order of the addresses can be

reversed, for example, IB5 monitors address 10, IB4 monitors address 11).

(2) How to use the GSD file of G0306-SS

- **Configure device user parameters**

In the interface shown in Figure 13, right-click on the gateway device and select Object Properties Parameter Assignment.

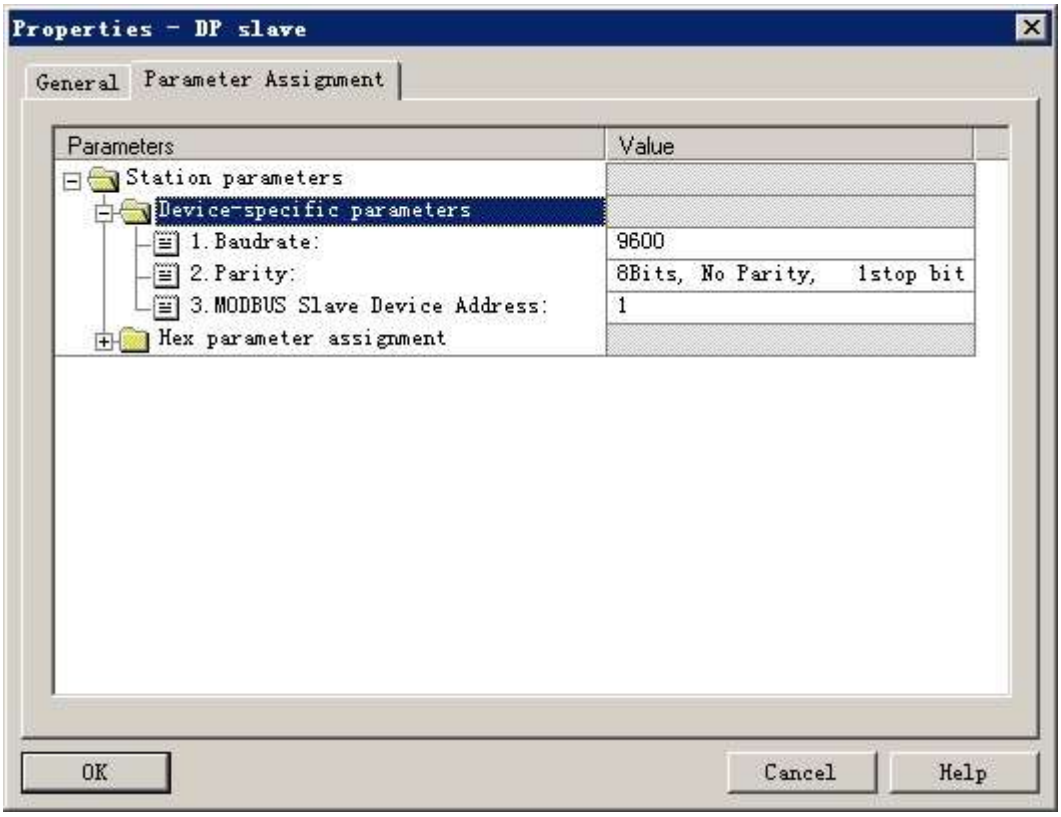


Figure 40 G0306-SS Device user parameter configuration interface

Here you can modify all the parameters mentioned in Table 14. The user should make modifications according to the actual situation. Configurable gateway as Modbus slave's baud rate, checksum, and Modbus slave address, etc.

- **Introduction to Configure Gateway Module**

As mentioned in Chapter 4.3.1 (2), this gateway GSD contains 20 slots, 67 modules, and supports up to 237 user parameters. The specific meaning of each module option is shown in Table 13.

Among them, slot 1 is fixed as the overall state module (module 2), slot 2 is fixed as the control module (module 3), and the remaining 20 slots can be configured as required.

Take the configuration "Input: 24 bits (0xxxx)" module (module 6) into slot 3 as an example, select slot 3, and double-click the "Input: 24 bits (0xxxx)" module in the device list on the right to add the module Into slot 3.

This gateway forms a Modbus storage area according to the module sequence actually configured by the user. The following will introduce in detail how to use each module.

- **Overall status module (module 2)**

This module displays the sending and receiving status of the gateway message in real time and detects whether the received message is abnormal. The exception will not be automatically eliminated until the error flag

is cleared. Refer to the control module (module 3) for the method of clearing the error mark.

B7: Parity check	B6:CRCcheck	B5:Keep	B4-B1: Exception response code	B0: Send/receive
0: Current slave parity check is correct	0: The current master station CRC is correct	Do not use	See Appendix A.3 for details	0: Send message or wait to receive
1: Current slave parity error	1: The current master station CRC error	Do not use		1: Receive or process messages

a) B0: Send/receive

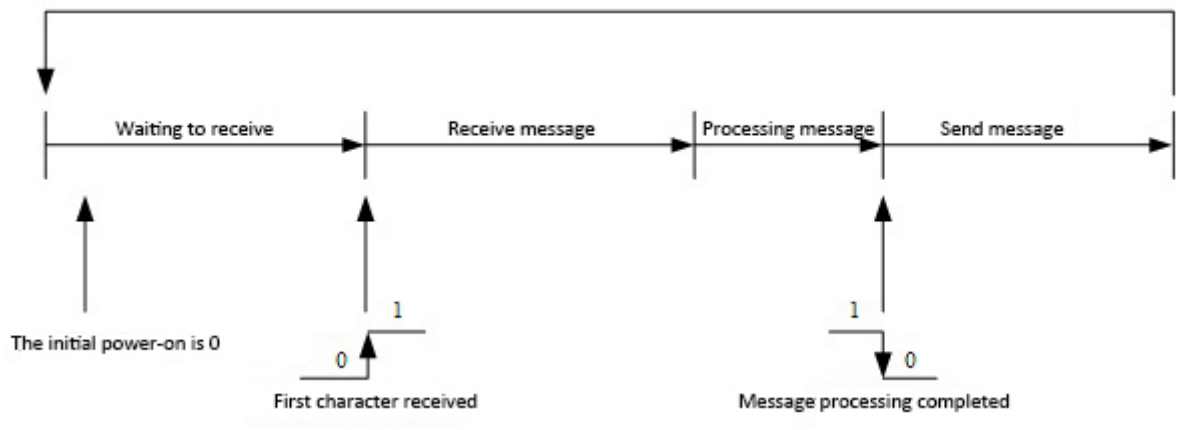


Figure 41 Transmitting and receiving state transition diagram

Since G0306-SS is a slave station, the gateway automatically enters the waiting state after power on.

b) B4-B1: Abnormal response code

When the gateway receives the message sent by the master station, there is no transmission error, but the gateway cannot correctly execute the master station command or can not make a correct response, the gateway will reply with "abnormal response code". See Appendix A.3 for details.

c) B6:CRC check

When the gateway receives a CRC check error in the message sent by the master station, this bit is set to 1. At this time, the gateway considers the message data to be unreliable, does not execute the command, and does not respond to the message.

d) B7: Parity check

When the gateway receives data and finds a character parity error, this bit is set to 1. At this time, the gateway considers that the MODBUS response data is unreliable, and does not execute commands or respond to messages.

