

# 西门子 1200PLC 组态森特奈 ModbusTCP 协议

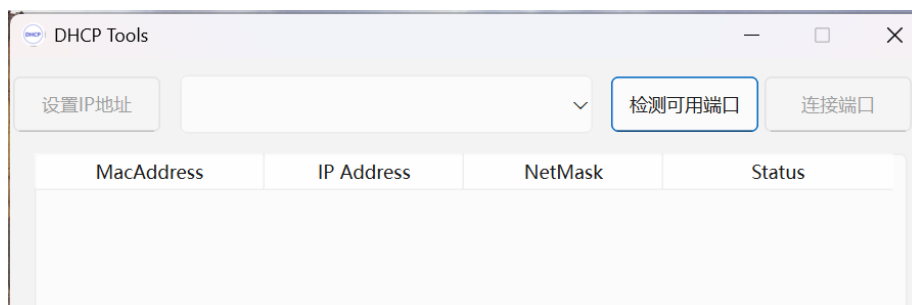


## IO-LINK 主站模块使用教程

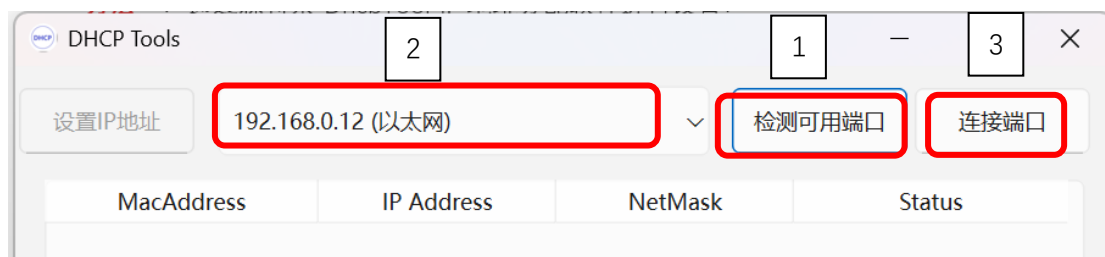
- 1、设定森特奈 ModbusTCP 协议 IO-LINK 主站模块的 IP 地址。

**方法一：**通过森特奈“DhcpTool”IP 地址分配软件进行设置：

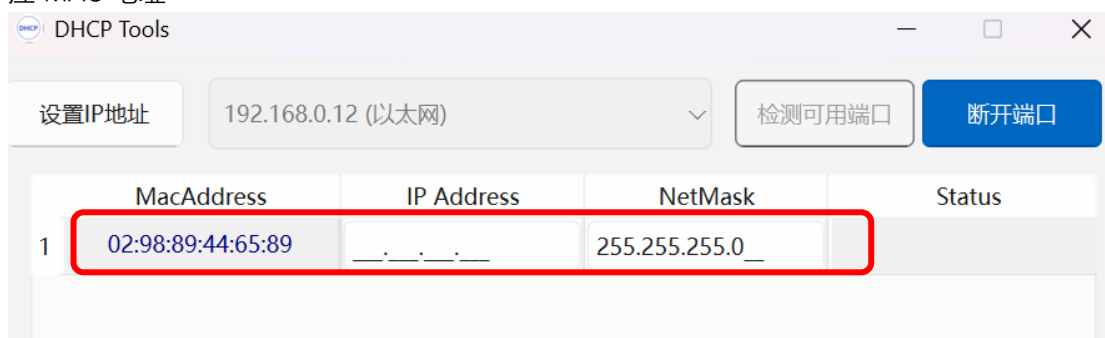
- ①、首先可在森特奈官网获取“DhcpTool V1.2”软件并安装(仅支持 WIN10 及以上系统)，打开软件



