

S7-200 SMART 与 S7-1200 之间 TCP 通信— S7-200 SMART 作为客户端

TCP 协议通信

TCP 通信为面向连接的通信，需要双方都调用指令以建立连接及交换数据。S7-200 SMART 与 S7-1200 通过 TCP 通信，在 S7-1200 调用 T-block 指令 (TCON, TDISCON, TSEND, TRCV)，在 S7-200 SMART 调用 Open User Communication 库指令(TCP_CONNECT,DISCONNECT,TCP_SEND,TCP_RECV)。双方的发送和接收指令必须成对出现。

客户端：主动建立连接，可以理解为主站； 服务器：被动建立连接，可以理解为从站。

⚠ 注意：

S7-200 SMART 在 CPU 硬件固件及编程软件版本均升级到 V2.2 之后才开始支持开放式通信。编程软件版本低于V2.2，无 Open User Communication 库指令；硬件固件低于 V2.2，硬件不支持开放式通信协议。

S7-200 SMART TCP 连接资源：8个主动连接资源，8个被动连接资源

S7-200 SMART TCP 通信数据量：1024 字节

硬件和软件需求及所完成的通信任务

硬件：

- ① S7-1200 CPU (IP 地址 192.168.0.100；子网掩码 255.255.255.0)
- ② S7-200 SMART CPU (固件版本V2.2)(IP 地址 192.168.0.55；子网掩码 255.255.255.0)
- ③ PC (带以太网卡)
- ④ TP 以太网电缆

软件：

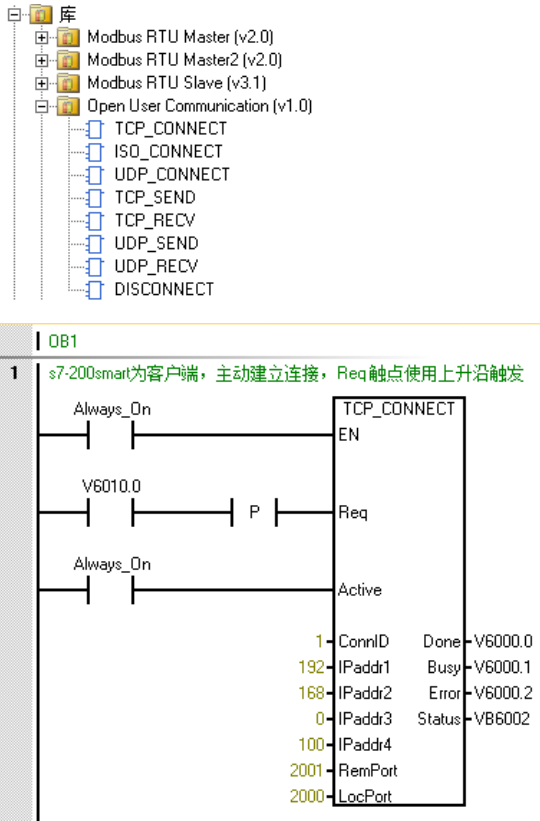
- ① STEP7 V13 SP1 Upd 9
- ② STEP 7 Micro/WIN SMART (软件版本 V2.2)

所完成的通信任务：

- ① S7-200 SMART 发送 10 个字节数据：(S7-200 SMART 侧) VB0~VB9 --> (S7-1200侧) DB4.DBB10~DB4.DBB19
- ② S7-200 SMART 接收 10 个字节数据：(S7-200 SMART 侧) VB2000~VB2009<-- (S7-1200 侧) DB4.DBB0~DB4.DBB9

S7-200 SMART侧编程 (客户机侧组态编程)

1.打开STEP 7 Micro/WIN SMART>项目树>指令树>库>Open User Communication ,调用TCP_CONNECT，如图1所示。

