**TCP自由协议**

自由协议通讯也称为无协议通讯，即通讯双方没有共同的通讯协议，只能临时根据某一方的协议进行发送和接收数据，以达到交换数据的目的。

TCP 是面向连接的传输控制协议，相较于 UDP 具有高可靠性，确保传输数据的正确性，不出现丢失或乱序。

使用 TCP 自由协议指令前须在工程中添加“NetWork”库文件。

本文介绍SC2-C使用TCP自由协议分别作TCP主站、TCP从站实现与上位机 （电脑）的数据交互。

**1.TCP自由协议指令**

TCP自由协议指令一览。

|  |  |  |
| --- | --- | --- |
| **指令类别** | **名称** | **功能** |
| TCP自由通讯指令 | TCP\_Client | TCP客户端通讯 |
| TCP\_Server | TCP服务器端通讯 |
| TCP\_Connection | 连接到服务器 |
| TCP\_Write | TCP通讯数据写入 |
| TCP\_Read | TCP通讯数据接收 |

1.1 TCP客户端通讯TCP\_Client

该指令能将PLC设置为客户端，连接到指定的TCP服务器，输出的客户端句柄用于建立通讯。其格式如下。

NBS.TCP\_Client(xEnable:=使能信号, xDone=>完成信号, xBusy=>执行中信号,xError=>错误标志,

udiTimeOut:=超时时间, ipAddr:=服务器IP地址, uiPort:=端口号,

eError=>错误码, xActive=>连接成功标志, hConnection=>客户端句柄 );

1.2 TCP服务器端通讯TCP\_Server

该指令能将PLC设置为服务器，监听TCP客户端。通过输出句柄hServer用于通过TCP\_Connection建立连接，若实例化多个TCP\_Connection建立连接，可实现一个服务器处理多个客户端连接功能。其格式如下。

NBS.TCP\_Server(xEnable:=使能信号, xDone=>完成信号, xBusy=>执行中信号,xError=>错误标志,

ipAddr:= 设置服务器IP地址, uiPort:= 端口号, eError=>错误信号, hServer=>服务器句柄 );

1.3 连接到服务器TCP\_Connection

该指令能使用TCP\_Server指令创建的服务器句柄，与客户端建立连接。若有多个客户端需要与PLC建立连接，可通过实例化多个TCP\_Connetction实现，生成的连接句柄用于读写操作。其格式如下。

NBS.TCP\_Connection(xEnable:= 使能信号, xDone=> 完成信号, xBusy=> 执行中信号,

xError=> 错误标志, hServer:= 服务器句柄, eError=> 错误信号,

xActive=> 连接成功标志, hConnection=> 连接句柄 );

1.4 TCP通讯数据写入TCP\_Write

该指令能对TCP\_Client指令与TCP\_Connetction指令中已建立连接的句柄hConnection进行写入数据操作。其格式如下。

NBS.TCP\_Write(xExecute:= 启动信号, udiTimeOut:= 超时时间, xDone=> 完成信号 ,

xBusy=> 执行中信号, xError=> 错误标志, hConnection:=连接句柄 ,

szSize:= 数据大小, pData:= 发送数据缓存区指针, eError=>错误信号 );

1.5 TCP通讯数据写入TCP\_Read

该指令能对TCP\_Client指令与TCP\_Connetction指令中已建立连接的句柄hConnection进行读取数据操作。其格式如下。

NBS.TCP\_Read(xEnable:=使能信号, xDone=>完成信号, xBusy=>执行中信号, xError=>错误标志,

hConnection:= 连接句柄, szSize:= 读取数据的大小,

pData:= 读取数据缓存区指针, eError=> 错误信号,

xReady=> 读取数据成功标志, szCount=> 实际读取的字数 );