- **Control module (module 3)**

This module is mainly used to control Profibus output and clear error flags.

Table 20 Control module format

B7: Error mark	B6-B1:Keep	B0:Profibus Output enable
0: No clear operation	——	0: Prohibit Profibus output data from entering Modbus 1xxxx and 3xxxx

1: Clear error mark B7~B1	——	1: Enable Profibus output data to enter Modbus 1xxxx and 3xxxx

a) B0: Profibus Output enable

Control whether Profibus output data enters Modbus 1xxxx and 3xxxx storage areas. The initial storage area data are all 0.

b) B7: Error mark

If this bit is set to 1, the B7-B1 bit in the overall status module is cleared. When the bit is 0, it will re-check whether an abnormal message is received.

- **Example of using input xxx bit module (module 4-19)**

Using these modules, the data in Modbus storage area 0xxxx can be mapped to Profibus input area Ix.y. Users can use function codes 1, 5, and 15 to operate Modbus storage area 0xxxx.

Take the "Input: 32 bits (0xxxx)" module as an example. This module maps the 32 coils of Modbus storage area 0xxxx to the Profibus input area.

a) Add the "Input: 32 bits(0xxxx)" module, as shown in Figure 42：

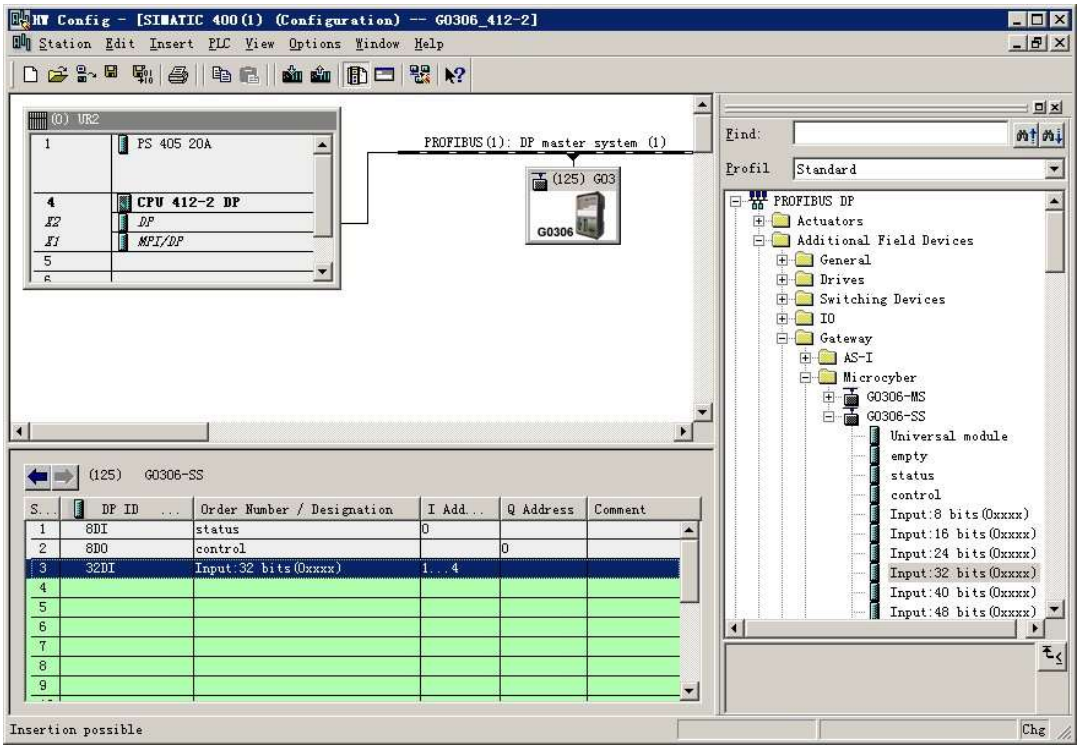


Figure 42 Add“Input:32 bits(0xxxx)”module

Take this module into slot 3 as an example, select slot 3 and double-click the "Input: 32 bits(0xxxx)" module. IB1-IB4 are the input data addresses assigned by the Profibus master station to the gateway, corresponding to Modbus coil 00001-00032.

b) Correspondence between Modbus storage area 0xxxx and Profibus input area

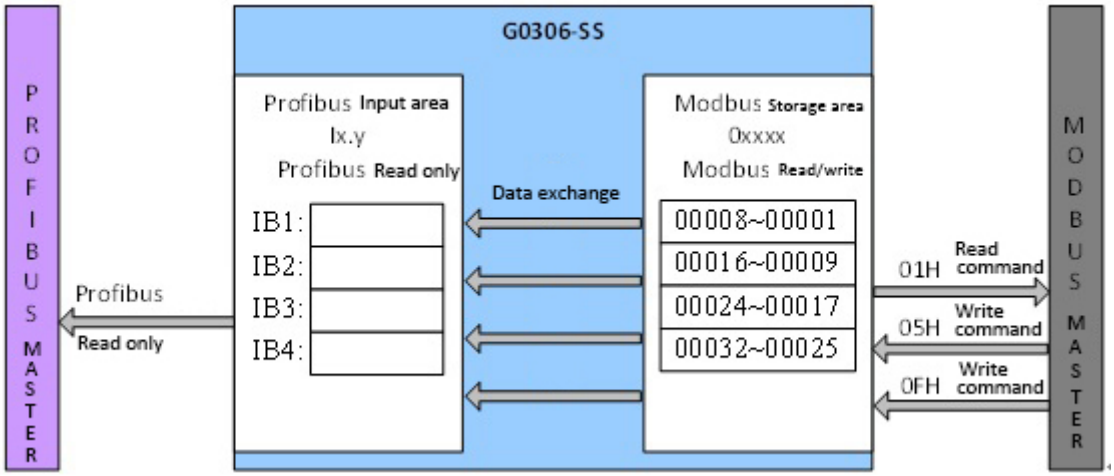


Figure 43 Correspondence between Modbus storage area 0xxxx and Profibus input area

Note: IB0 corresponds to the overall status module. The Modbus side coil address must start from 00001. When another input xxx bit module is inserted, the Modbus coil addresses are assigned sequentially. For example: insert another "Input: 32 bits (0xxxx)" module, the address sequence is assigned as 00033-00064 consecutively, corresponding to IB5-IB8 of Profibus

- **Example of using output xxx bit module (module 36-51)**

Using these modules, the data in the Profibus output area Qx.y can be mapped to the Modbus storage area 1xxxx. User can use function code 2 to operate Modbus storage area 1xxxx.

