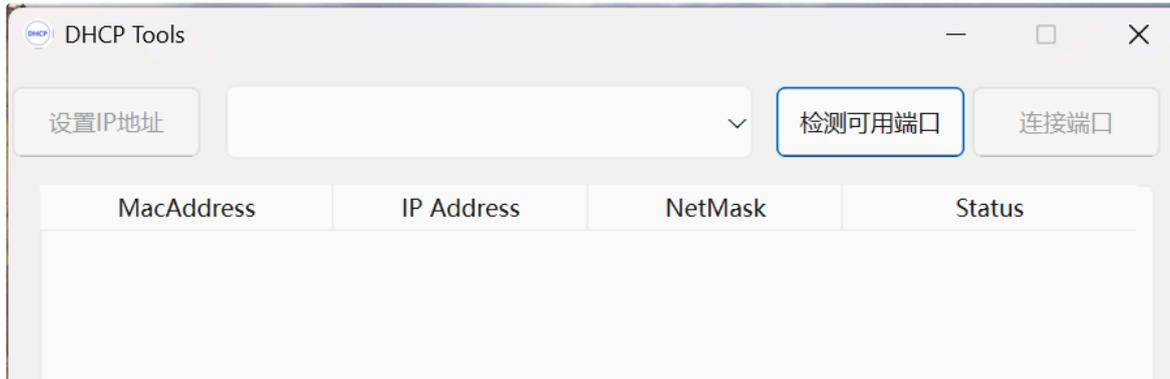


1) Set the IP Address of the Sentinel Modbus/TCP IO-Link Master

Method 1: Set via Sentinel “DhcpTool” IP Assignment Software

- ① Download “DhcpTool V1.2” from the Sentinel official website and install it (**Windows 10 or later only**). Launch the software.

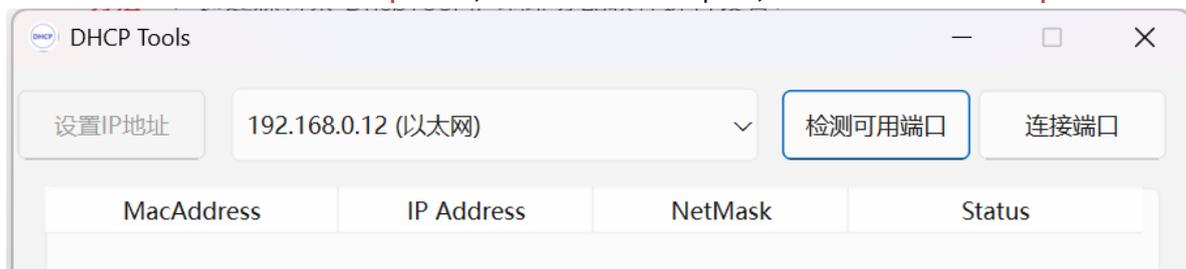


- ② Connect the Sentinel Modbus/TCP IO-Link master to the PC, then set the IP address selector switches to “0xFF” (on the right-side window: set **ADDR_H = F** and **ADDR_L = F**; refer to the manual).

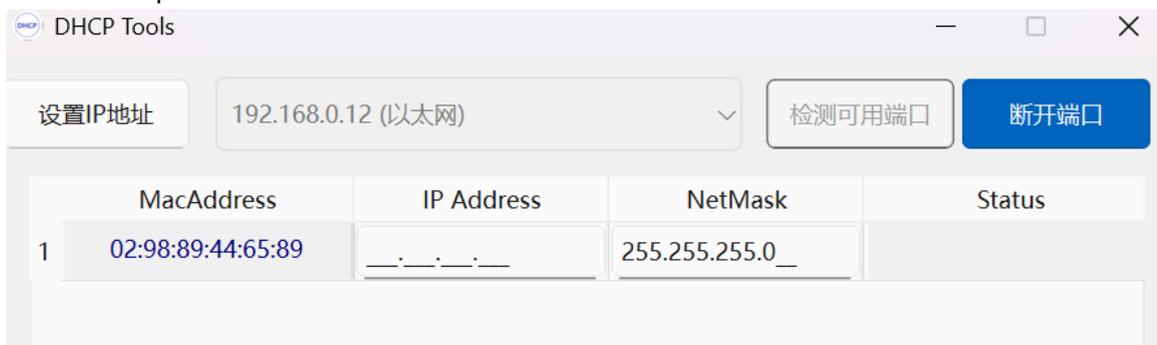
This means “**DHCP mode: always waits for IP assignment after power-up.**”

After changing the switches, **power-cycle the module**. At this time, the module **BUS red LED will flash**.

Then click “**Detect available ports**”, select the local port, and click “**Connect port.**”



- ③ The scanned Sentinel Modbus/TCP IO-Link master will appear in the list below. The front field corresponds to the **MAC address**.