表12.4 自由协议错误码说明

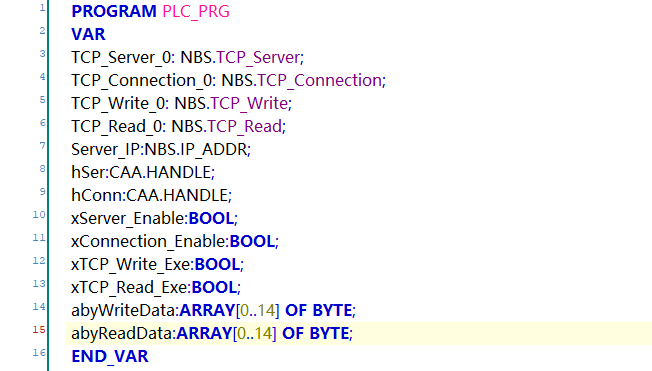
|  |  |  |
| --- | --- | --- |
| **错误码** | **定义** | **描述** |
| 0 | NO\_ERROR | 无错误 |
| 6000 | FIRST\_ERROR | 保留 |
| 6001 | TIME\_OUT | 超时 |
| 6002 | INVALID\_ADDR | 客户端所连接服务器的IP地址无效 |
| 6003 | INVALID\_HANDLE | 连接句柄无效 |
| 6004 | INVALID\_DATAPOINTER | 数据指针无效 |
| 6005 | INVALID\_DATASIZE | 数据大小无效 |
| 6006 | UDP\_RECEIVE\_ERROR | UDP接收错误 |
| 6007 | UDP\_SEND\_ERROR | UDP发送错误 |
| 6008 | UDP\_SEND\_NOT\_COMPLETE | UDP发送未完成 |
| 6009 | UDP\_OPEN\_ERROR | UDP打开错误 |
| 6010 | UDP\_CLOSE\_ERROR | UDP关闭错误 |
| 6011 | TCP\_SEND\_ERROR | TCP发送错误 |
| 6012 | TCP\_RECEIVE\_ERROR | TCP发送未完成 |
| 6013 | TCP\_OPEN\_ERROR | TCP打开错误 |
| 6014 | TCP\_CONNECT\_ERROR | TCP连接错误 |
| 6015 | TCP\_CLOSE\_ERROR | TCP关闭错误 |
| 6016 | TCP\_SERVER\_ERROR | TCP服务器错误 |
| 6017 | WRONG\_PARAMETER | 参数错误 |
| 6018 | ERROR\_UNKNOWN | 未知错误 |
| 6019 | TCP\_NO\_CONNECTION | 无TCP连接 |
| 6020 | LOCTL\_ERROR | 内部错误（本机不支持） |
| 6050 | FIRST\_MF | 保留 |
| 6099 | LAST\_ERROR | 保留 |