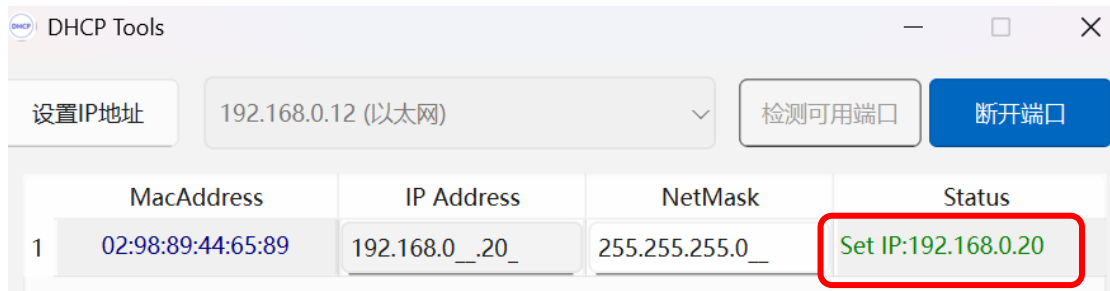
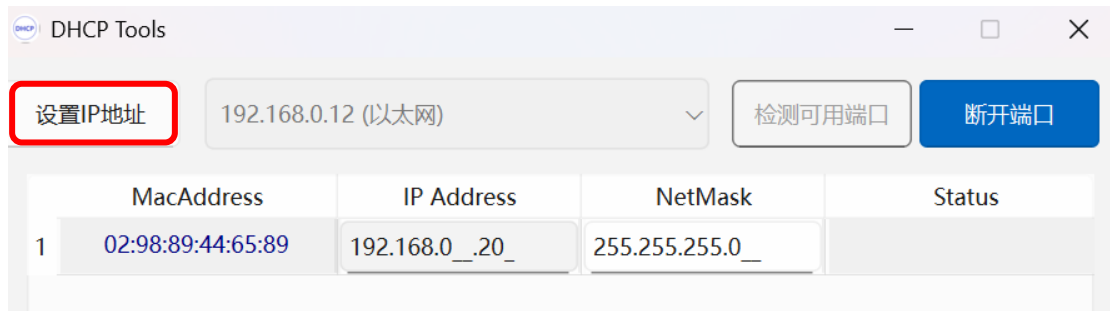
- ②、将森特奈 ModbusTCP 协议 IO-LINK 主站模块与电脑连接，然后将模块 IP 地址设置拨码拨到“0XFF”（窗口右侧 ADDR\_H 拨到 F, ADDR\_L 拨到 F, 可参考说明书），即“DHCP 模式，上电一直等待分配 IP”。拨完后，模块需要重新上电，此时，模块 BUS 红灯将闪烁。然后点击软件中“检测可用端口”，找到本地端口，然后点击“连接端口”



- ③、此时，下方将显示扫描出的森特奈 ModbusTCP 协议的 IO-LINK 主站模块，前面对应 MAC 地址

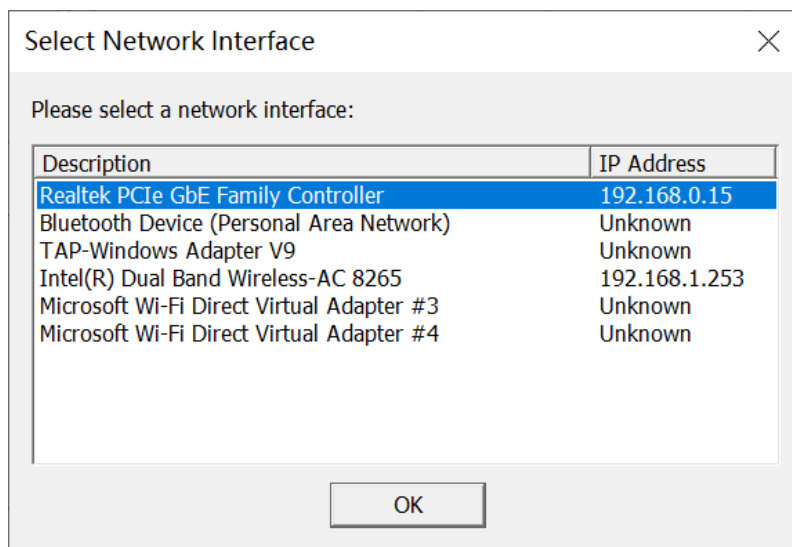


- ④、在“IP Address”下方填入 IP 地址，如“192.168.0.20”，然后点击左上角“设置 IP 地址”，然后在模块后面“Status”下方将显示绿色 IP 地址，表示模块 IP 地址已经设置完成，此时可以关闭软件。设置完毕后，模块 BUS 红灯将不再闪烁。可以将模块 IP 地址设置拨码拨到“0X00”，即“按照上次的 DHCP 分配的 IP 地址运行”，那么每次断电上电后，模块都将按之前分配的地址运行（即 192.168.0.20）。另外拨码也可以拨到 0x01 - 0xFE，此时前 3 位网段，按照 DHCP 分配的运行，第 4 位按照该拨码设置运行（比如，将 ADDR\_H 拨到 1，ADDR\_L 拨到 4，那么模块 IP 地址也是 192.168.0.20）。