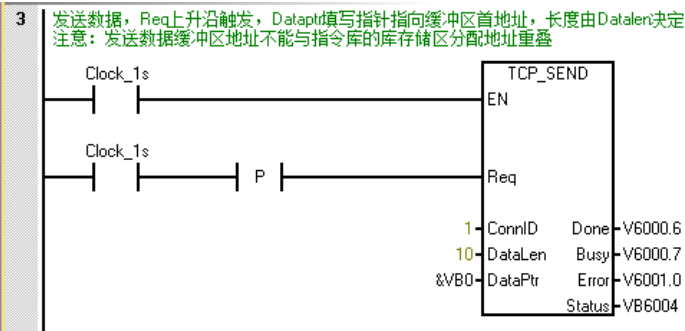


TCP_CONNECT		//调用TCP_CONNECT
EN	: SM0.0	//使能输入
Req	: V6010.0	//如果 Req = TRUE，CPU 启动连接操作。如果 Req = FALSE，则输出显示连接的当前状态。使用上升沿触发

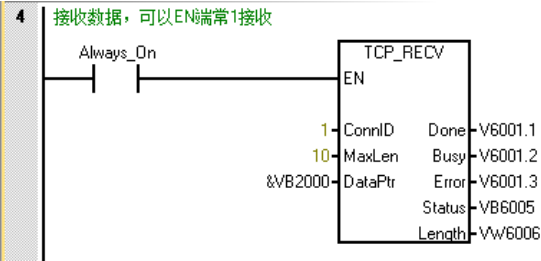
Active	: SM0.0	//TRUE = 主动连接 ;FALSE = 被动连接
ConnID	: 1	// CPU 使用连接 ID (ConnID) 为其它指令标识该连接。可能的 ConnID 范围为 0 到 65534。
IPAddr1~4	: 192.168.0.100	//IPAddr1 是 IP 地址的最高有效字节, IPAddr4 是 IP 地址的最低有效字节。 192.168.0.100为伙伴的IP地址
RemPort	: 2001	//RemPort 是远程设备上的端口号。远程端口号范围为 1 到 49151。
LocPort	: 2000	// LocPort 是本地设备上的端口号。本地端口号范围为 1 到 49151, 但存在一些限制。
DONE	: V6000.0	// 当连接操作完成且没有错误时, 指令置位 Done 输出。
BUSY	: V6000.1	// 当连接操作正在进行时, 指令置位 Busy 输出。
ERROR	: V6000.2	// 当连接操作完成但发生错误时, 指令置位 Error 输出
STATUS	: VB6002	// 如果指令置位 Error 输出, Status 输出会显示错误代码。如果指令置位 Busy 或 Done 输出, Status 为零 (无错误)

图1. 指令位置及调用连接指令

2.调用TCP_SEND 和 TCP_RCV 指令，如图2 所示。



CALL TCP_SEND		//调用TCP_SEND
EN	: SM0.5	//使能输入
Req	: SM0.5	//如果 Req = TRUE, CPU 启动发送操作。如果 Req = FALSE, 则输出显示发送操作的当前状态。
ConnID	:1	//连接 ID (ConnID) 是此发送操作所用连接的编号。使用您为 TCP_CONNECT 操作选择的 ConnID。
DataLen	:10	//DataLen 是要发送的字节数 (1 到 1024) 。
DataPtr	: &VB0	//DataPtr 是指向待发送数据的指针。这是指向 I、Q、M 或 V 存储器的 S7-200 SMART 指针 (例如, &VB100) 。
Done	: V6000.6	// 当连接操作完成且没有错误时, 指令置位 Done 输出。
Busy	: V6000.7	// 当连接操作正在进行时, 指令置位 Busy 输出。
Error	: V6001.0	// 当连接操作完成但发生错误时, 指令置位 Error 输出。
Status	: VB6004	// 如果指令置位 Error 输出, Status 输出会显示错误代码。如果指令置位 Busy 或 Done 输出, Status 为零 (无错误)



CALL TCP_RECV		//调用TCP_RECV
EN	: SM0.0	//使能输入,常1接收
ConnID	: 1	//CPU 将连接 ID (ConnID) 用于此接收操作 (连接过程中定义) 。
MaxLen	: 10	// MaxLen 是要接收的最大字节数 (例如, DataPt 中缓冲区的大小 (1 到 1024)) 。
DataPtr	: &VB2000	// DataPtr 是指向接收数据存储位置的指针。这是指向 I、Q、M 或 V 存储器的 S7-200