Take the "Output: 32 bits (1xxxx)" module as an example. This module writes 4 bytes of data in the Profibus output area to the 32 coils in the Modbus storage area 1xxxx.

a) Add "Output:32 bits(1xxxx)" module, as shown in Figure 44:

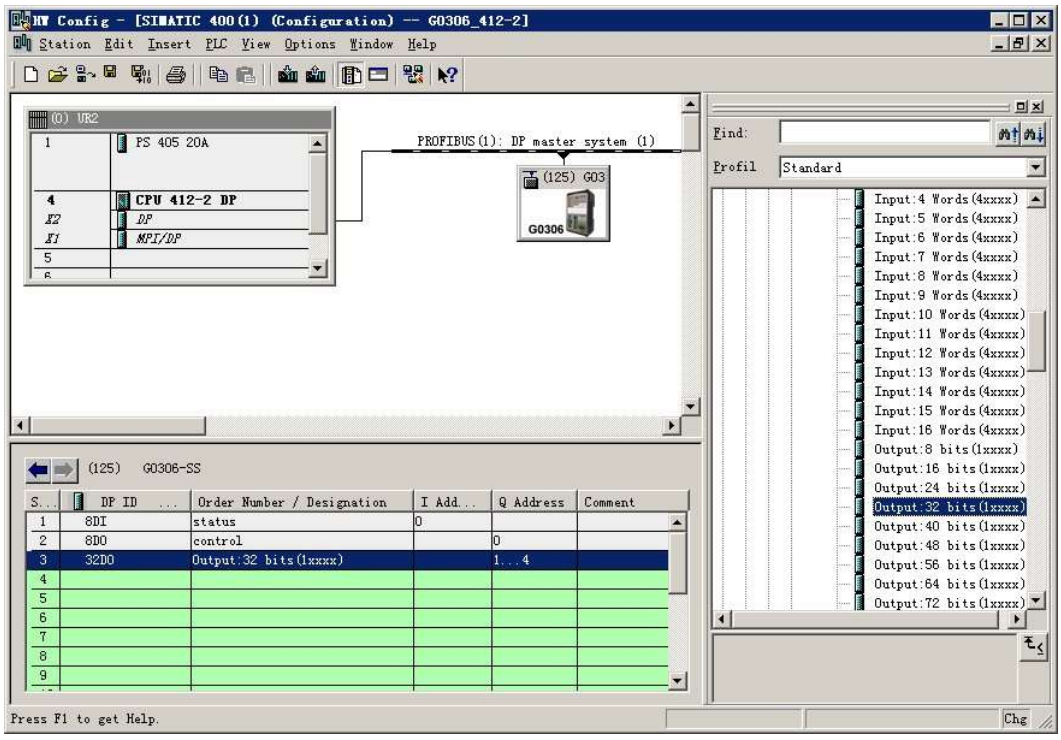


Figure 44 Add"Output:32 bits(1xxxx)"module

Take this module into slot 3 as an example, select slot 3 and double-click the "Output: 32 bits (1xxxx)" module. QB1-QB4 are the output data addresses assigned by the Profibus master station to the gateway, corresponding to Modbus coil 10001-10032.

b) Correspondence between Profibus output area and Modbus storage area 1xxxx

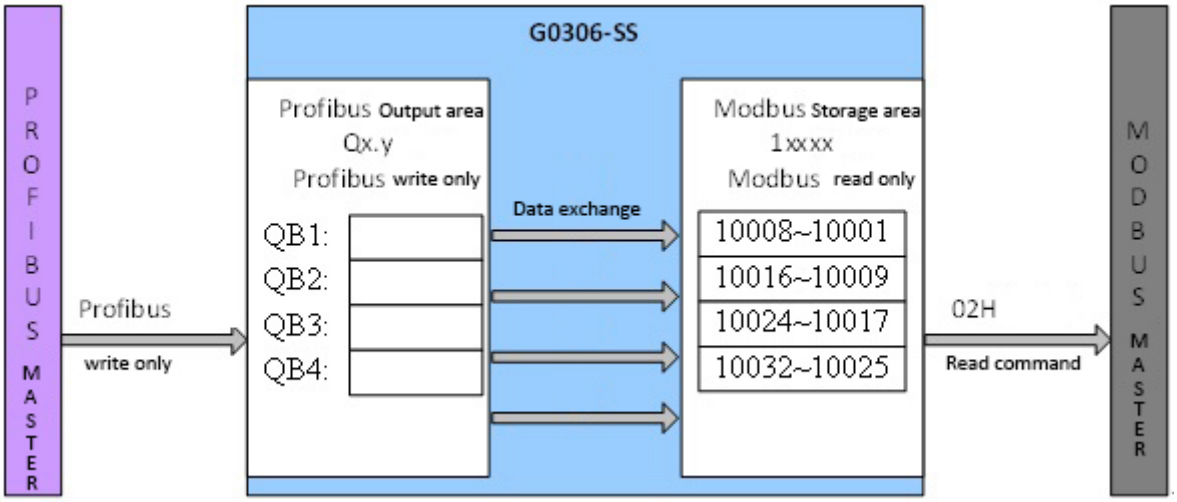


Figure 45 Correspondence between Profibus output area and Modbus storage area 1xxxx

Note: QB0 corresponds to the control module. The Modbus side coil address must start from 10001. When another output xxx bit module is inserted, the Modbus coil addresses are assigned sequentially. For example: insert another "Output:32 bits(1xxxx)" module, the address sequence will be continuously assigned as 10033-10064, corresponding to QB5-QB8 of Profibus.

- **Example of using input xxx byte module (module 20-35)**

Using these modules, the data in the Modbus storage area 4xxxx can be mapped to the Profibus input area IWx.y. Users can use function codes 3, 6, 16 to operate Modbus storage area 4xxxx.