**方法二：** 可以通过第三方设置软件进行 IP 地址设置。

①、本例中使用 AB 的“Bootp-DHCP Tool”软件，设置之前，先将模块 IP 地址设置拨码（窗口右侧 ADDR\_H，ADDR\_L，可参考说明书）拨到“0xFF”，即“DHCP 模式，上电一直等待分配 IP”。拨完后，模块需要重新上电，此时，模块 BUS 红灯将闪烁。打开该软件，选择本机相应网卡，点击“OK”。



②、双击扫描出的模块，输入要设置的 IP 地址（IP 地址与本机 IP 地址要在同一网段），点击“OK”，设置完毕后，模块 BUS 红灯将不再闪烁。可以将模块 IP 地址设置拨码拨到“0x00”，即“按照上次的 DHCP 分配的 IP 地址运行”，那么每次断电上电后，模块都将按之前分配的地址运行。另外拨码也可以拨到 0x01 - 0xFE，此时前 3 位网段，按照 DHCP 分配的运行，第 4 位按照该拨码设置运行。

BootP DHCP EtherNet/IP Commissioning Tool

File Tools Help

Add Relation Discovery History Clear History

Ethernet Address (MAC)	Type	(hr:min:sec)	#	IP Address	Hostname
02:98:89:44:55:89	DHCP	11:22:08	4		

Entered Relations

Ethernet Address (MAC)	Type	IP Address	Hostname	Description
------------------------	------	------------	----------	-------------

Errors and warnings

Unable to service DHCP request from 02:98:89:44:55:89.

Relations 0 of 256



Add Relation Discovery History

Ethernet Address (MAC)	Type	(hr:min:sec)	#	IP Address	Hostname
02:98:89:44:55:89	DHCP	11:22:08	4		

New Entry

Server IP Address: 192.168.0.15

Client Address (MAC): 02:98:89:44:55:89

Client IP Address: 0 . 0 . 0 . 0

Hostname:

Description:

