S7-200 SMART 与调试助手之间 TCP 通信

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 TCP&UDP Debug通过 TCP 通信，在 TCP&UDP Debug建立客户端或服务器，在 S7-200 SMART 调用 Open User Communication 库指令(TCP_CONNECT,DISCONNECT,TCP_SEND,TCP_RECV) 。

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 PC （带以太网卡） （IP 地址 192.168.0.254 ； 子网掩码 255.255.255.0）
② S7-200 SMART CPU (固件版本V2.2) (IP 地址 192.168.0.20 ； 子网掩码 255.255.255.0)
③ TP 以太网电缆
软件：
① TCP&UDP Debug
② STEP 7 Micro/WIN SMART （软件版本 V2.2 ）
所完成的通信任务：
① TCP&UDP Debug 发送 10 个字节数据： --> （ S7- 200 SMART 侧） VB2000~VB2009
② TCP&UDP Debug 接收 10 个字节数据： <-- (S7-200 SMART 侧） VB0~VB9

TCP&UDP Debug设置（客户端设置）

1. 打开TCP&UDP Debug软件

2. 右键“客户端模式”---“创建连接”

如图1所示，创建连接

图1.创建连接

① 选择连接类型：TCP
② 填写S7-200 SMART的IP地址及端口号
③ 设置PC的端口号
④ 单击创建

3. 右键“客户端模式”----“连接”

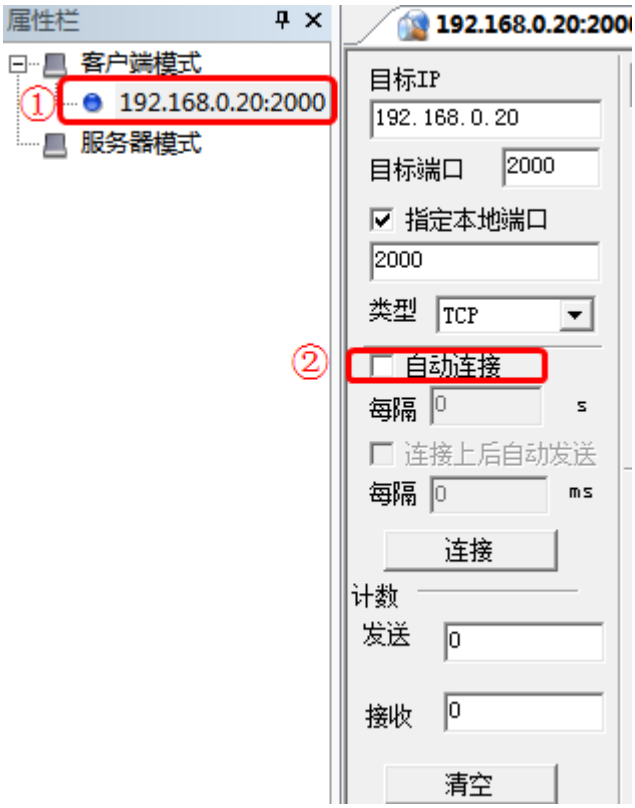

图2. 建立连接

① 蓝色圆点：未建立连接；黄色三角箭头：正在建立连接；绿色三角箭头：连接建立成功；
② 勾选“自动连接”后，点击连接按钮。

TCP&UDP Debug设置（服务器设置）

1. 打开TCP&UDP Debug软件

2. 右键“服务器模式”---“创建服务器”

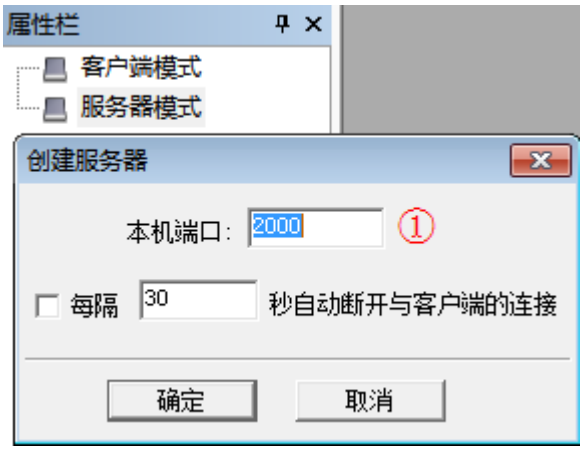

图3. 创建服务器

① 选择PC的端口号后，点击确定

3. 右键“服务器模式”----“启动服务器”