Take the "Input: 4 Words (4xxxx)" module as an example. This module maps the 4 registers of Modbus storage area 4xxxx to the Profibus input area.

a) Add“Input:4 Words(4xxxx)”module，as shown in Figure 46：

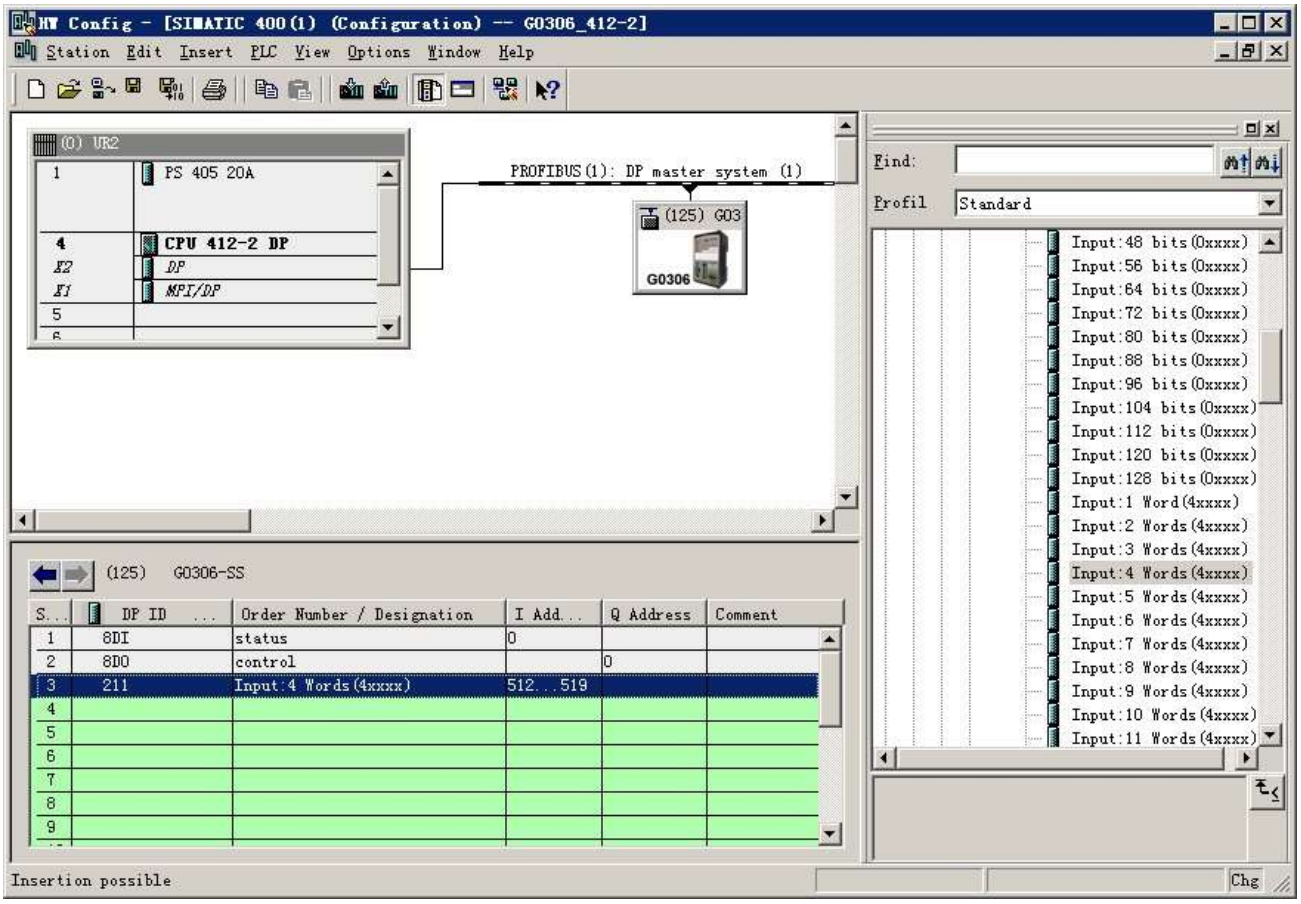


Figure 46 Add“Input:4 Words(4xxxx)”module

Take this module into slot 3 as an example, select slot 3 and double-click the "Input: 4 Words (4xxxx)" module. IW512- IW519 is the input data address assigned by the Profibus master to the gateway, corresponding to Modbus register 40001-40004.

b) Correspondence between Modbus storage area 4xxxx and Profibus input area

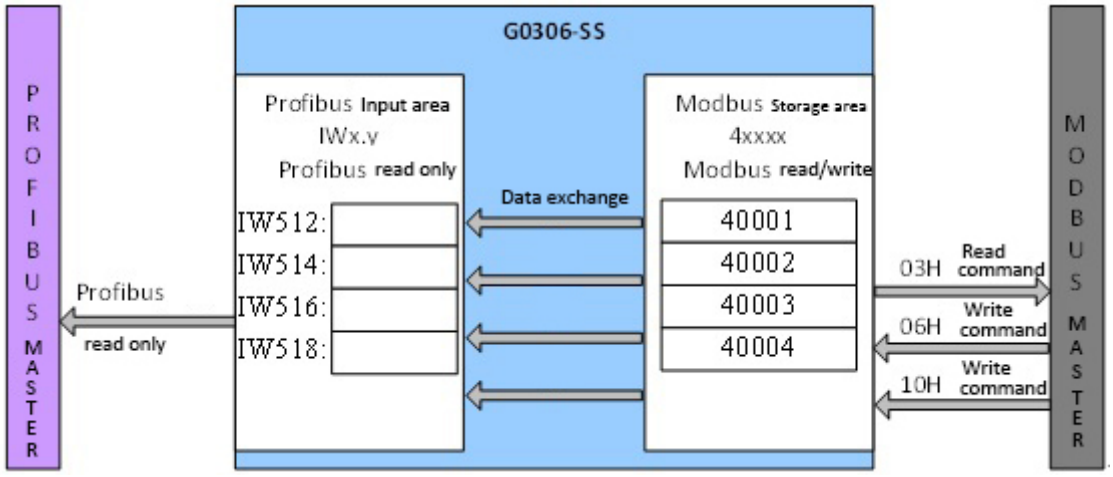


Figure 47 Correspondence between Modbus storage area 4xxxx and Profibus input area

Note: Modbus side register address must start from 40001. When another input xxx byte module is inserted, the Modbus register addresses are assigned sequentially. For example: insert another "Input: 4 Words (4xxxx)"

module, the address sequence is assigned to 40005-40008 consecutively, which corresponds to IW520-IW527 of Profibus.

- **Example of using output xxx byte module (module 52-67)**

Using these modules, the data in the Profibus output area QWx.y can be mapped to the Modbus storage area 3xxxx. The user can use function code 4 to operate the Modbus storage area 3xxxx.