OK Cancel



New Entry

Server IP Address: 192.168.0.15

Client Address (MAC): 02:98:89:44:55:89

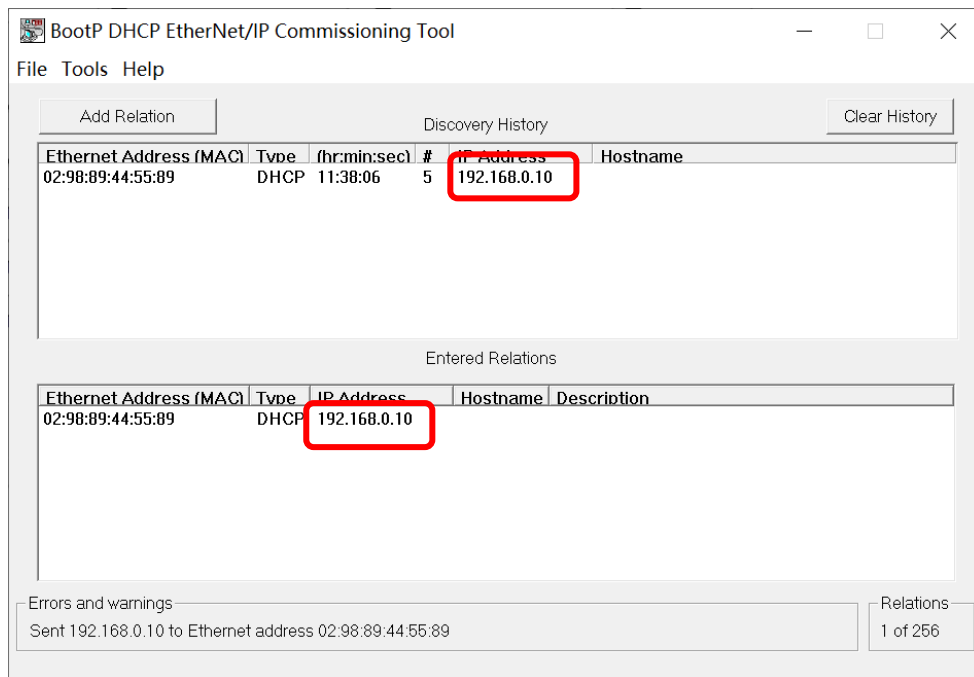
Client IP Address: 192 . 168 . 0 . 10

Hostname:

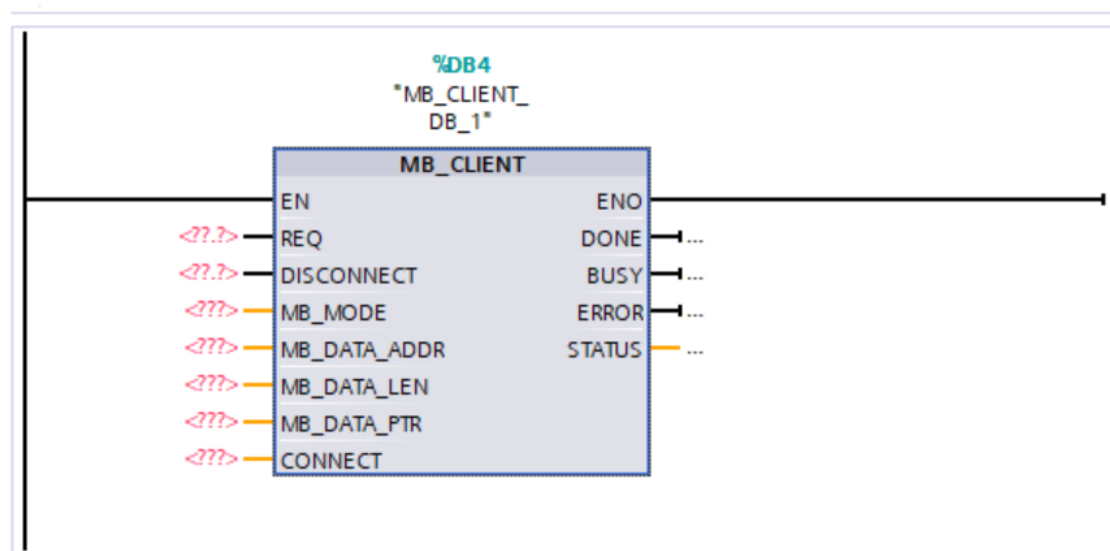
Description:

OK Cancel





2、打开博途软件，新建一个工程，组态 PLC（过程略）。打开主程序 OB，调用“MB\_CLIENT”指令块，调用时会自动生成背景数据块



3、创建一个新的全局数据块 DB2

名称：

数据块\_1

OB

组织块

FB

函数块

FC

函数

DB

数据块

类型：

全局 DB

语言：

DB

编号：

4

手动

自动

描述：

数据块 (DB) 保存程序数据。

[更多信息...](#)

其它信息

新增并打开(O)

确定

取消

4、双击打开 BD2，自定义变量名称为“SSS”，数据类型为“TCON\_IP\_v4”（可在该对话框中直接输入），该数据类型结构创建完毕。

	名称	数据类型	起始值	保
	Static			
	SSS	TCON_IP_v4		
	InterfaceId	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	
0	ADDR[2]	Byte	16#0	
1	ADDR[3]	Byte	16#0	
2	ADDR[4]	Byte	16#0	
3	RemotePort	UInt	0	
4	LocalPort	UInt	0	
5	<新增>			