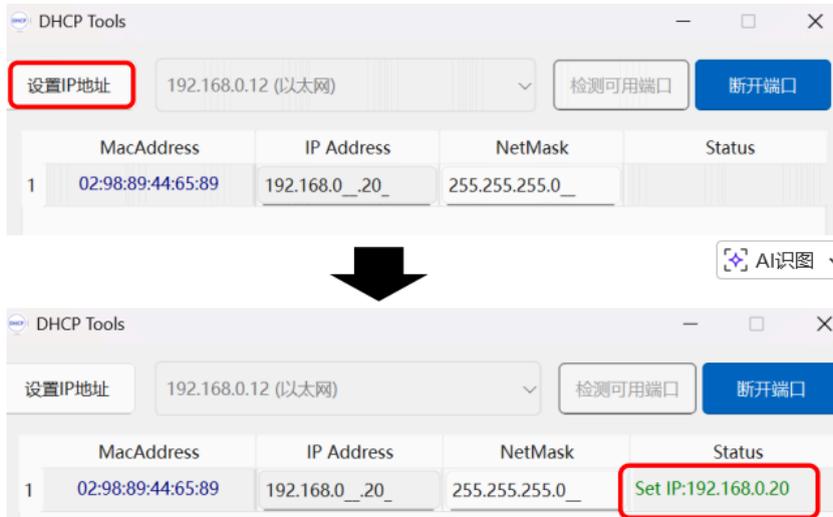
- ④ Enter the desired IP address under “IP Address”, e.g. **192.168.0.20**, then click “**Set IP Address**” (top-left).

When completed, a **green IP address** will be shown in the module **Status** column, indicating that the IP address has been set successfully. You may then close the software.

After the IP is set, the module **BUS red LED will stop flashing**. After setting, you can change the IP selector switches to:

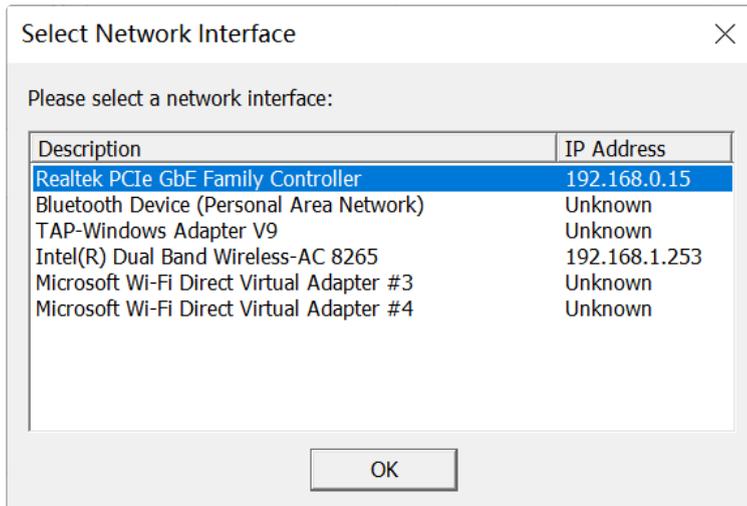
- **0x00: Run using the last DHCP–assigned IP address.** Each power cycle will use the previously assigned address (e.g. **192.168.0.20**).
- **0x01 – 0xFE:** The **first three octets (subnet)** follow the DHCP assignment, and the **4th octet** follows the switch value.

Example: if **ADDR_H = 1** and **ADDR_L = 4**, then the last octet is **0x14 = 20**, so the IP becomes **192.168.0.20**.

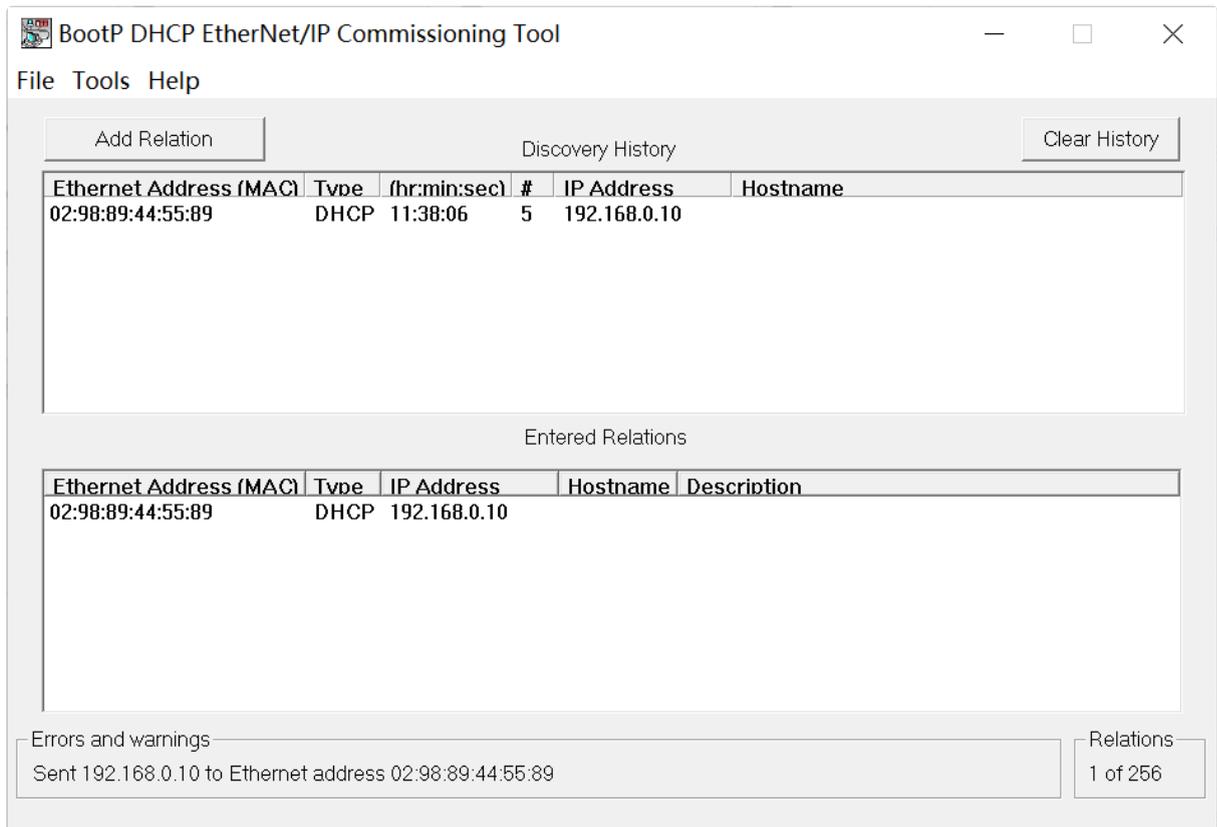
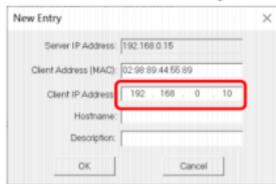
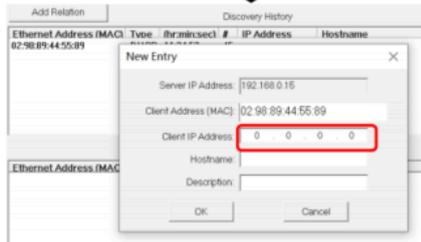
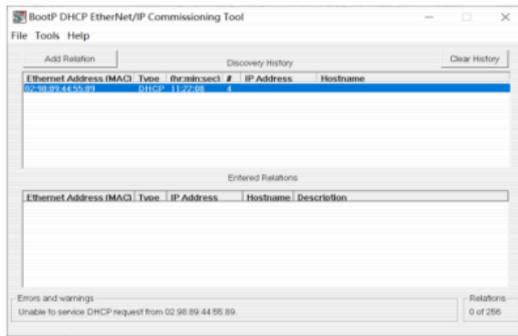