Take the "Output: 4 Words(3xxxx)" module as an example. This module writes 8 bytes of data in the Profibus output area to 4 registers in the Modbus storage area 3xxxx.

a) Add“Output:4 Words (3xxxx)”module，as shown in Figure 48：

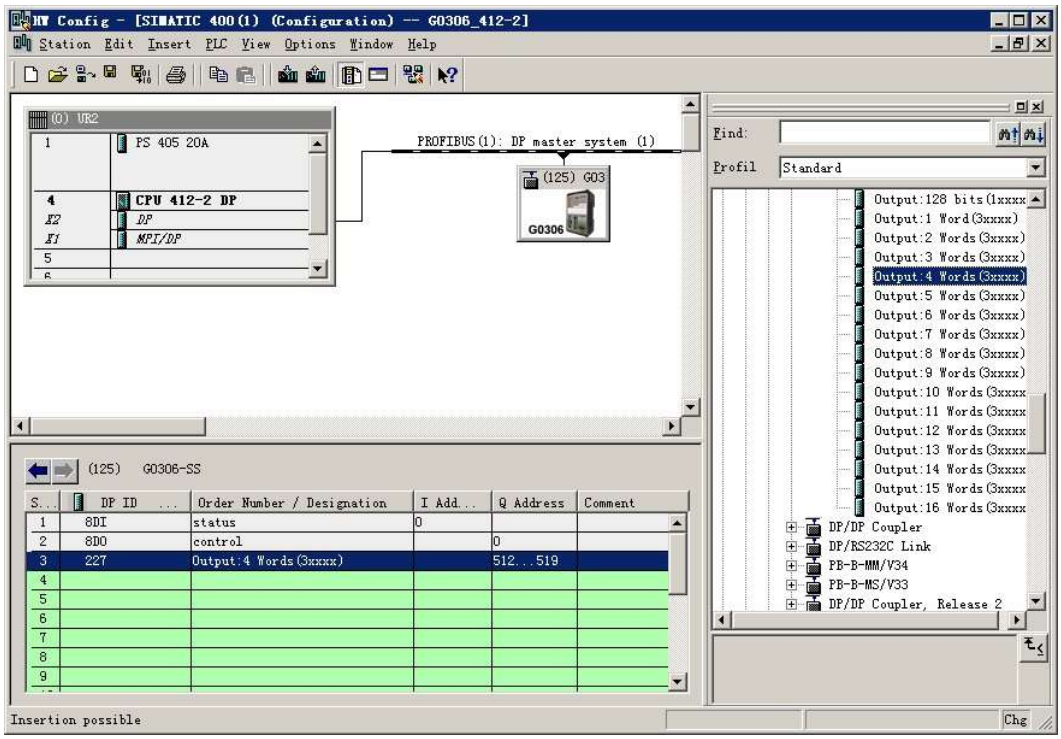


Figure 48 Add“Output:4 Words (3xxxx)”module

Take this module into slot 3 as an example, select slot 3 and double-click the "Output: 4 Words(3xxxx)" module. QW512-QW519 is the output data address assigned by the Profibus master station to the gateway, corresponding to Modbus register 30001-30004.

b) Correspondence between Profibus output area and Modbus storage area 3xxxx

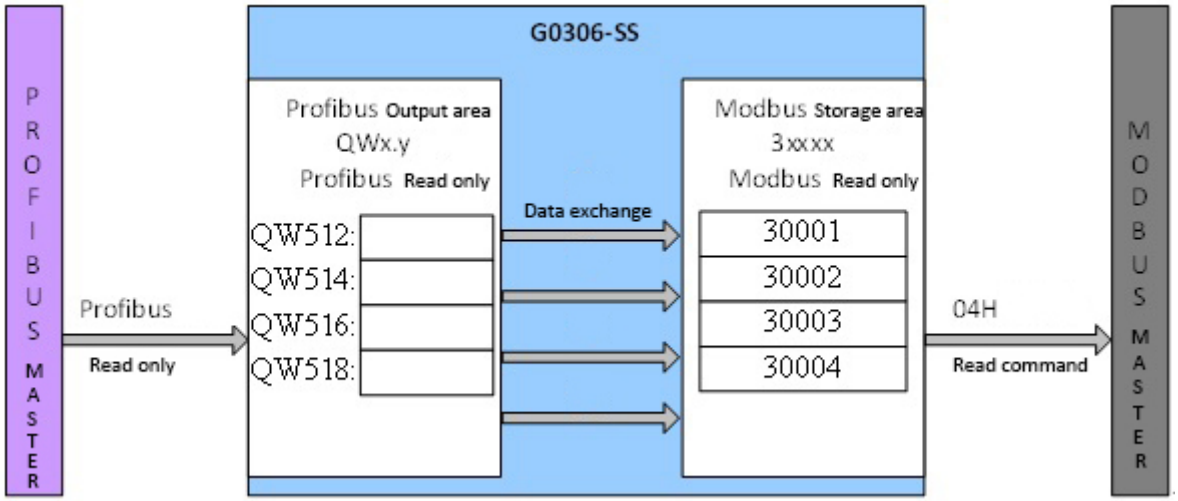


Figure 49 Correspondence between Profibus output area and Modbus storage area 3xxxx

Note: The register address of Modbus side must start from 30001. When another output xxx byte module is inserted, the Modbus register addresses are assigned in sequence. For example: insert a "Output: 4 Words (3xxxx)" module, the address sequence is assigned as 30005-30008, corresponding to QW520-QW527 of Profibus.

4.3.5. Precautions for use

When using G0306 Modbus to DP gateway, there are the following points that are easy to make mistakes, please pay attention to them when using：

1. After the G0306-MS and G0306-SS functions are switched, the device must be powered on again to take effect.
2. Although one of the address dial switches is 16-bit, it is calculated according to the decimal system. For example: 16-bit knob is 0xB, 10-bit knob is 5, then the address is 11（0xB）*10+5*1=115.
3. The detailed diagnosis function of G0306-MS is realized by configuring the device user parameter "MODBUS slave station status monitoring" and the gateway module "xxx bit/byte Modbus slave station monitoring module". If the configured "MODBUS slave station status monitoring" parameter is not 0, you must configure the corresponding "xxx bit/byte Modbus slave station monitoring module" at the last position of all configured modules.