各参数定义说明如下

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

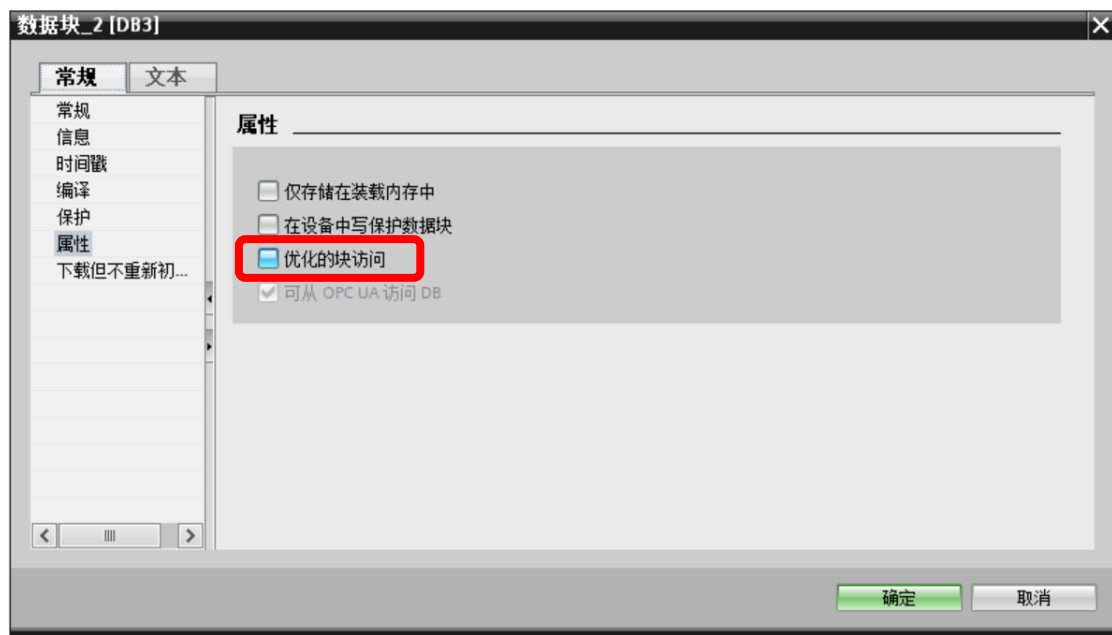
5、本例中 IO-LINK 主站模块 IP 地址为 192.168.0.10，远程端口号设为 502，ID 设为 1，ActiveEstablished 为 1，LocalPORT 一般使用默认值 0，意思是本地使用随机端口。所以客户端侧，该数据结构的各项值设置如下：

	名称	数据类型	起始值	保持
	Static			<input type="checkbox"/>
	SSS	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"/>
0	ADDR[2]	Byte	16#A8	<input type="checkbox"/>
1	ADDR[3]	Byte	0	<input type="checkbox"/>
2	ADDR[4]	Byte	16#A	<input type="checkbox"/>
3	RemotePort	UInt	502	<input type="checkbox"/>
4	LocalPort	UInt	0	<input type="checkbox"/>

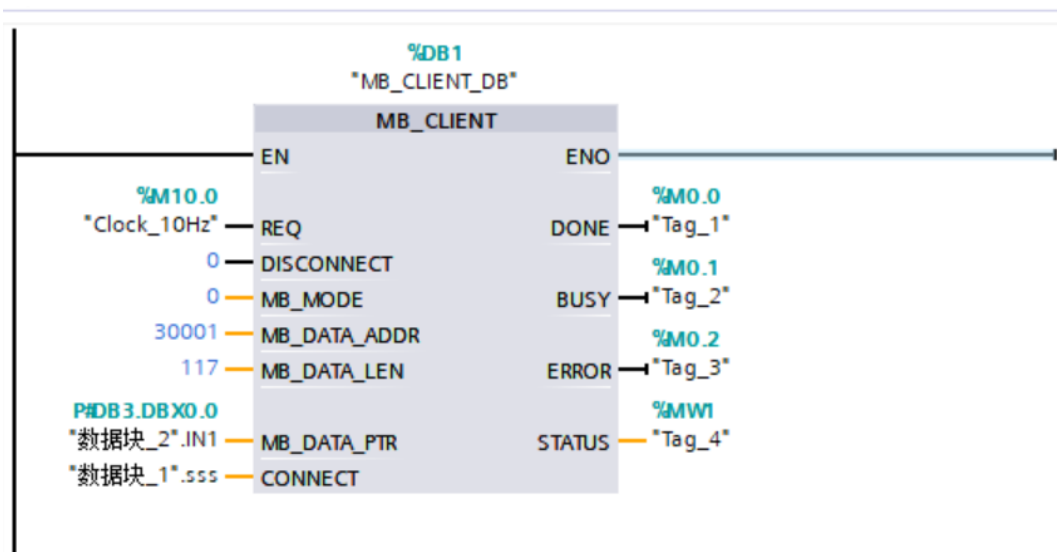
6、再创建一个新的全局数据块 DB3，用于存放缓存数据。在 DB3 内新建一个 Word 数组的数据类型，以便存放数据