Method 2: Set via Third–Party IP Assignment Software

- ① In this example, the AB (Allen–Bradley) “**Bootp–DHCP Tool**” is used. Before setting, set the module IP selector switches (**ADDR_H / ADDR_L**) to **0xFF**, meaning “**DHCP mode: always waits for IP assignment after power–up.**” After changing the switches, **power–cycle the module**. The module **BUS red LED will flash**. Launch the tool, select the correct local network adapter, then click “**OK.**”

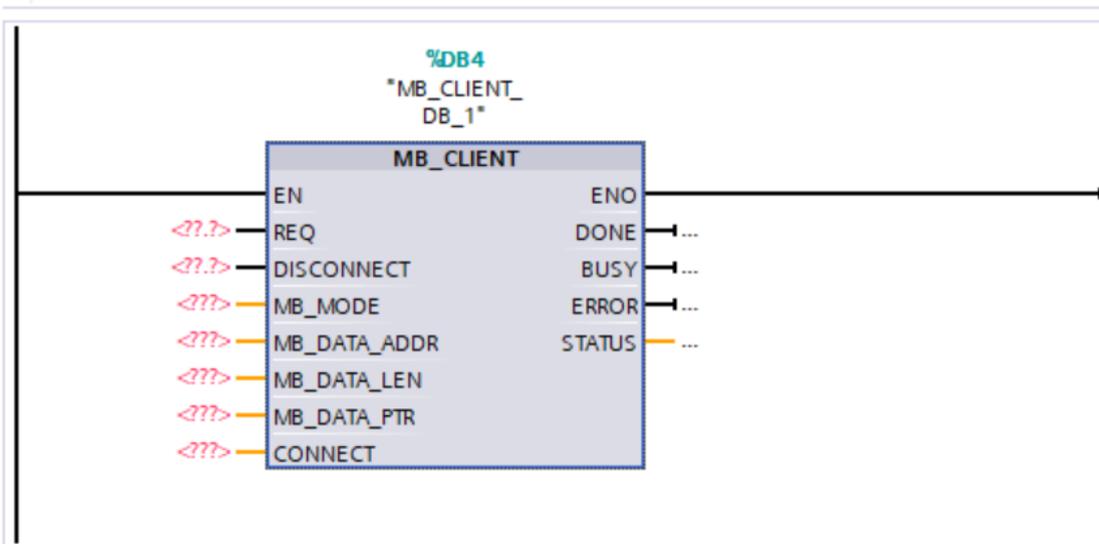


- ② Double–click the scanned module, enter the IP address to be assigned (**the module IP must be in the same subnet as the PC IP**), then click “**OK.**” After successful assignment, the module **BUS red LED will stop flashing**. You may then set the selector switches to:
 - **0x00:** run using the last DHCP–assigned IP address after each power cycle
 - **0x01 – 0xFE:** first three octets follow DHCP assignment; last octet follows the switch value