图4. 启动服务器

S7-200 SMART侧编程

1.打开STEP 7 Micro/WIN SMART>项目树>指令树>库>Open User Communication ,调用TCP_CONNECT，如图5、图6所示。

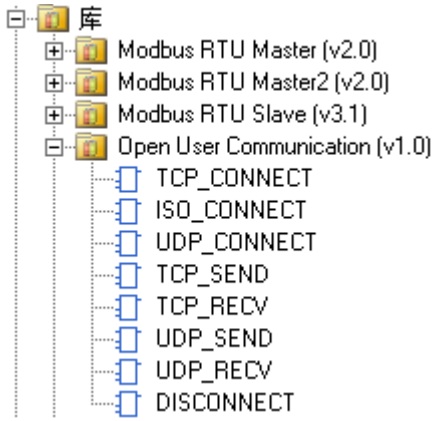

图5. 开放式以太网通讯指令库

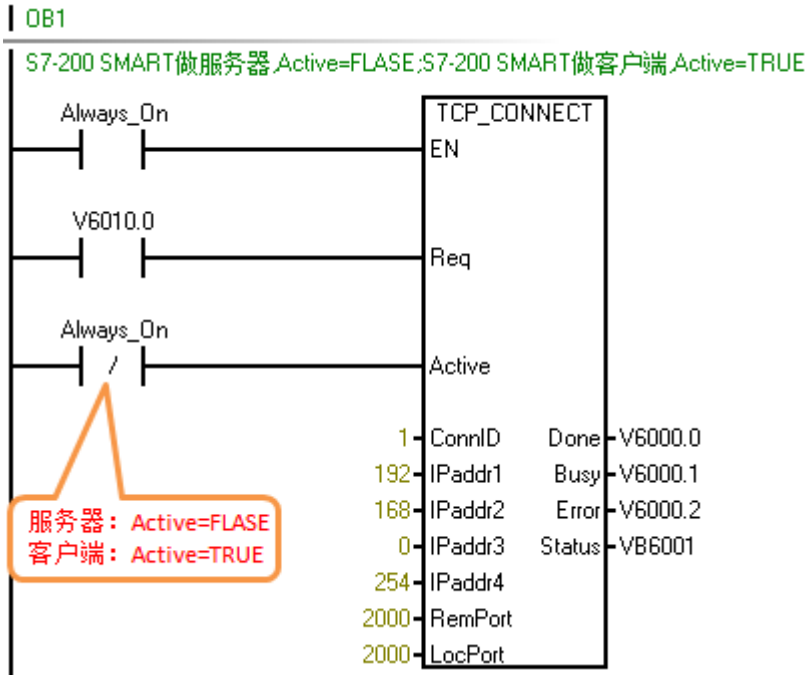

图6. TCP连接块

表1. 引脚说明

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 闭点	//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	：0~0	//IPaddr1 是 IP 地址的最高有效字节，IPaddr4 是 IP 地址的最低有效字节。服务器侧IP地址写0，表示接收所有请求
RemPort	：0	//RemPort 是远程设备上的端口号。远程端口号范围为 1 到 49151。对于被动连接，使用零。
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）