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

7、“MB\_CLIENT”指令块中的“MB\_DATA\_PTR”所指定的数据缓冲区可以为 DB 块或者存储区域 M 地址, DB 块可以是优化的数据块, 也可以是标准的数据块结构, 若为优化的数据块结构, 必须是基本数据类型数组, 如 int、Real、BOOL、数组, 编程时需要以符号寻址的方式填写该引脚; 若为标准的数据块结构, 可以以指针的方式填写该引脚。本例中使用标准的数据块结构 (可以右键点击 DB3, 在“属性”中将“优化的块访问”前面的√去掉, 如图)



8、回到“MB\_CLIENT”指令块, 参考“MB\_CLIENT”指令的参数表, 依次填入各项参数, 利用 CPU 的时钟脉冲, 每隔 0.01s 触发一次第一个 MB\_Client 指令的 REQ 进行读操作, 使用 F04 功能码, MB\_MODE 为 0, 读输入字, MB\_DATA\_ADDR 为 30001 开始, MB\_DATA\_LEN 为读取长度 (长度范围 1-125), MB\_DATA\_PTR 指向待从 Modbus 服务器接收数据的数据缓冲区或指向待发送到 Modbus 服务器的数据所在数据缓冲区的指针。CONNECT 为指向连接描述结构的指针, 如下

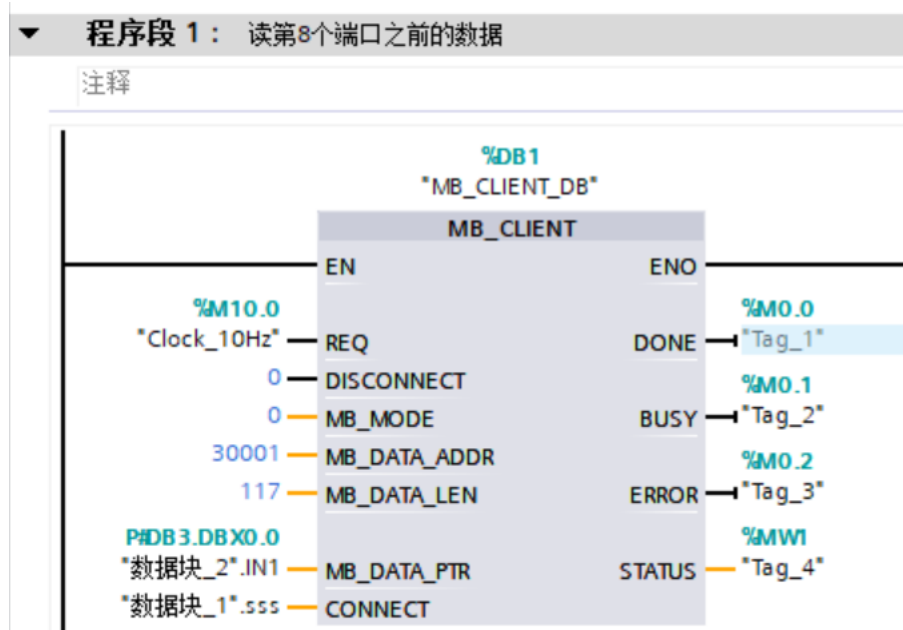


9、将程序下载后，就可以在 MB\_DATA\_PTR 引脚指定的 DB 块（DB3）变量中读取到输入数据。如：第一个字节数据“16#FF”表示 8 端口 IO-LINK 当前状态，即 8 个子站均通讯正常

数据块_2								
	名称	数据类型	偏移量	起始值	监视值	保持	可从 HMI...	
1	Static							
2	IN1	Array[0..132] of W...	0.0					
3	IN1[0]	Word	0.0	16#0	16#FF00			
4	IN1[1]	Word	2.0	16#0	16#0000			
5	IN1[2]	Word	4.0	16#0	16#0000			