SMART 指针 (例如, &VB100)		
Done	: V6001.1	//当接收操作完成且没有错误时, 指令置位 Done 输出。当指令置位 Done 输出时, Length 输出有效。
Busy	: V6001.2	// 当接收操作正在进行时, 指令置位 Busy 输出。
Error	: V6001.3	// 当接收操作完成但发生错误时, 指令置位 Error 输出
Status	: VB6005	// 如果指令置位 Error 输出, Status 输出会显示错误代码。如果指令置位 Busy 或 Done 输出, Status 为零 (无错误)。
Length	: VW6006	//Length 是实际接收的字节数。

图2. 调用发送和接收指令

3.分配库存储区, 如图3 所示。

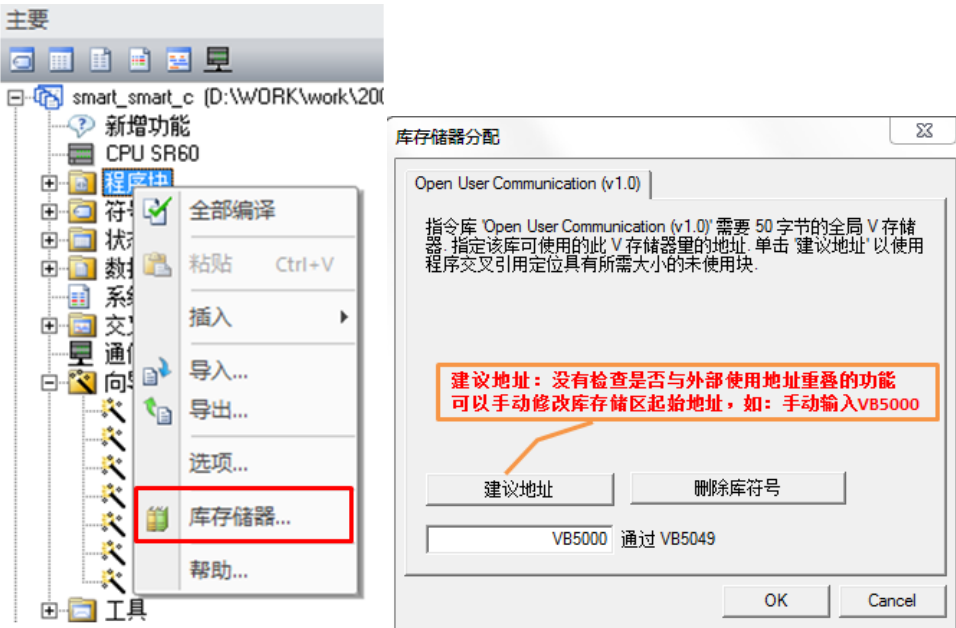


图3. 分配库存储区

⚠ 注意: 为保证指令库可以正常工作, 分配库存储区的时候, 一定不能与程序中使用的其他 V 区地址重叠!!

S7-1200 侧硬件组态和编程 (服务器侧组态编程)