Chapter 5 Chapter 5 Maintain

- Simple maintenance

Table 21 LED indicator status table

LED Indicator light	Colour	Normal status	Abnormal state	Abnormal reason	Correction method
Power	Green	Keep on	off	Power failure	Check power supply and connection
				Internal fault	Contact technical support
Online	Yellow	Keep on	off	Configuration error	Detect whether the hardware configuration is abnormal or not configured
				wrong address	Check whether the address matches the configuration
				Environmental failure	Check whether the DP bus connection is correct and the terminal matching is correct
				Internal fault	Contact technical support
Offline	Red	off	on	Configuration error	Detect whether the hardware configuration is abnormal or not configured
				wrong address	Check whether the address matches the configuration
				Environmental failure	Check whether the DP bus connection is correct and the terminal matching is correct
				Internal fault	Contact technical support
TxD	Green	Flashing	off	Modbus device not connected	Correctly connect Modbus devices
				Configuration error	Check whether the module parameters are correctly configured
				Power failure	Check power supply and connection
				Internal fault	Contact technical support
RxD	Yellow	Flashing	off	Modbus device not connected	Correctly connect Modbus devices
				Configuration error	Check whether the module parameters are correctly configured
				Power failure	Check power supply and connection
				Internal fault	Contact technical support

- Daily maintenance is limited to cleaning equipment.
- Failure repair: If a failure is found, please return to the factory for repair.
- Configuration error: The following configuration errors will not cause the device to be dropped on the DP side, but the Modbus communication will be abnormal.

Table 22 Configuration error detection table

No.	Anomalies	Abnormal reason	Correction method
1	Some Modbus commands are sent incorrectly	Overall status module configuration error	The overall status module is only allowed to be configured in the first slot, do not configure it in other slots
2	Some Modbus commands are sent incorrectly	Control module configuration error	The control module is only allowed to be configured in the second slot, do not configure it in other slots
3	Some Modbus commands are sent incorrectly	The slave station monitoring module of G0306-MS is not configured to the end	Please configure the slave monitoring module corresponding to the "MODBUS slave status monitoring" parameter to the last valid slot, not in the middle of the

			communication module
4	Some Modbus commands are sent incorrectly	The "MODBUS slave station status monitoring" parameter of G0306-MS does not match the configured slave station monitoring module	Please use the slave monitoring module specified by the "MODBUS slave status monitoring" parameter. If the "MODBUS slave station status monitoring" parameter is that there is no slave station monitoring module, then please do not configure the slave station monitoring module
5	After sending all configured Modbus commands, a few bytes of data will be sent	The "MODBUS slave station status monitoring" parameter of G0306-MS does not match the configured slave station monitoring module, and the configured slave station monitoring module length is greater than the "MODBUS slave station status monitoring" parameter length	Please configure the slave monitoring module to be consistent with the "MODBUS slave status monitoring" parameter

Chapter 6 Chapter 6 Technical specifications

6.1. Basic parameters

Operating Voltage	24VDC(±20%)
Rated current	I_{24V}:≤60mA
Operating temperature	-20℃～70℃
Storage temperature	-40℃～70℃
Humidity range	5%～95%RH
Modbus Physical interface	RS485 （Configurable terminal）/ RS232
Modbus Character transmission mode	RTU mode

6.2. Performance Index

Protection level	Enclosure protection grade reaches IP20
Electromagnetic Compatibility	Comply with the immunity requirements of 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 Test configuration, working conditions and performance judgments with integrated or remote signal conditioning transmitters according to"

6.3. Physical characteristics

weight	0.2kg
Structural materials	Case：ABS；Clip：POM

6.4. Default communication parameters

General default communication parameters	
Baud rate	9600
Data bit	8
Stop bit	1
check	NO
G0306-MS default communication parameters	
MODBUSSlaves Condition monitoring	NO
Data update mode	After all MD response
Write mode	Keep writing
Master sending interval	Slave response to send
Interval time value	500ms
Groove	Groove 1：Overall status module；Groove 2：Control module；Groove 3-39：No module
G0306-SS default communication parameters	
ModbusSlave address	1

6.5. Support Modbus function code

1	Read coil
2	Read discrete input
3	Read holding register value
4	Read input register value
5	Write a single coil

6	Write a single register
15	Write multiple coils
16	Write multiple register values

Appendix 1 Modbus communication protocol

Disclaimer: It is not necessary to understand the technical details of Modbus to use G0306 Modbus to DP gateway. This appendix is only for users to understand the Modbus communication protocol.

1.1 Modbus communication protocol

1) Modbus protocol is mainly used for communication between controllers. Through this protocol, two controllers can communicate with each other or between the controllers and other devices through a network (such as Ethernet). Many devices currently use Modbus communication protocol standards.

2) According to the international ISO/OSI 7-layer network model, the standard Modbus protocol defines the communication physical layer, link layer and application layer;

Physical layer: Defined asynchronous serial communication specifications based on RS232 and RS485;

Link layer: Provides media access control based on station number identification and master/slave mode;

Application layer: Provides information specification (or message format) and communication service functions;

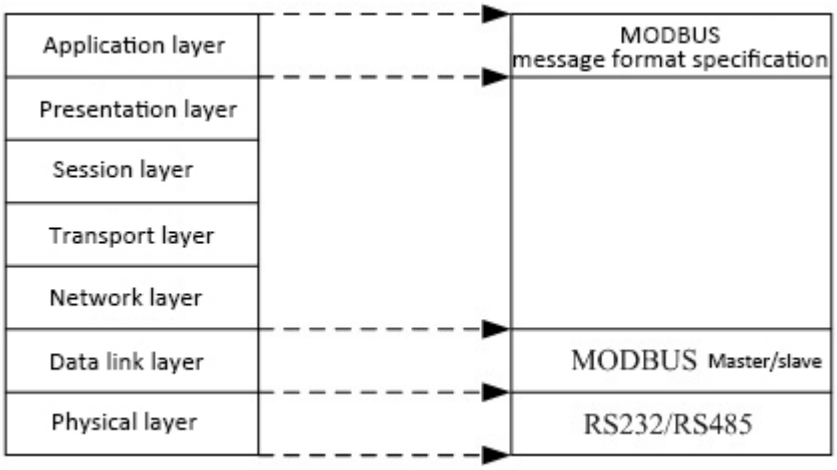


Figure 50 Modbus Protocol model

3) At present, many Modbus device applications are based on RS232/485, and there are also changes in Modbus network communication. Only the Modbus application layer (information specification) is used, while the bottom layer uses other communication protocols, such as: Ethernet + TCP/IP at the bottom layer Modbus network communication, wireless spread spectrum communication Modbus network, etc.

1.2 Key points of Modbus protocol

1) Modbus is a master/slave communication protocol. The master station actively sends a message, and only the slave station with the same call address in the message sent by the master station sends a response message to the master station.