**2.SC2-C作服务器（Server）**

SC2-C作Server端，电脑NetAssist网络调试助手软件作Client端，实现SC2-C与电脑之间的数据交互。

2.1创建变量

创建下图所示变量。



2.2 设置服务器参数

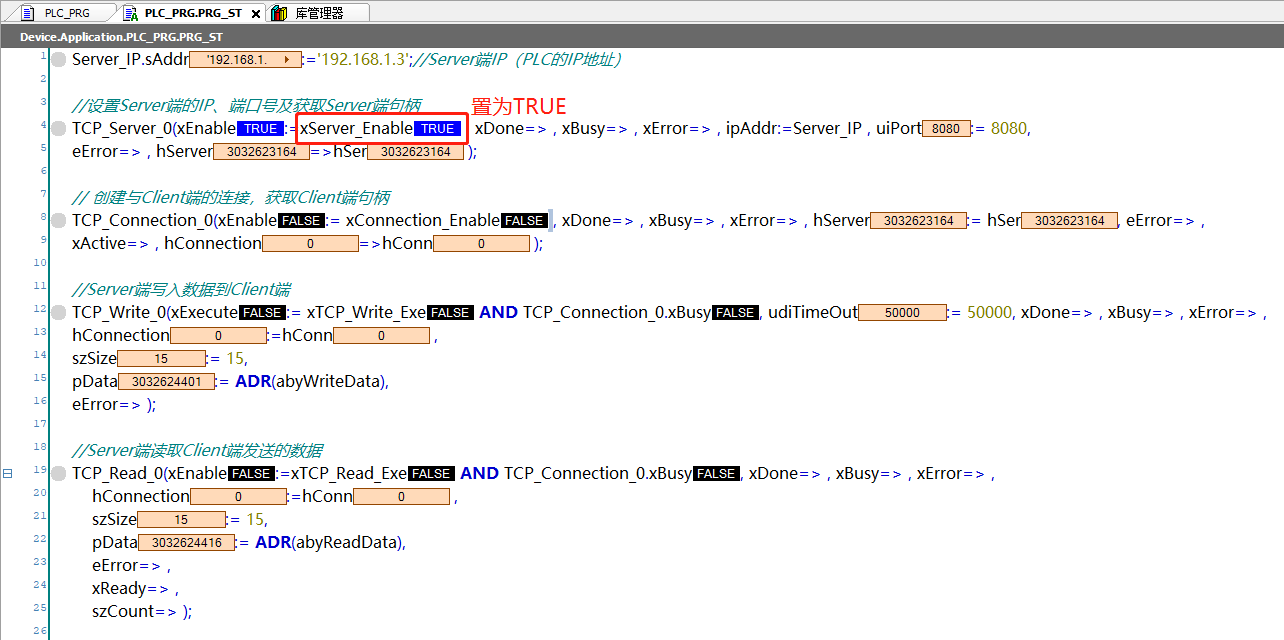
① 调用TCP\_Server指令；

② 设置相关参数，ipAddr为NBS.IP\_Server 结构体变量，需要对结构体变量成员sAddr 进行赋值（Server端的IP地址，为字符串），即IP.sAddr:='192.168.1.3'; （Server端 IP 地址为192.168.1.3）；

③ uiPort为Server端的端口号，UINT数据类型，端口号为 8080（自由定义，但不能和其余通信端口号重复）；

④ hServer为通信句柄，CAA.HANDLE结构体变量；

⑤ TCP\_Server的xEnable至TRUE。



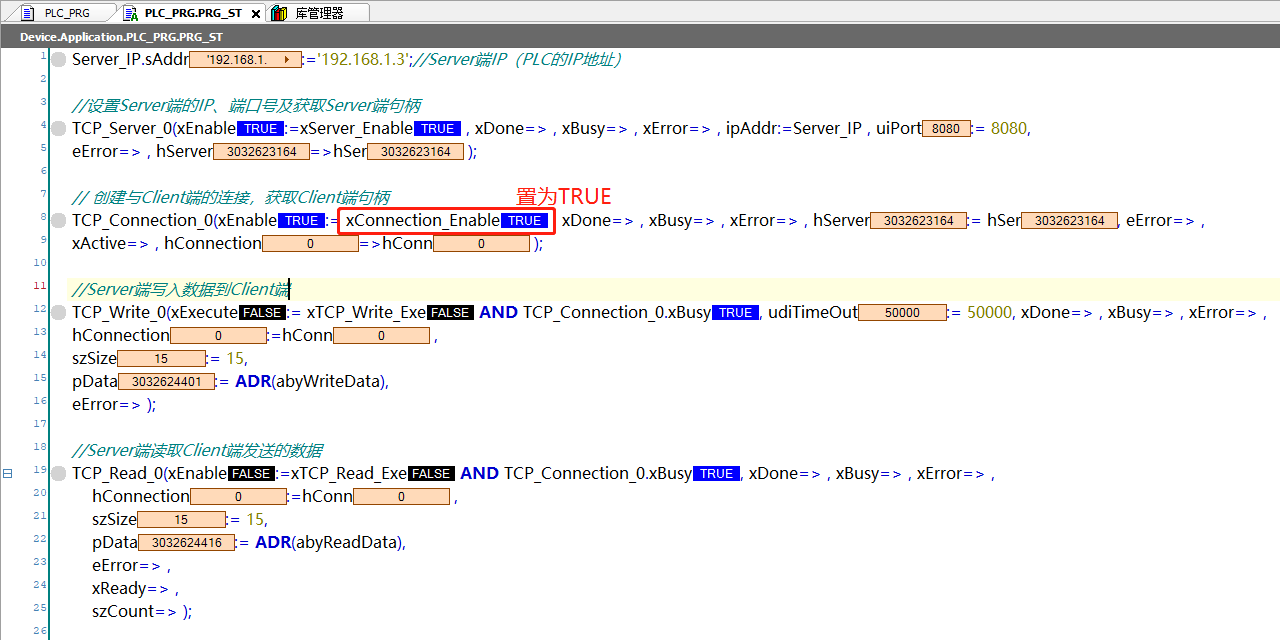
2.3 打开SC2-C服务器

① 调用TCP\_Connection;