1. 使用 TIA V13 SP1 Upd 9 软件 Portal 视图中, 创建新项目

2. 添加硬件并命名PLC

进入“项目视图”, 在“项目树”下双击“添加新设备”, 在对话框中选择所使用的 S7-1200 CPU 添加到机架上, 命名为 PLC_1 如图4 所示

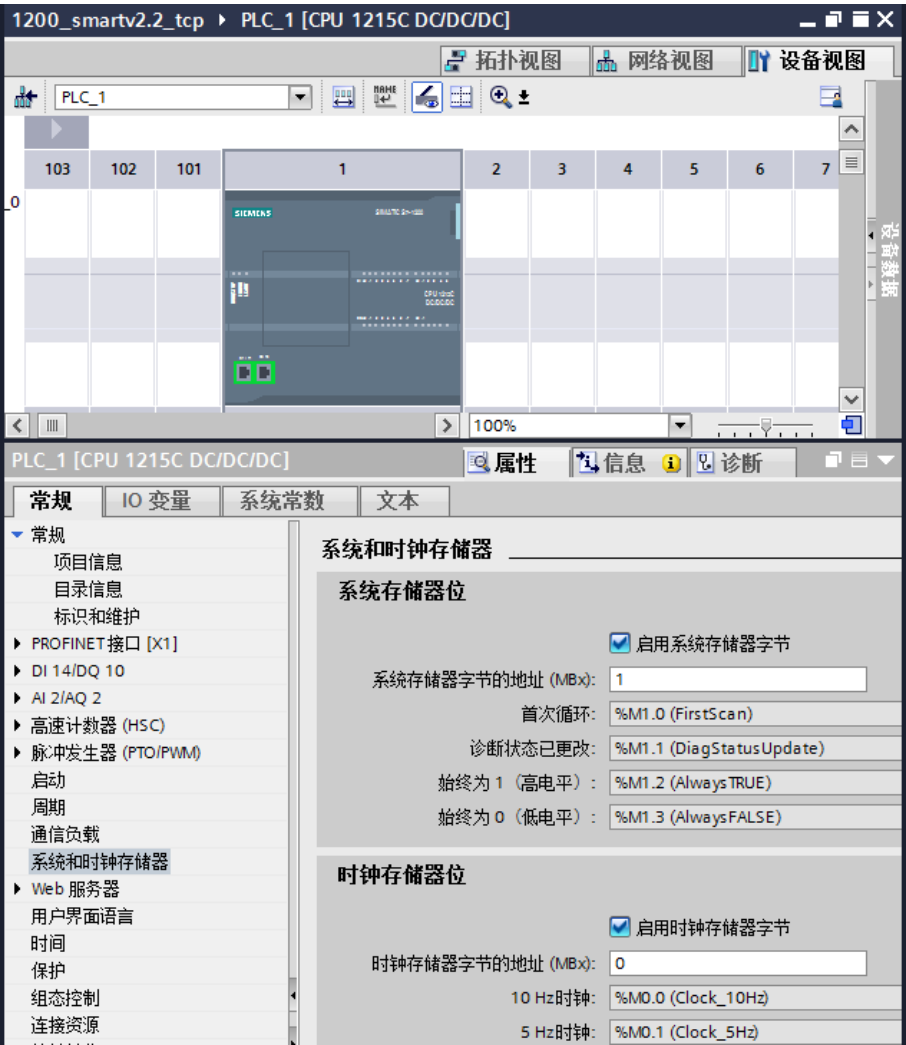


图5.启用时钟位

4. 为 PROFINET 通信口分配以太网地址

在“设备视图”中点击 CPU 上代表 PROFINET 通信口的绿色小方块，在下方会出现 PROFINET 接口的属性，在“以太网地址”下分配 S7-1200 CPU 的 IP 地址为 192.168.0.100，子网掩码为255.255.255.0，如图6 所示