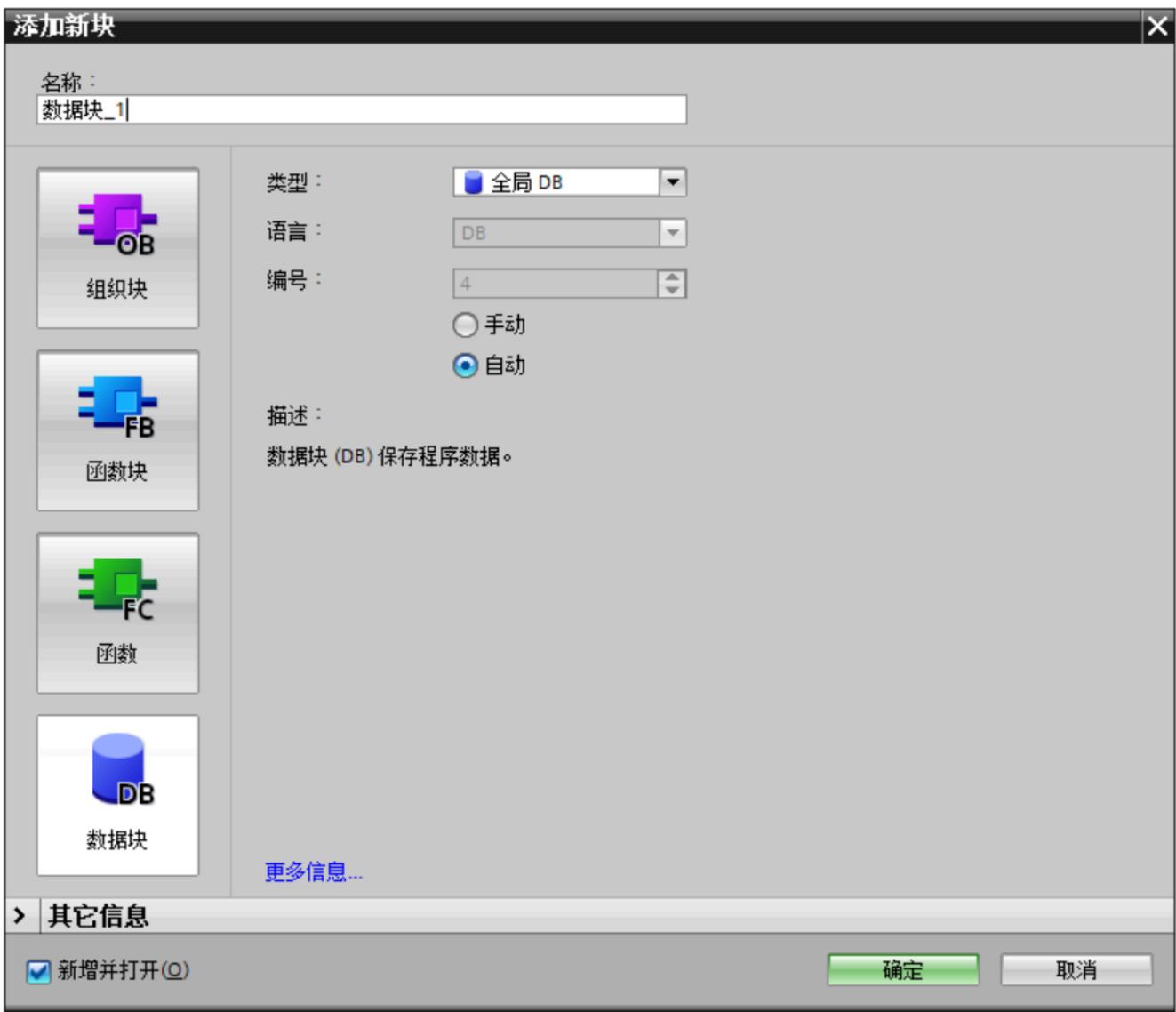


2) Open TIA Portal, Create a New Project, and Configure the PLC (steps omitted)

Open the main program **OB**, and call the **MB_CLIENT** instruction block. When inserted, it will automatically generate the associated instance data block.



3) Create a New Global Data Block: DB2



4) Open DB2 and Create the Connection Structure

Double-click to open **DB2**, create a custom variable named **"SSS"**, and set its data type to **"TCON_IP_v4"** (you can type it directly in the field). The structure will be created automatically.

名称	数据类型	起始值	保
Static			
sss	TCON_IP_v4		
Interfaceld	HW_ANY	0	
ID	CONN_OUC	16#0	
ConnectionType	Byte	16#0B	
ActiveEstablished	Bool	false	
RemoteAddress	IP_V4		
ADDR	Array[1..4] of Byte		
ADDR[1]	Byte	16#0	
ADDR[2]	Byte	16#0	
ADDR[3]	Byte	16#0	
ADDR[4]	Byte	16#0	
RemotePort	UInt	0	
LocalPort	UInt	0	
<新增>			

Parameter definitions are as follows:

参数	说明
Interfaceld	网口硬件标识符，对于本体网口为 64，即 16#40。
ID	连接 ID，取值范围 1~4095
Connection Type	连接类型。TCP 连接默认为：16#0B
ActiveEstablished	建立连接。主动为 1（客户端），被动为 0（服务器）。
ADDR	服务器侧的 IP 地址
RemotePort	远程端口号
LocalPort	本地端口号

5) Configure the Connection Parameters (Example)

In this example, the IO-Link master IP is **192.168.0.10**, the remote port is **502**, **ID = 1**, and **ActiveEstablished = 1**.