② hServer为Server端通信句柄，使用TCP\_Server获取的通信句柄即可；

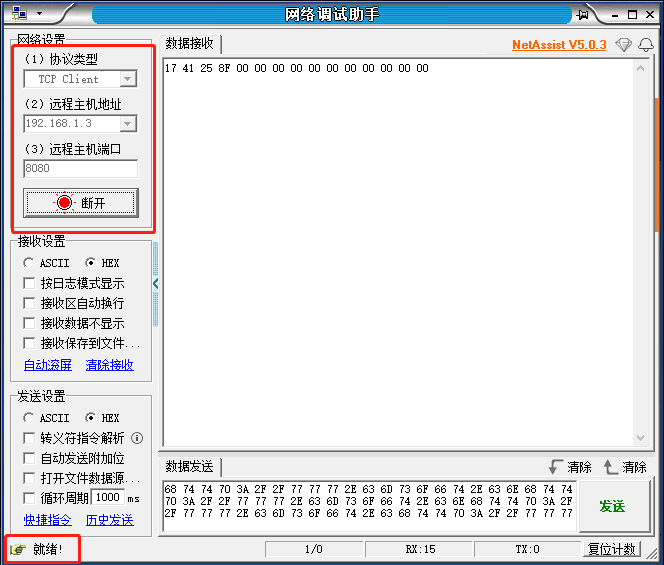
③ hConnection为 Client端输出句柄，CAA.HANDLE结构体变量；

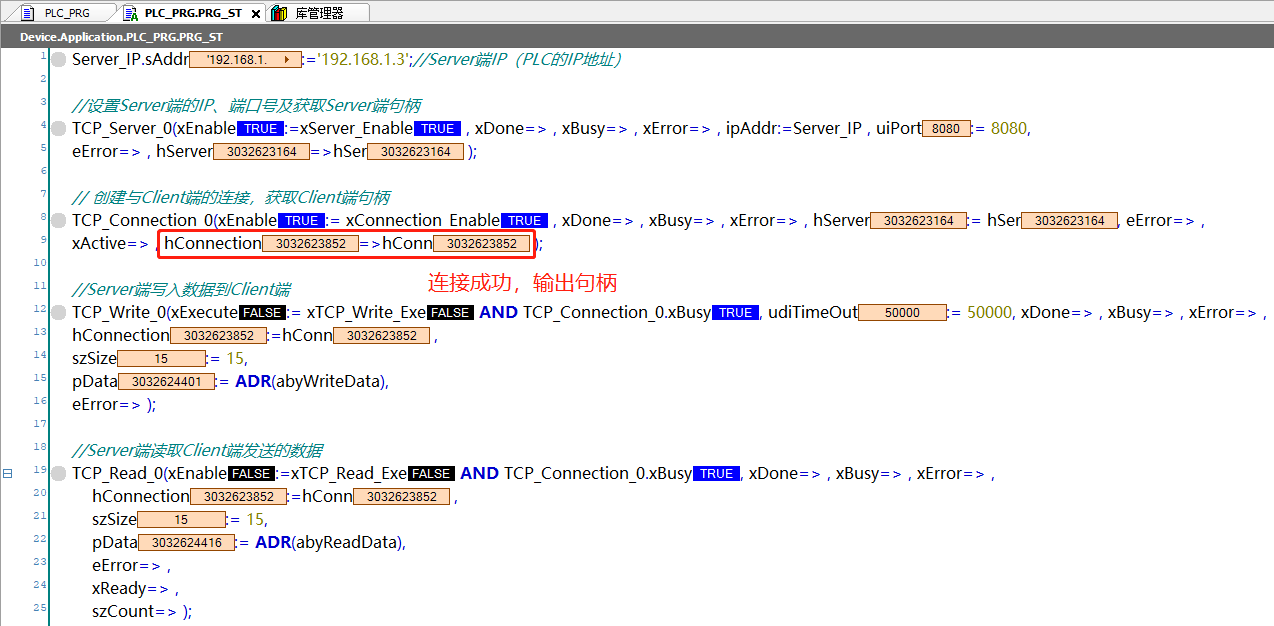
④ TCP\_Connection的xEnable至TRUE。



2.4 连接NetAssist客户端

打开 NetAssist 网络调试助手软件，【协议类型】为TCP Client，【本地IP地址】为 192.168.1.3，【本地端口】为 8080，设置好参数后左键单击【连接】，确保 TCP\_Server与TCP\_Client端连接成功。连接成功时，TCP\_Connection的hConnection输出句柄。