2) When the message is sent with address 0, it is in broadcast mode, and no response from the slave is required.

3) Modbus specifies two character transmission modes: ASCII mode and RTU (binary) mode; the two transmission modes cannot be mixed.

※This product can only be used in RTU mode.

Table 23 Character transmission mode table

Characteristic	RTU mode	ASCII mode
Coding	Binary	ASCII（Print characters：0-9，a-z，A-Z）
Digits per character	Start bit：1BIT	Start bit：1BIT
Data bit	Data bit：8BITS	Data bit：7BITS
Check Digit	Parity bit (optional)：1 位	Parity bit (optional): 1 bit
Stop bit	Stop bit: 1 or 2	Stop bit: 1 or 2
Message check	CRC(Cyclic redundancy check)	LRC(Longitudinal redundancy check)

4) Transmission error check

- Transmission error check is checked by parity check and redundancy check.
- When a check error occurs, the message processing stops, the slave station does not continue to communicate, and does not respond to this message；
- Once a communication error occurs, the message is regarded as unreliable; the Modbus master station does not receive a response from the slave station after a certain period of time, that is, it makes a judgment that a communication error has occurred.

5) Message level (character level) adopts CRC-16 (cyclic redundancy error check)

6) Modbus message RTU format

Table 24 Modbus message RTU format table

Message interval not less than 3.5 characters	Address	function code	Data	CRC check	Message interval not less than 3.5 characters
	1*byte	1*byte	N*bytes	2*bytes	

1.3 Modbus abnormal response

1) The master station message received by the slave station has no transmission error, but the slave station cannot execute the master station command correctly or cannot make a correct response. The slave station will reply with an "abnormal response"

2) Exception response message format

Example: the master station sends a request message, function code 01, read 1 coil value of 0x04A1

Table 25 Request message sent by the master station (hexadecimal)

Slave address	Function code	High start address	Low start address	High number of coils	Low number of coils	CRC
0A	01	04	A1	00	01	xxxx

Since the highest coil address of the slave is 0x0400, 0x04A1 exceeds the upper limit of the address, and the slave responds as follows (note: the highest position of the function code is 1)：

Table 26 Response message sent by the slave station (hexadecimal)

Slave address	Function code	Exception code	CRC
0A	**81**	**02**	xxxx

3) Abnormal response code

Table 27 Modbus exception response codes

Exception code	Name	Meaning
0x01	Illegal function	The function of the received message is not allowed to be executed by the addressed slave. If an inquiry command is issued, this code means that there is no programming function before that
0x02	Illegal data address	The address in the data field is forbidden to the addressed slave
0x03	Illegal data	The value in the data field is forbidden to the addressed slave.
0x04	Slave equipment failure	When the server (or slave) is trying to perform the requested operation, an unrecoverable error occurs

1.4 Modbus storage area

The storage area of the controller (or Modbus device) involved in Modbus is identified by 0xxxx, 1xxxx, 3xxxx, 4xxxx;

Table 28 Modbus storage area

Store ID	Name	Type	Read/write	Storage unit address (decimal)
0xxxx	Coil	Bit	Read/write	00001~0xxxx xxxx：Related to equipment
1xxxx	Discrete input	Bit	Read only	10001~1xxxx xxxx：Related to equipment
3xxxx	Input register	Word	Read only	30001~3xxxx， xxxx：Related to equipment
4xxxx	Hold/output register	Word	Read/write	40001~xxxx： xxxx Related to equipment

1.5 Modbus function code

The Modbus protocol has three types of Modbus function codes: public function codes, user-defined function codes, and reserved function codes

1) Public function codes use

- It is a well-defined function code
- The guarantee is unique
- MODBUS organization can be changed
- Publicly proven
- Has available conformance tests
- Proved in MB IETF RFC
- Includes defined publicly assigned function codes and unassigned function codes reserved for future

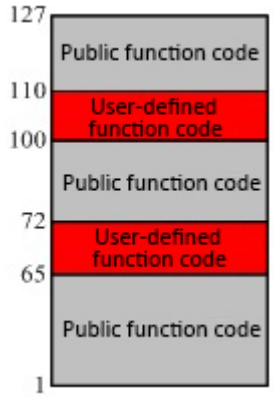


Figure 51 Modbus function code

2) User-defined function code

- There are two definition ranges of user-defined function codes, namely 65 to 72 and 100 to 110
- The user can select and implement a function code without any approval from the MODBUS organization
- There is no guarantee that the selected function code is unique
- If the user wants to reset the function as a public function code, the user must initiate an RFC to

introduce the change into the public classification and assign a new public function code

3) Reserved function code

- Function codes commonly used by some companies for traditional products, and function codes that are invalid for public use

1.5.1 01（0x01）Example of reading coil

Table 29 Master station request message format (hexadecimal)

Slave address	Function code	High bit of coil start address	Low bit of coil start address	High number of coils	Low number of coils	CRC
11	01	00	13	00	25	xxxx

Function: Read slave coil 0xxxx status

Note: The coil start address 00000 in the message corresponds to the address 00001 in the device, and others are postponed.

This example: read the coil of No. 17 (0x11) slave station, the starting address=0x0013=19, corresponding to the address 00020 in the device; the number of coils=0x0025=37; the last address=00020+37-1=00056.

Therefore, the function of this request message is: read the coil 00020—00056 of the slave station number 17 (0x11), a total of 37 coil states

Table 30 Slave station response message format (hexadecimal)

Address	Function code	Byte count	Coil status 20-27	Coil status 28-35	Coil status 36-43	Coil status 44-51	Coil status 52-56	CRC
11	01	05	CD	6B	B2	0E	1B	xxxx

Function: Slave station returns coil 0xxxx status

This example：0xCD=11001101，correspond 00020-00027；0x6B=01101011，correspond 00028-00035；

0xB2=10110010，correspond 00036-00043；0x1B=00011011，correspond 00052-00056

1.5.2 02（0x02）Example of reading discrete input

Table 31 Master request message format (hexadecimal)

Slave address	Function code	High start address	Low start address	Discrete input number high	Discrete input number low	CRC
11	02	00	C4	00	16	xxxx

Function: Read the status of the slave discrete input 1xxxx

Note: The start address of discrete input 00000 in the message corresponds to the address of 10001 in the device, and others are postponed

This example: Reading No. 17 (0x11) discrete input from the station, starting address=0x00C4=196, corresponding to address 10197; discrete input number=0x0016=22, last address=10197+22-1=10218

Therefore, the function of this request message is: Reading No. 17 (0x11) slave station discrete input 10197-10218, a total of 22 discrete inputs

Table 32 Slave station response message format (hexadecimal)

Slave address	Function code	Byte count	Discrete input 10197-10204	Discrete input 10205-10212	Discrete input 10213-10218	CRC
11	02	03	AC	DB	35	xxxx

Function: The slave returns to the discrete input 1xxxx state

1.5.3 03（0x03）Example of Reading Holding Register

Table 33 Master station request message format (hexadecimal)

Slave address	Function code	Register start address high	Low bit of register start address	High register number	Low register number	CRC
11	03	00	6B	00	03	xxxx

Function: Read the value of the slave holding register 4xxxx

Note: The register start address 00000 in the message corresponds to the address 40001 in the device, and others are postponed.