LocalPORT typically uses the default value **0**, meaning the local side uses a **random port**.

Set the fields in the structure as follows:

名称	数据类型	起始值	保持
Static			<input type="checkbox"/>
555	TCON_IP_v4		<input type="checkbox"/>
Interfaceld	HW_ANY	16#40	<input type="checkbox"/>
ID	CONN_OUC	1	<input type="checkbox"/>
ConnectionType	Byte	16#0B	<input type="checkbox"/>
ActiveEstablished	Bool	1	<input type="checkbox"/>
RemoteAddress	IP_V4		<input type="checkbox"/>
ADDR	Array[1..4] of Byte		<input type="checkbox"/>
ADDR[1]	Byte	16#C0	<input type="checkbox"/>
ADDR[2]	Byte	16#A8	<input type="checkbox"/>
ADDR[3]	Byte	0	<input type="checkbox"/>
ADDR[4]	Byte	16#A	<input type="checkbox"/>
RemotePort	UInt	502	<input type="checkbox"/>
LocalPort	UInt	0	<input type="checkbox"/>

6) Create a New Global Data Block DB3 (Data Buffer)

Create **DB3** to store buffered data. In **DB3**, create a **Word array** type to hold the data.

名称	数据类型	偏移量	起始值	保持
Static				<input type="checkbox"/>
IN1	Array[0..132...]	0.0		<input type="checkbox"/>
IN1[0]	Word	0.0	16#0	<input type="checkbox"/>
IN1[1]	Word	2.0	16#0	<input type="checkbox"/>
IN1[2]	Word	4.0	16#0	<input type="checkbox"/>
IN1[3]	Word	6.0	16#0	<input type="checkbox"/>
IN1[4]	Word	8.0	16#0	<input type="checkbox"/>
IN1[5]	Word	10.0	16#0	<input type="checkbox"/>
IN1[6]	Word	12.0	16#0	<input type="checkbox"/>
IN1[7]	Word	14.0	16#0	<input type="checkbox"/>
IN1[8]	Word	16.0	16#0	<input type="checkbox"/>
IN1[9]	Word	18.0	16#0	<input type="checkbox"/>
IN1[10]	Word	20.0	16#0	<input type="checkbox"/>
IN1[11]	Word	22.0	16#0	<input type="checkbox"/>
IN1[12]	Word	24.0	16#0	<input type="checkbox"/>

7) Notes on the MB_CLIENT “MB_DATA_PTR” Buffer Type

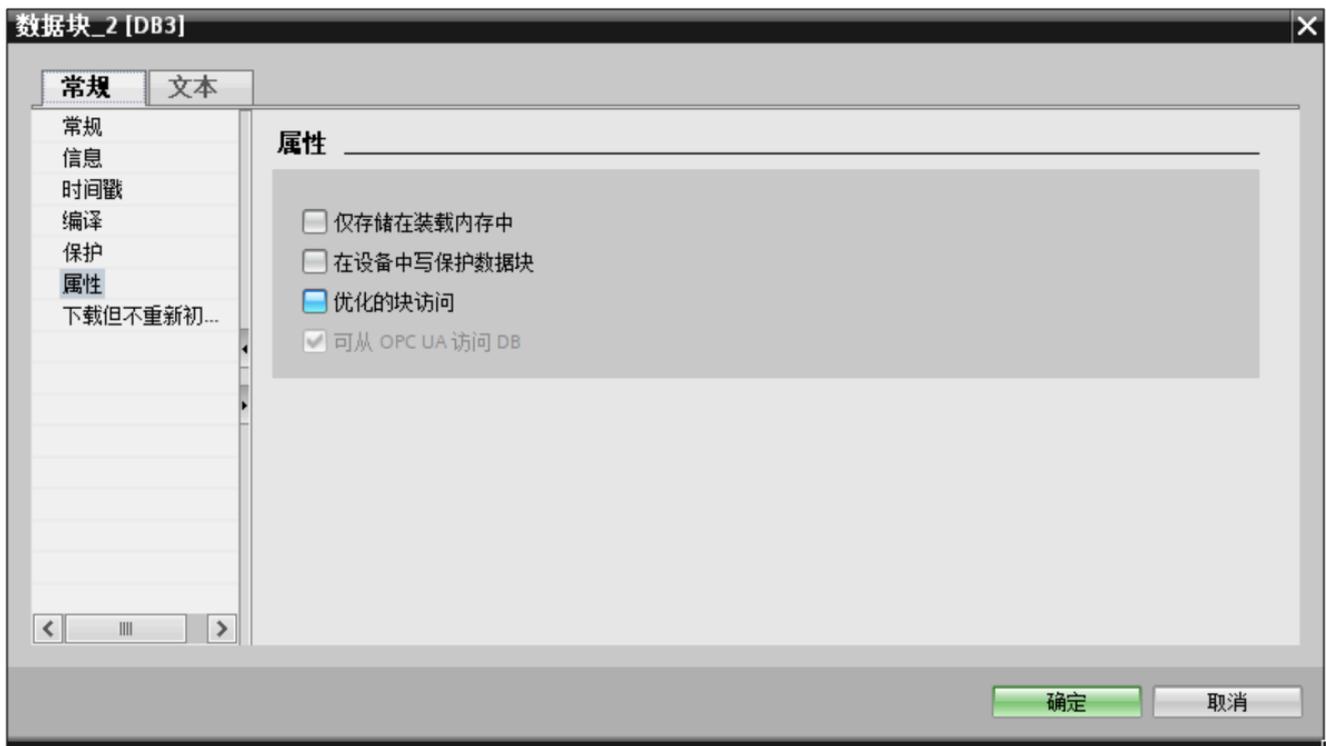
The buffer specified by **MB_DATA_PTR** in **MB_CLIENT** can be:

- A **DB block**, or
- A memory area (e.g. **M**)

If using an **optimized DB**, it must be an array of **basic data types**, such as **INT / REAL / BOOL arrays**, and you must assign **MB_DATA_PTR** using **symbolic addressing**.