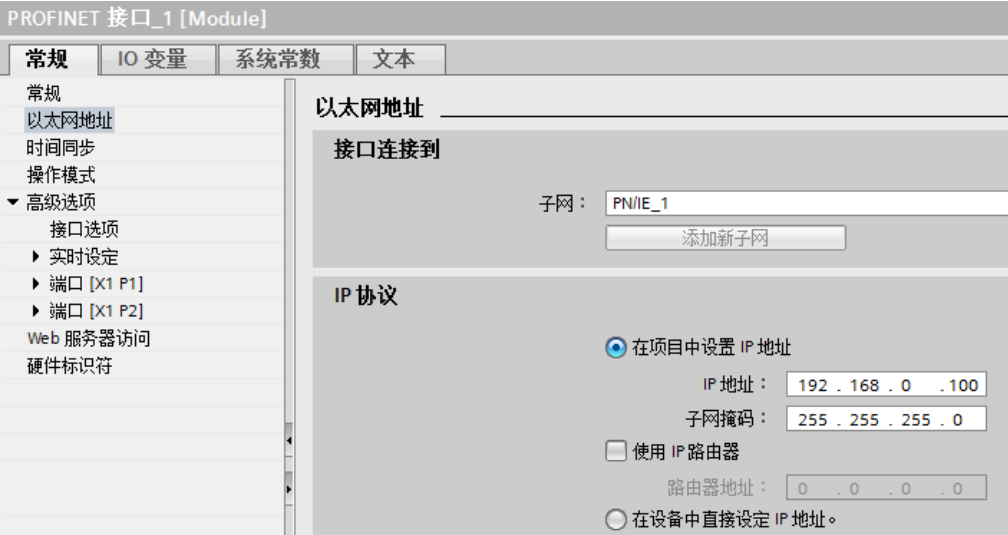


图6. 分配 IP 地址

5.在S7-1200 中调用建立连接指令，进入“项目树”>“ PLC_1”>“程序块”>“OB1” 主程序中，从右侧窗口 “指令”>“通信”>“开放式用户通信”下调用“TCON” 指令，配置连接参数，如图7所示。