2.调用TCP_SEND 和 TCP_RCV 指令，如图7、图8所示。

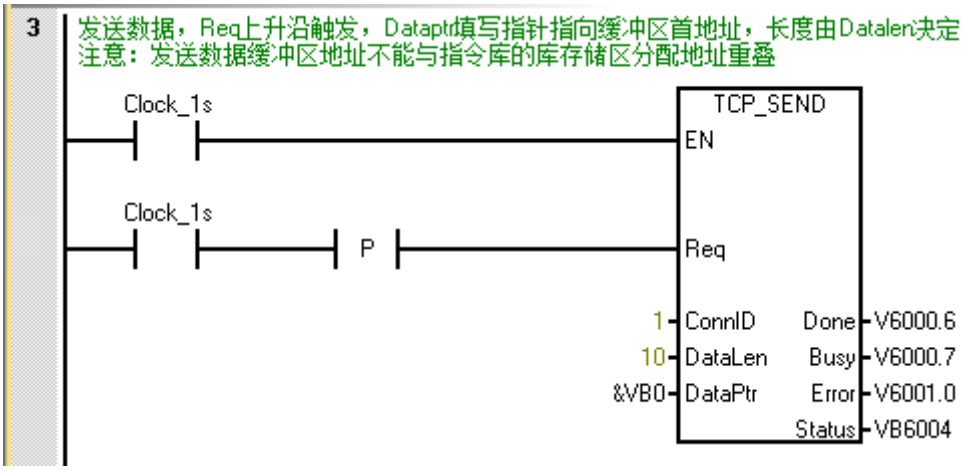

图7. TCP发送块

表2.引脚说明

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）

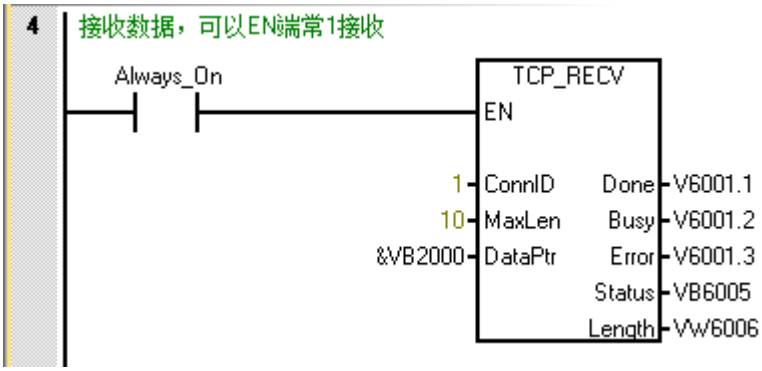

图8. 发TCP接收块

表3. 引脚说明

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。

3.分配库存储区，如图9 所示。

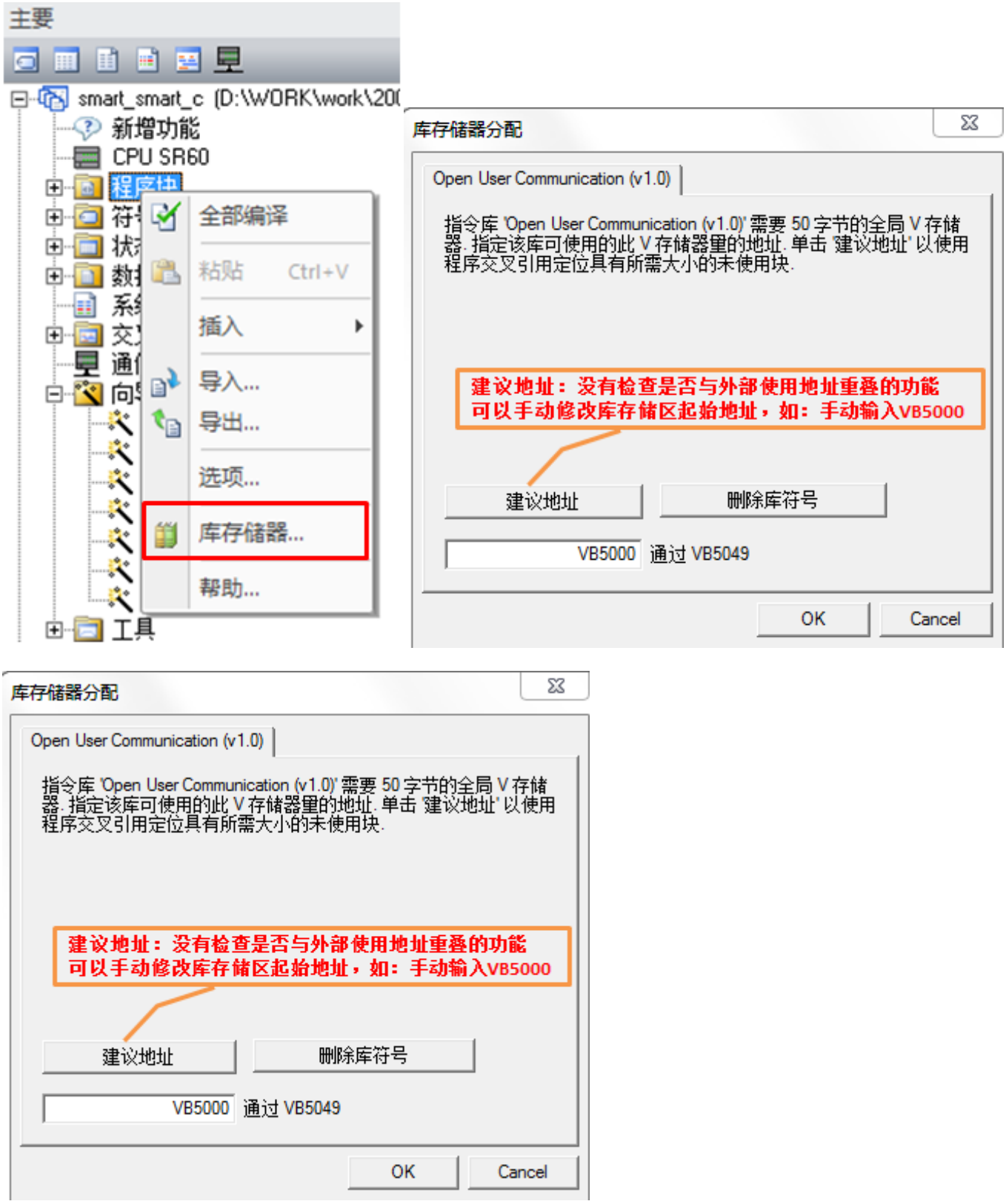

图9. 分配库存储区

⚠ 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！！

通信结果