If using a **standard (non-optimized) DB**, you may assign **MB_DATA_PTR** using a **pointer**.

This example uses a **standard DB** structure. You can disable optimized access by right-clicking **DB3** → **Properties** → **uncheck “Optimized block access”**, as shown:



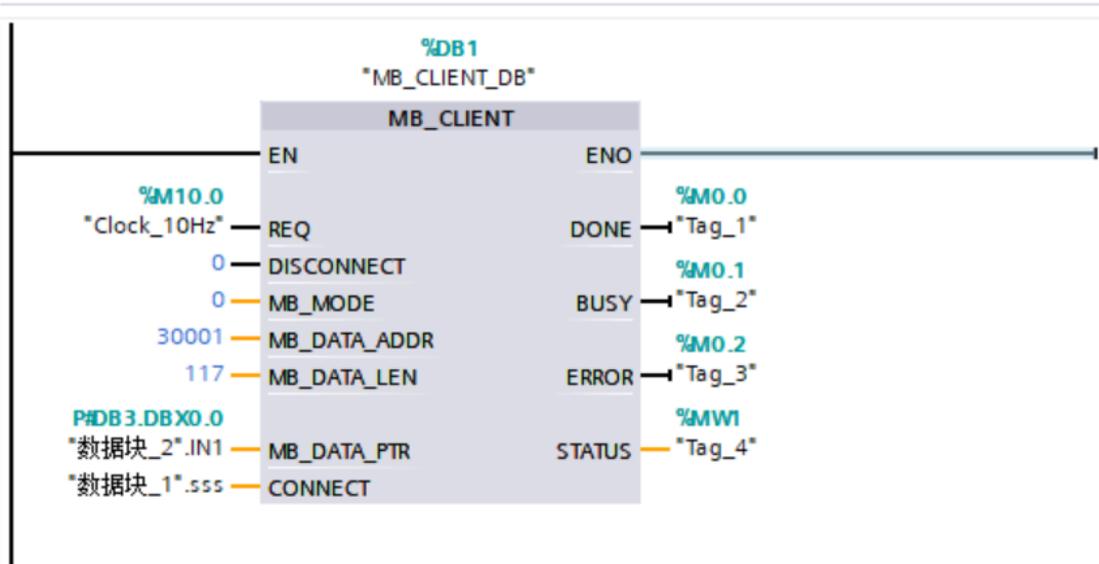
8) Configure MB_CLIENT Parameters and Execute Read (Example)

Return to the **MB_CLIENT** block and fill in the parameters according to the MB_CLIENT parameter table.

Use the CPU clock pulse to trigger the first **MB_CLIENT** block **REQ** every **0.01 s** to perform a read operation:

- Use **F04** (Read Input Registers)
- **MB_MODE = 0** (read input words)
- **MB_DATA_ADDR = 30001** (start address)
- **MB_DATA_LEN = read length** (range **1–125**)
- **MB_DATA_PTR** points to the buffer to receive data from the Modbus server (or the buffer to send data to the server)
- **CONNECT** points to the connection description structure

Example wiring/parameter setting:



9) Verify Data After Download

After downloading the program, you can read input data from the DB block specified by **MB_DATA_PTR** (DB3 in this example).

Example: the first byte **16#FF** indicates the current 8–port IO–Link status, meaning **all 8 IO–Link devices are communicating normally**.

名称	数据类型	偏移量	起始值	监视值	保持	可从 HMI/...
Static		0.0			<input type="checkbox"/>	<input type="checkbox"/>
IN1	Array[0..132] of W...	0.0			<input type="checkbox"/>	<input checked="" type="checkbox"/>
IN1[0]	Word	0.0	16#0	16#FF00	<input type="checkbox"/>	<input checked="" type="checkbox"/>
IN1[1]	Word	2.0	16#0	16#0000	<input type="checkbox"/>	<input checked="" type="checkbox"/>
IN1[2]	Word	4.0	16#0	16#0000	<input type="checkbox"/>	<input checked="" type="checkbox"/>

10) Reading/Writing the Full Data Area Requires Multiple MB_CLIENT Calls

According to the Sentinel Modbus/TCP IO–Link master manual:

- **Input** occupies **Word[0]–Word[132]** (133 words total)
Since the maximum read length is **125**, one MB_CLIENT cannot read all input words in a single call.
- **Output** occupies **Word[0]–Word[127]** (128 words total)
Since the maximum write length is **123** (per the instruction limit in this context), one MB_CLIENT cannot write all output words in a single call.