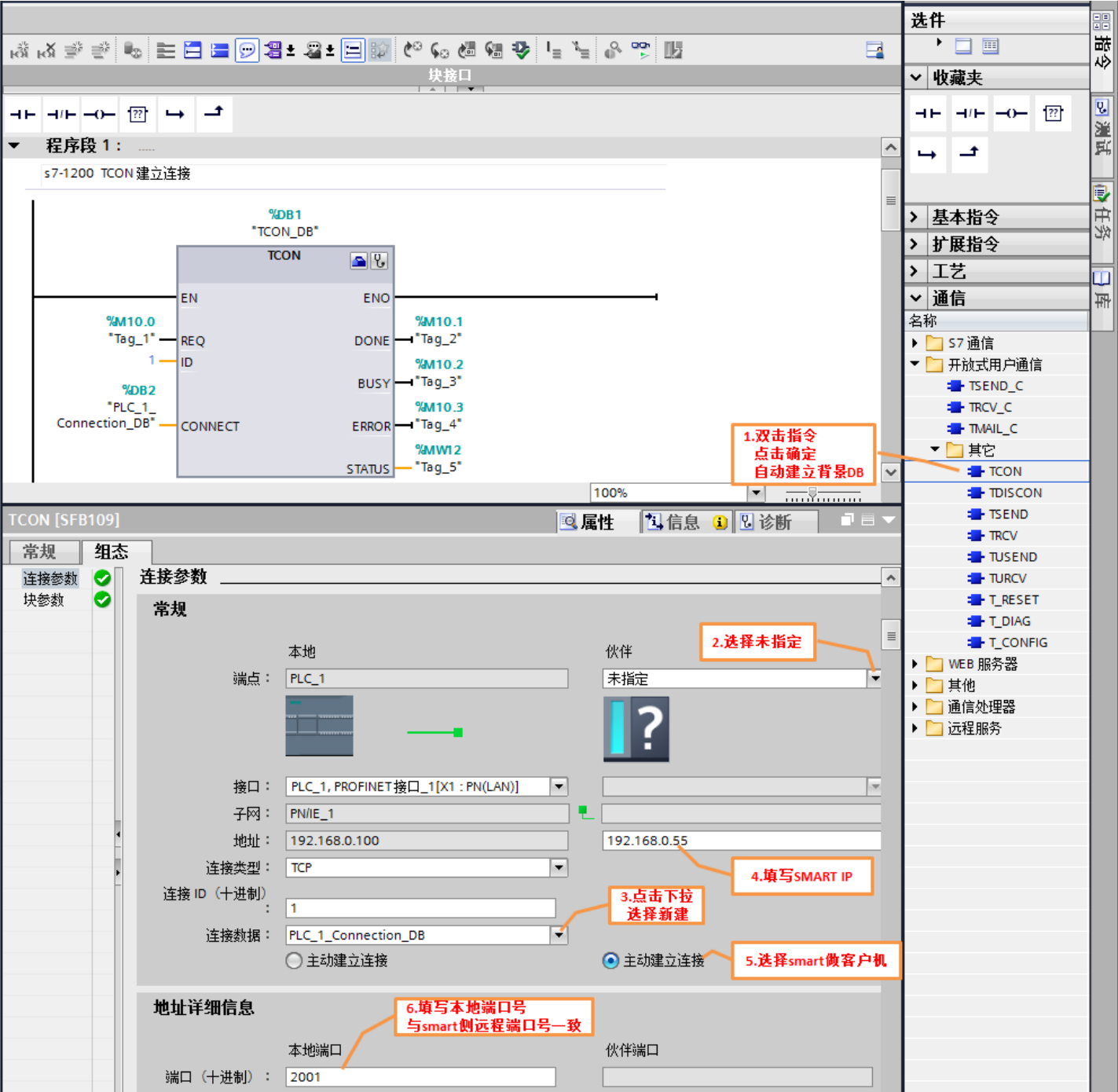


图7. 调用TCON指令并配置连接参数

6.创建用于交换数据的全局DB块及定义变量

通过“项目树”>“PLC_1”>“程序块”>“添加新块”，选择“数据块” 创建全局 DB 块，点击“确定” 生成数据块。如图8所示。

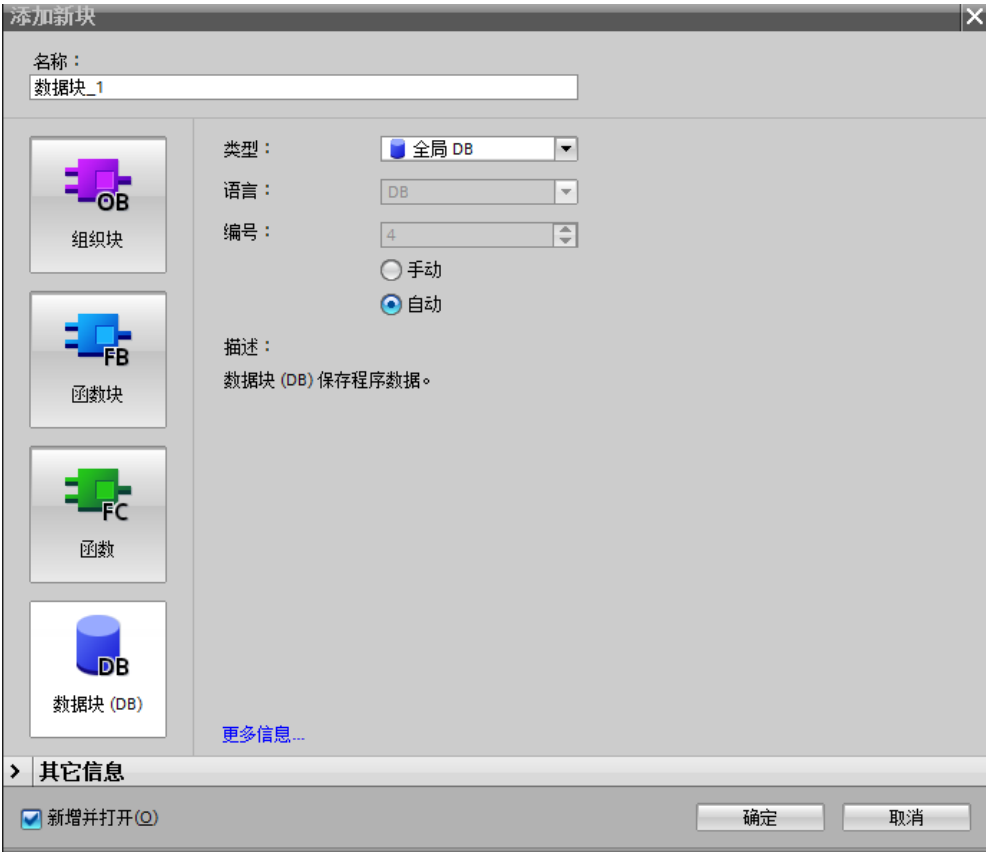


图8.添加新块

通过“项目树”>“PLC_1”>“程序块”>点击生成的 “数据块_1” (DB4)，右键属性，取消勾选“优化的块访问”。如图9所示。

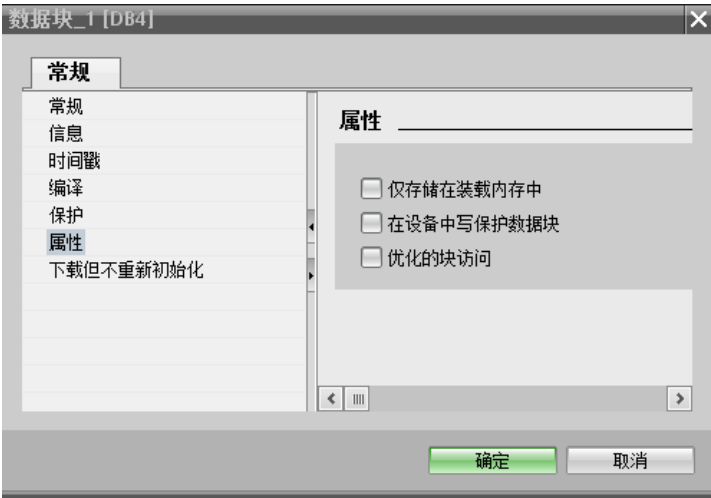


图9.取消勾选优化的块访问

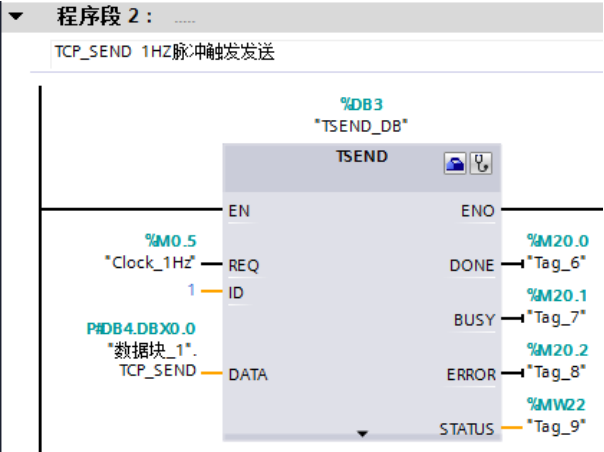
双击生成的 “数据块_1” (DB4)，定义数组，10 个字节用于发送，10个字节用于接收，然后编译数据块。结果如图10所示。

数据块_1			
	名称	数据类型	偏移量
1	Static		