10、参考森特奈 ModbusTCP 协议 IO-Link 主站模块说明书，发现输入占用 Word[0]-Word[132]，一共 133 个字，对于读输入数据，由于最大长度为 125，所以一个“MB\_CLIENT”不能完全读出所有输入数据，同理输出占用 Word[0]-Word[127]，一共 128 个字，但是一个“MB\_CLIENT”最大长度是 123，也不能完全写入所有输出数据，所以为了完整读取输入和写入输出，此时可以分两次调用“MB\_CLIENT”读取和分两次调用“MB\_CLIENT”写入数据（模块的具体字节分配，请看文末附录）如：

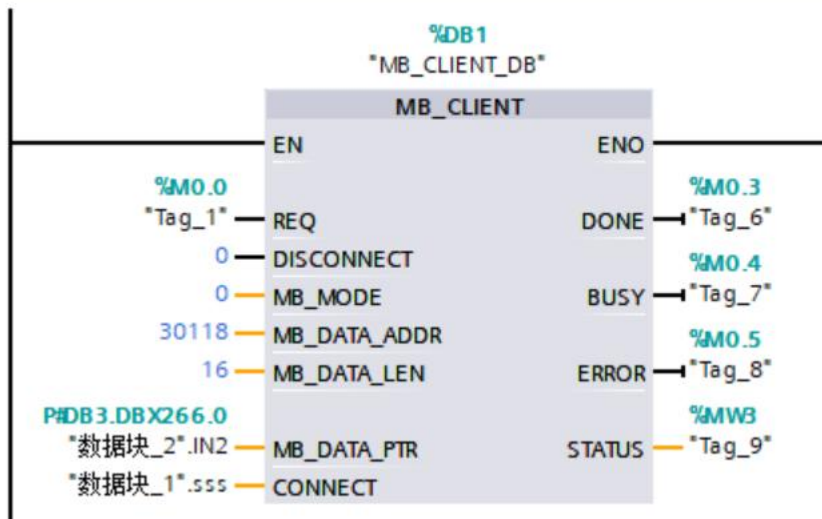
注：四次调用，用的是同一个块 DB1，同时第一个块的“DONE”作为第二个块的“REQ”，以此类推