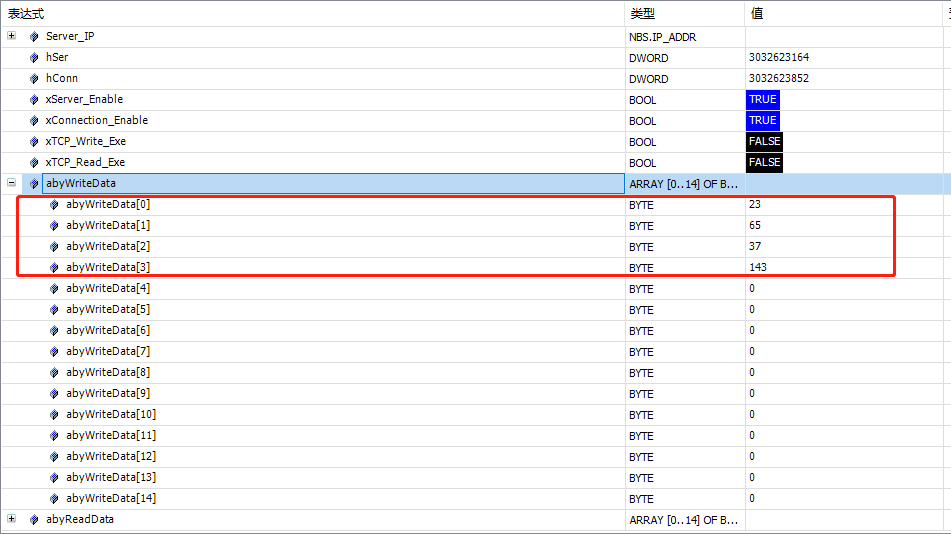




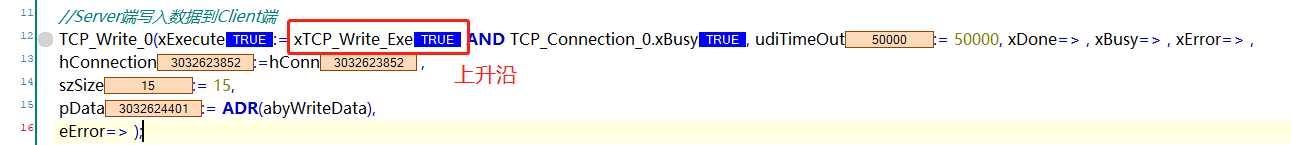
2.5 服务器（SC2-C）发送数据

① 调用TCP\_Write指令。hConnection为TCP\_Client的输出句柄；pData为指针变量，是写入数据变量所指向的地址；szSize为写入数据变量的数据长度；

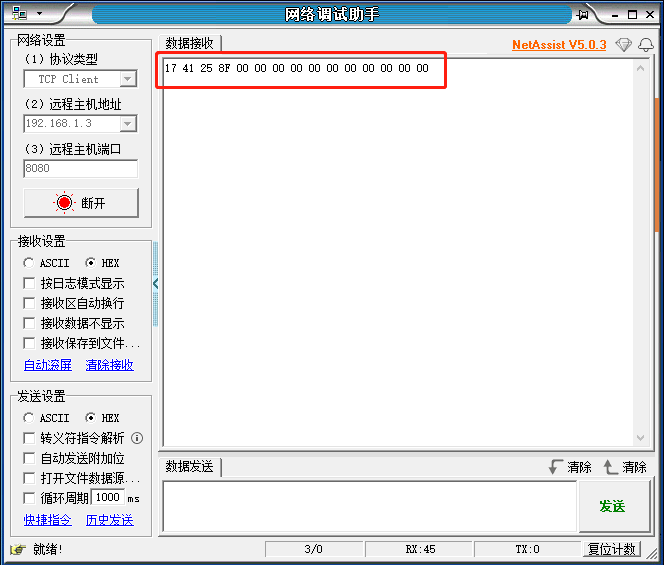
② 将 byWriteData[0]、byWriteData[1]、byWriteData[2]及byWriteData[3]分别赋值十进制数值23、65、37及143；



③ 触发TCP\_Write指令xExecute上升沿。



④ 查看NetAssist软件接收数据结果。 (17H=23D，41H=65D，25H=37D，8FH=143D)