2	TCP_SEND	Array[1..10] of Byte	0.0
3	TCP_RCV	Array[1..10] of Byte	10.0

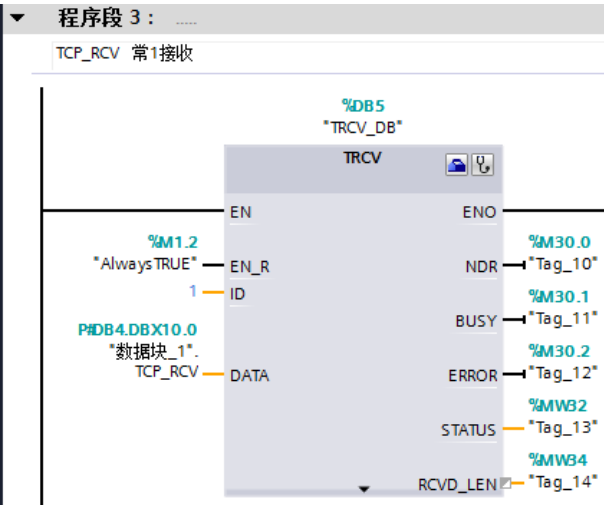
图10. 定义交换数据区

7.在OB1中调用 TSEND 和 TRCV

指令块位置及调用方法参考 TCON 指令的调用，调用结果如图11所示。



CALL TSEND	, DB3	//调用TSEND，使用背景DB块:DB3
REQ	: M0.5	//使用1Hz的时钟脉冲，上升沿激活发送任务
ID	: 1	//连接号，要与连接配置中一致
DATA	: P#DB4.DBX0.0 BYTE 10	// 发送数据区的数据，需要注意DB块要选用绝对寻址
DONE	: M20.0	// 任务执行完成并且没有错误，该位置1
BUSY	: M20.1	// 该位为1，代表任务未完成，不能激活新任务
ERROR	: M20.2	// 通信过程中有错误发生，该位置1
STATUS	: MW22	// 有错误发生时，会显示错误信息号