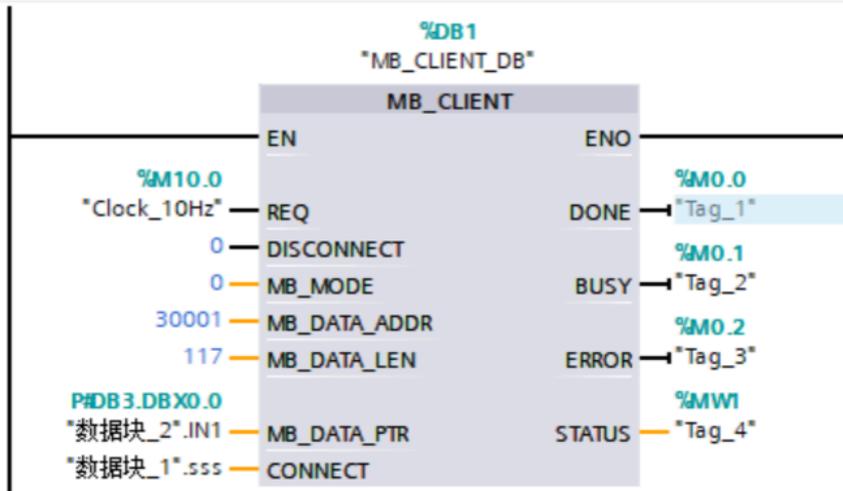
Therefore, to read all input data and write all output data, you can:

- Read input in **two MB_CLIENT calls**
- Write output in **two MB_CLIENT calls**

Note: In this example, all four calls use the same block **DB1**, and the **DONE** output of the first block is used as the **REQ** trigger of the second block, and so on.

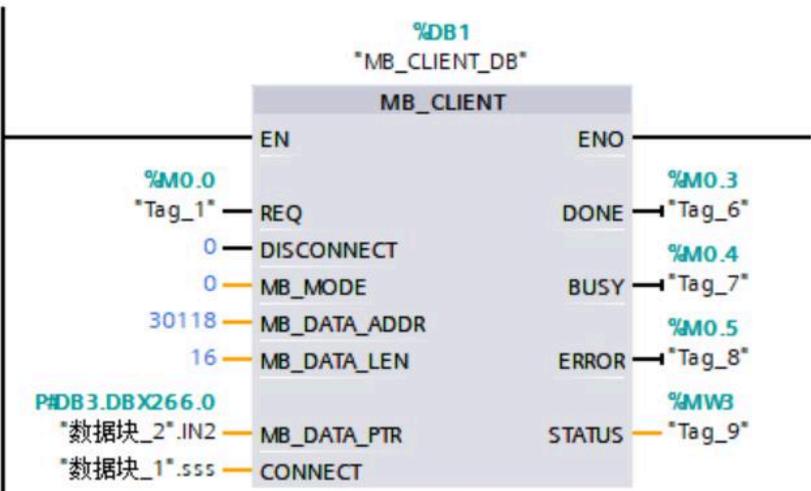
▼ 程序段 1：读第8个端口之前的数据

注释



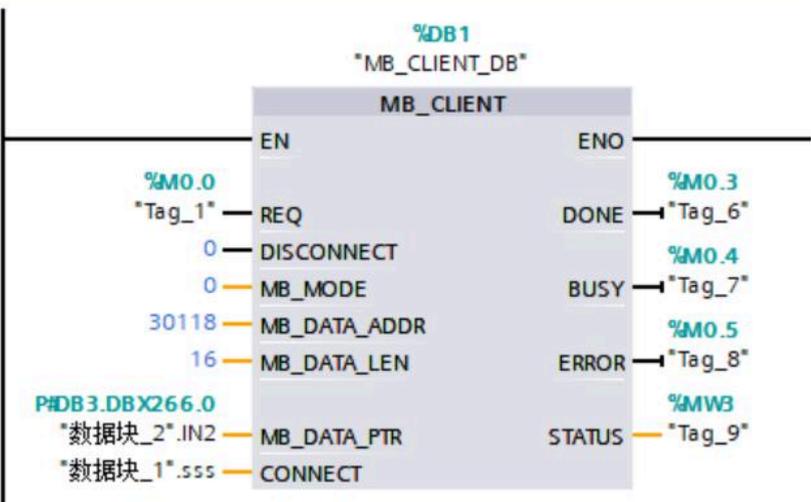
▼ 程序段 2：读第8个端口数据

注释



▼ 程序段 2：读第8个端口数据

注释



11) Detailed Settings for the Four MB_CLIENT Calls

1. MB_CLIENT #1 (Read Input – Part 1)

- Function code: **F04**
- **MB_MODE = 0** (Read input words)
- **MB_DATA_ADDR = 30001**
- **MB_DATA_LEN = 117** (covers all input data up to before port 8)
- **MB_DATA_PTR → DB3.IN1**

2. **MB_CLIENT #2 (Read Input – Part 2)**

- Function code: **F04**
- **MB_MODE = 0**
- **MB_DATA_ADDR = 30118**
- **MB_DATA_LEN = 16** (input data for port 8)
- **MB_DATA_PTR → DB3.IN2**

3. **MB_CLIENT #3 (Write Output – Part 1)**

- Function code: **F16** (Write Multiple Registers)
- **MB_MODE = 2** (Write holding registers)
- **MB_DATA_ADDR = 40001**
- **MB_DATA_LEN = 112** (all output data before port 8)
- **MB_DATA_PTR → DB3.OUT1**