以S7-200 SMART做服务器为例。将 S7-200 SMART 项目编译，下载并触发建立连接；在TCP&UDP Debug中点击连接。

通信结果如下：

VB0	十六进制	16#01
VB1	十六进制	16#02
VB2	十六进制	16#03
VB3	十六进制	16#04
VB4	十六进制	16#05
VB5	十六进制	16#06
VB6	十六进制	16#07
VB7	十六进制	16#08
VB8	十六进制	16#09
VB9	十六进制	16#0A
	有符号	
VB2000	十六进制	16#11
VB2001	十六进制	16#12
VB2002	十六进制	16#13
VB2003	十六进制	16#14
VB2004	十六进制	16#15
VB2005	十六进制	16#16
VB2006	十六进制	16#17
VB2007	十六进制	16#18
VB2008	十六进制	16#19
VB2009	十六进制	16#20

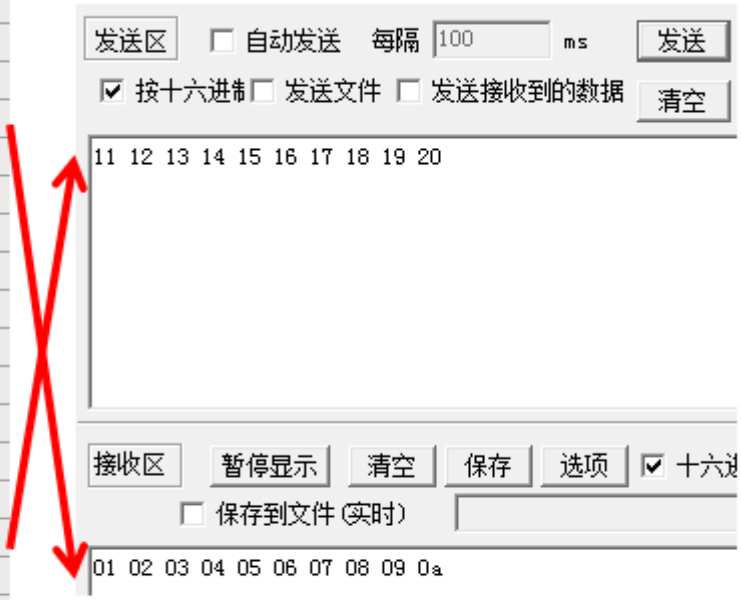

图10. 测试结果