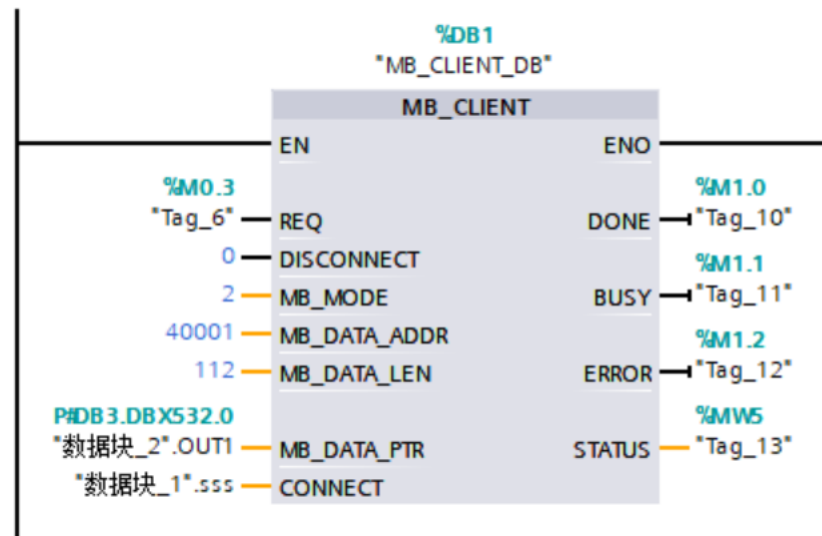
## ▼ 程序段 2： 读第8个端口数据

注释

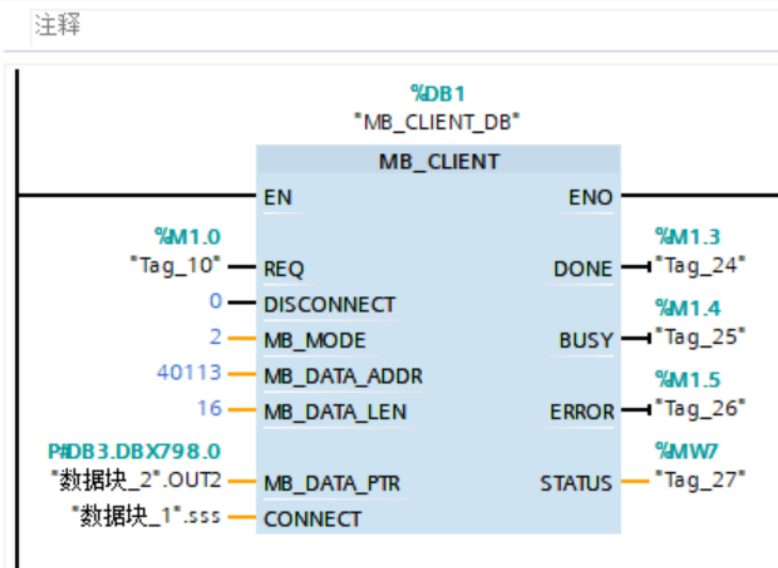


## ▼ 程序段 3： 写第8个端口之前的数据

注释



▼ 程序段 4：写第8个端口数据



11、第一个 MB\_Client 指令，使用 F04 功能码，MB\_MODE 为 0，读输入字，MB\_DATA\_ADDR 为 30001 开始，读取长度 MB\_DATA\_LEN 为 117，正好是第 8 个端口之前的所有输入数据，MB\_DATA\_PTR 指向 DB3 中的“IN1”（为了读写数据，已经在 DB3 中新建 IN1、IN2、OUT1、OUT2 用于读写数据，数据缓冲区必须足够大。）

第二个 MB\_Client 指令，也使用 F04 功能码，MB\_MODE 为 0，读输入字，MB\_DATA\_ADDR 为 30118 开始，读取长度 MB\_DATA\_LEN 为 16，是第 8 个端口的输入数据，MB\_DATA\_PTR 指向 DB3 中的“IN2”

第三个 MB\_Client 指令，使用 F16 功能码，MB\_MODE 为 2，写保持寄存器，MB\_DATA\_ADDR 为 40001 开始，写入长度 MB\_DATA\_LEN 为 112，是第 8 个端口之前的所有输出数据，MB\_DATA\_PTR 指向 DB3 中的“OUT1”

第四个 MB\_Client 指令，使用 F16 功能码，MB\_MODE 为 2，写保持寄存器，MB\_DATA\_ADDR 为 40113 开始，写入长度 MB\_DATA\_LEN 为 16，是第 8 个端口的输出数据，MB\_DATA\_PTR 指向 DB3 中的“OUT2”

当 TCP 连接建立后，利用 CPU 的时钟脉冲，每隔 0.01s 触发一次第一个 MB\_Client 指令的 REQ 进行读操作，然后，利用其输出的 DONE 触发第二个 MB\_Client 指令的 REQ 进行读操作，再利用第二个 MB\_Client 输出的 DONE 触发第三个 MB\_Client 指令的 REQ 进行写操作，最后，利用第三个 MB\_Client 输出的 DONE 触发第四个 MB\_Client 指令的 REQ 进行写操作（注意：本例中为 0.01s 触发一次连续的操作，而在实际的项目中，请根据实际的读/写操作次数和访问数据量，调整触发间隔，保证读/写操作都能正常完成）

保持实际值 快照 将快照值复制到起始值中 将起始值加载为实际值

数据块\_2

	名称	数据类型	偏移量	起始值	保持	可从 HMI/...	从 H...	在 HMI ...
1	Static				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	IN1	Array[0..132] of W...	0.0		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3	IN2	Array[0..132] of W...	266.0		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4	OUT1	Array[0..132] of W...	532.0		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
5	OUT2	Array[0..132] of W...	798.0		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

附录

1、IO-LINK 过程数据输入（占用 133 Word）

支持功能码 F04（读输入寄存器）

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><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><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 过程数据输出（占用 128 Word）

支持功能码 F03（读保持寄存器）、F16（写保持寄存器）、F23（读、写保持寄存器）

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）