CALL TRCV	, DB5	//调用TRCV，使用背景DB块:DB5
EN_R	: M1.2	//准备好接受数据 ,常1接收
ID	: 1	//连接号，要与连接配置中一致
DATA	: P#DB4.DBX10.0 BYTE 10	// 接受数据区的数据，需要注意DB块要选用绝对寻址
DONE	: M30.0	// 任务执行完成并且没有错误，该位置1
BUSY	: M30.1	// 该位为1，代表任务未完成，不能激活新任务
ERROR	: M30.2	// 通信过程中有错误发生，该位置1
STATUS	: MW32	// 有错误发生时，会显示错误信息号
RCVD_LEN	: MW34	// 实际接受数据的字节数

图11. 调用发送和接收指令

⚠ 接收和发送指令调用需要注意:

程序块的 ID 要与指令 TCON 中的 ID 保持一致

DATA 参数的填写方法 (以 TRCV 程序块为例):

- 1、直接输入 : P#DB4.DBX10.0 BYTE 10
- 2、打开数据块, 点击数据块右上角浮动按钮, 移动数据块与OB1并排显示, 选中数据块变量名称“ TCP_RCV ”拖拽到 TRCV 指令 DATA 参数处, 该区域显示为绿色, 释放鼠标即可。

通信结果

将 S7-1200 和 S7-200 SMART 项目编译, 下载 ;

S7-1200 侧触发 TCON 指令的REQ (M10.0 置位为1) , S7-200 SMART 侧触发 TCP_CONNECT 指令的Req (V6010.0 置位为1) 以建立连接;

通信结果如下:

	名称	数据类型	偏移量	启动值	监视值
1	Static				
2	TCP_SEND	Array[1..10] of Byte	0.0		
3	TCP_SEND[1]	Byte	0.0	16#0	16#21
4	TCP_SEND[2]	Byte	1.0	16#0	16#22
5	TCP_SEND[3]	Byte	2.0	16#0	16#23
6	TCP_SEND[4]	Byte	3.0	16#0	16#24
7	TCP_SEND[5]	Byte	4.0	16#0	16#25
8	TCP_SEND[6]	Byte	5.0	16#0	16#26
9	TCP_SEND[7]	Byte	6.0	16#0	16#27
10	TCP_SEND[8]	Byte	7.0	16#0	16#28
11	TCP_SEND[9]	Byte	8.0	16#0	16#29
12	TCP_SEND[10]	Byte	9.0	16#0	16#30
13	TCP_RCV	Array[1..10] of Byte	10.0		
14	TCP_RCV[1]	Byte	0.0	16#0	16#01
15	TCP_RCV[2]	Byte	1.0	16#0	16#02
16	TCP_RCV[3]	Byte	2.0	16#0	16#03
17	TCP_RCV[4]	Byte	3.0	16#0	16#04
18	TCP_RCV[5]	Byte	4.0	16#0	16#05
19	TCP_RCV[6]	Byte	5.0	16#0	16#06
20	TCP_RCV[7]	Byte	6.0	16#0	16#07
21	TCP_RCV[8]	Byte	7.0	16#0	16#08
22	TCP_RCV[9]	Byte	8.0	16#0	16#09
23	TCP_RCV[10]	Byte	9.0	16#0	16#0A

	地址	格式	当前值
1	VB0	十六进制	16#01
2	VB1	十六进制	16#02
3	VB2	十六进制	16#03
4	VB3	十六进制	16#04
5	VB4	十六进制	16#05
6	VB5	十六进制	16#06
7	VB6	十六进制	16#07
8	VB7	十六进制	16#08
9	VB8	十六进制	16#09
10	VB9	十六进制	16#0A
11	VB2000	十六进制	16#21
12	VB2001	十六进制	16#22
13	VB2002	十六进制	16#23
14	VB2003	十六进制	16#24
15	VB2004	十六进制	16#25
16	VB2005	十六进制	16#26
17	VB2006	十六进制	16#27
18	VB2007	十六进制	16#28
19	VB2008	十六进制	16#29
20	VB2009	十六进制	16#30