In this example: read 17 (0x11) slave station holding register value, start address=0x006B=107, corresponding address 40108; register number=0003; end address=40108+3-1=40110.

Therefore, the function of this request message is: read the value of the 3 holding registers 40108-40110 of the number 17 (0x11) slave station.

Table 34 Slave station response message format (hexadecimal)

Slave address	Function code	Byte count	Register 40108 high	Register 40108 low	Register 40109 high	Register 40109 low	Register 40110 high	Register 40110 low	CRC
11	03	06	02	2B	01	06	2A	64	xxxx

Function: Slave station returns the value of holding register 40108-40110; (40108)=0x022B，(40109)=0x0106，(40110)=0x2A64

1.5.4 04（0x04）Example of reading input register

Table 35 Master station request message format (hexadecimal)

Slave address	Function code	Register start address high	Register start address low	High register number	Low register number	CRC
11	04	00	08	00	01	xxxx

Function: Read the value of the slave input register 3xxxx

Note: The register start address 00000 in the message corresponds to the address 30001 in the device, and the others are postponed.

This example: read the value of the register input from the station on No. 17 (0x11), the starting address=0x0008=0008, corresponding to the address 30009; the number of registers=0001; the last address=30009.

Therefore, the function of this request message is: Read the value of 1 holding register 30009 of the slave station No. 17 (0x11)

Table 36 Slave station response message format (hexadecimal)

Slave address	Function code	Byte count	Input register high	Input register low	CRC
11	04	02	01	01	xxxx

Function: Slave station returns the value of input register 30009； （30009）=0x0101

1.5.5 05（0x05）Example of writing a single coil

Table 37 Master station request message format (hexadecimal)

Slave address	Function code	Coil address high	Coil address low	Disconnect mark	Disconnect mark	CRC
11	05	00	AC	FF	00	xxxx

Function: Write the value of coil 0xxxx of No. 17 slave station. The coil start address 00000 in the message corresponds to the address 00001 in the device, and the others are postponed.

Disconnect mark=0xFF00, turn on the coil

Disconnect mark =0x0000， Set coil OFF。

In this example: the starting address=0x00AC=172, the address in the corresponding device is 00173.

Therefore, the function of this request message is: write the No. 17 slave coil 00173 as ON state.

Table 38 Slave station response message format (hexadecimal, original text returned)

Slave address	Function code	Coil address high	Coil address low	Disconnect mark	Disconnect mark	CRC
11	05	00	AC(172)	FF	00	xxxx

Function: After writing No. 17 slave coil 0173 to ON, the original text will be returned

1.5.6 06（0x06）Example of writing a single register

Table 39 Master station request message format (hexadecimal)

Slave address	Function code	Register address high	Register address low	Data value high	Data value low	CRC
11	06	00	87(135)	03	9E	xxxx

Function: Write the value of a single holding register 4xxxx. The register starting address 00000 in the message corresponds to the address 40001 in the device, and the others are postponed.

This example: write the value of the single holding register 40136 of the 17th slave station=0x039E

Table 40 Slave station response message format (hexadecimal, original text returned)

Slave address	Function code	Register address high	Register address low	Data value high	Data value low	CRC
11	06	00	87(135)	03	9E	xxxx

Function: After writing the value of the single holding register 40136 of the 17th slave station = 0x039E, the original text is returned

1.5.7 15（0x0F）Example of writing multiple coils

Table 41 Master station request message format (hexadecimal)

Slave address	Function code	Coil address High	Coil address Low	Number of coils High	Number of coils Low	byte count	Coil state 20-27	Coil state 28-29	CRC
11	0F	00	13	00	0A	02	CD	00	xxxx

Function: Write multiple continuous coils 0xxxx as ON/OFF status.

Note: The coil start address 00000 in the message corresponds to the address 00001 in the device, and others are postponed.

In this example: write multiple continuous coils from the slave station with number 17 (0x11), the start address of the coil=0x0013=19, corresponding to the address 00020; the number of coils=0x000A=10; the last address=00020+10-1=00029.

Therefore, the function of this request message is: write the value of the 10 coils 00020—00029 of the slave station No. 17 (0x11); among them, the address of 00020-00027 is written in 0xCD, and the address of 00028-00029 is written in 0x00

Table 42 Slave station response message format (hexadecimal)

Slave address	Function code	Coil address high	Coil address low	High number of coils	Low number of coils	CRC
11	0F	00	13	00	0A	xxxx

Function: return coil address and coil number

1.5.8 16（0x10）Write multiple registers

Table 43 Master station request message format (hexadecimal)

Address	Function code	Register address high	Register address low	High register number	Low register number	byte count	data High position	data Low	data High position	data Low	CRC
11	10	00	13	00	0A	02	CD	00			xxxx

Function: preset multiple holding register values 4xxxx of the slave.

Note: The starting address of the holding register 40000 in the message corresponds to the address 40001 in the device, and others are postponed.

In this example: preset multiple holding register values from the slave station of No. 17 (0x11), the register start address=0x0087=135, the corresponding address is 40136, the number of coils=0x0002=2, the last address=40135+2-1=40137;

Therefore, the function of this request message is: preset the value of 2 holding registers of the number 17 (0x11) slave station; write 0x0105 to the 40136 address; write 0x0A10 to the 40137 address

Table 44 Slave station response message format (hexadecimal)

Slave address	Function code	Register address high	Register address low	High register number	Low register number	CRC
11	10	00	87	00	02	xxxx

Function: return register address and register number

Appendix 2 G0306 Modbus to DP Gateway Selection Code Table

Selection code table	G0306	Modbus to DP gateway			
		Codename	Hardware interface		
		R4（Can be omitted）	RS485		
			Codename	Software interface	
			MRM（Can be omitted）	Modbus RTU Master	
	G0306 - （ R4 - MRM ） —— Selection example				

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