4. **MB_CLIENT #4 (Write Output – Part 2)**

- Function code: **F16**
- **MB_MODE = 2**
- **MB_DATA_ADDR = 40113**
- **MB_DATA_LEN = 16** (output data for port 8)
- **MB_DATA_PTR → DB3.OUT2**

Execution sequence:

After the TCP connection is established, use a CPU clock pulse to trigger **MB_CLIENT #1 REQ** every **0.01 s** to read. Then:

- Use **DONE** of MB_CLIENT #1 to trigger **REQ** of MB_CLIENT #2 (read)
- Use **DONE** of MB_CLIENT #2 to trigger **REQ** of MB_CLIENT #3 (write)
- Use **DONE** of MB_CLIENT #3 to trigger **REQ** of MB_CLIENT #4 (write)

Note: This example triggers the full chain every **0.01 s**. In real projects, adjust the trigger interval based on the number of read/write operations and the data volume to ensure each operation

completes reliably.



Appendix

1) IO-Link Process Data Input (133 Words)

Supported function code: **F04 (Read Input Registers)**

Modbus引用编号 WORD	Modbus数据地址 WORD	IO-LINK字节 BYTE	描述																		
30001	0	Byte0	8位代表8个端口当前IO-LINK状态: 1正常通信, 0未通信 <table border="1"> <tr> <td>位</td> <td>7</td> <td>6</td> <td>5</td> <td>4</td> <td>3</td> <td>2</td> <td>1</td> <td>0</td> </tr> <tr> <td>端口</td> <td>C8</td> <td>C7</td> <td>C6</td> <td>C5</td> <td>C4</td> <td>C3</td> <td>C2</td> <td>C1</td> </tr> </table>	位	7	6	5	4	3	2	1	0	端口	C8	C7	C6	C5	C4	C3	C2	C1
		位	7	6	5	4	3	2	1	0											
端口	C8	C7	C6	C5	C4	C3	C2	C1													
Byte1	8位代表8个端口IO-LINK断线记录: 1有过断线, 0未有过断线 <table border="1"> <tr> <td>位</td> <td>7</td> <td>6</td> <td>5</td> <td>4</td> <td>3</td> <td>2</td> <td>1</td> <td>0</td> </tr> <tr> <td>端口</td> <td>C8</td> <td>C7</td> <td>C6</td> <td>C5</td> <td>C4</td> <td>C3</td> <td>C2</td> <td>C1</td> </tr> </table>	位	7	6	5	4	3	2	1	0	端口	C8	C7	C6	C5	C4	C3	C2	C1		
位	7	6	5	4	3	2	1	0													
端口	C8	C7	C6	C5	C4	C3	C2	C1													
30002	1	Byte2	C1端口断线次数																		
		Byte3	C2端口断线次数																		
30003	2	Byte4	C3端口断线次数																		
		Byte5	C4端口断线次数																		
30004	3	Byte6	C5端口断线次数																		
		Byte7	C6端口断线次数																		
30005	4	Byte8	C7端口断线次数																		
		Byte9	C8端口断线次数																		
30006 - 30021	5 - 20	Byte10 - Byte41	C1端口过程输入数据 (32Byte)																		
30022 - 30037	21 - 36	Byte42 - Byte73	C2端口过程输入数据 (32Byte)																		
30038 - 30053	37 - 52	Byte74 - Byte105	C3端口过程输入数据 (32Byte)																		
30054 - 30069	53 - 68	Byte106 - Byte137	C4端口过程输入数据 (32Byte)																		
30070 - 30085	69 - 84	Byte138 - Byte169	C5端口过程输入数据 (32Byte)																		
30086 - 30101	85 - 100	Byte170 - Byte201	C6端口过程输入数据 (32Byte)																		
30102 - 30117	101 - 116	Byte202 - Byte233	C7端口过程输入数据 (32Byte)																		
30118 - 30133	117 - 132	Byte234 - Byte265	C8端口过程输入数据 (32Byte)																		

2) IO-Link Process Data Output (128 Words)

Supported function codes:

- **F03** (Read Holding Registers)
- **F16** (Write Multiple Holding Registers)

- **F23** (Read/Write Multiple Holding Registers)

Modbus引用编号 WORD	Modbus数据地址 WORD	IO-LINK字节 BYTE	描述
40001 - 40016	0 - 15	Byte0 - Byte31	C1端口过程输出数据 (32Byte)
40017 - 40032	16 - 31	Byte32 - Byte63	C2端口过程输出数据 (32Byte)
40033 - 40048	32 - 47	Byte64 - Byte95	C3端口过程输出数据 (32Byte)
40049 - 40064	48 - 63	Byte96 - Byte127	C4端口过程输出数据 (32Byte)
40065 - 40080	64 - 79	Byte128 - Byte159	C5端口过程输出数据 (32Byte)
40081 - 40096	80 - 95	Byte160 - Byte191	C6端口过程输出数据 (32Byte)
40097 - 40112	96 - 111	Byte192 - Byte223	C7端口过程输出数据 (32Byte)
40113 - 40128	112 - 127	Byte224 - Byte255	C8端口过程输出数据 (32Byte)