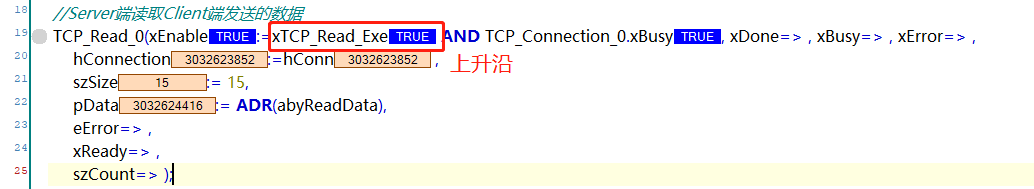


2.6 客户端（NetAssist软件）发送数据

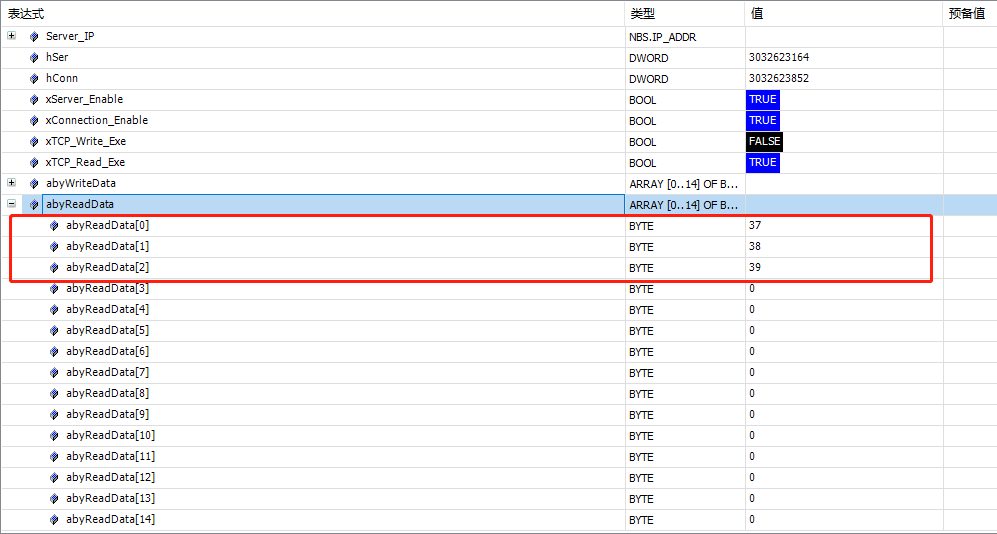
①使用 NetAssist 网络调试助手作为Client端发送16进制数值25 26 27（十进制为37、38及39）；



②触发TCP\_Read的xExecute上升沿；



③查看SC2-C接收数据结果。abyReadData[0]、abyReadData[1]及abyReadData[2]分别的数值为37、38及39，说明读取Client端发送的数据成功。

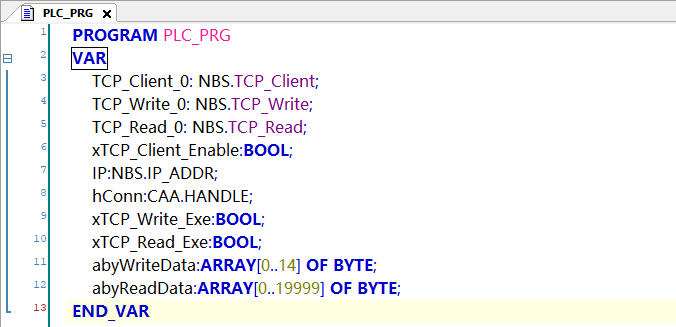


**3.SC2-C作客户端（Client）**

SC2-C作Server端，电脑NetAssist网络调试助手软件作Client端，实现SC2-C与电脑之间的数据交互。

3.1创建变量

创建下图所示变量。



3.2打开NetAssist服务器

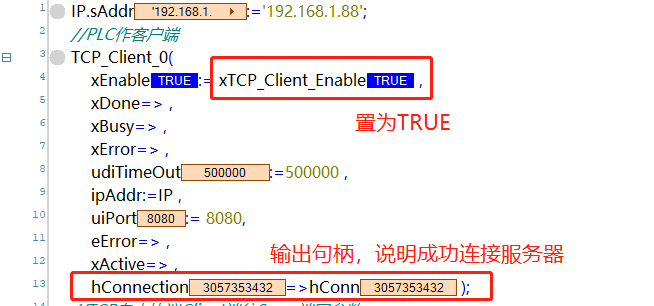
①打开NetAssist网络调试助手软件，【协议类型】为 TCP Server，【本地 IP地址】为 192.168.1.88，【本地端口】为 8080，设置好参数后左键单击【打开】；



3.3连接SC2-C客户端

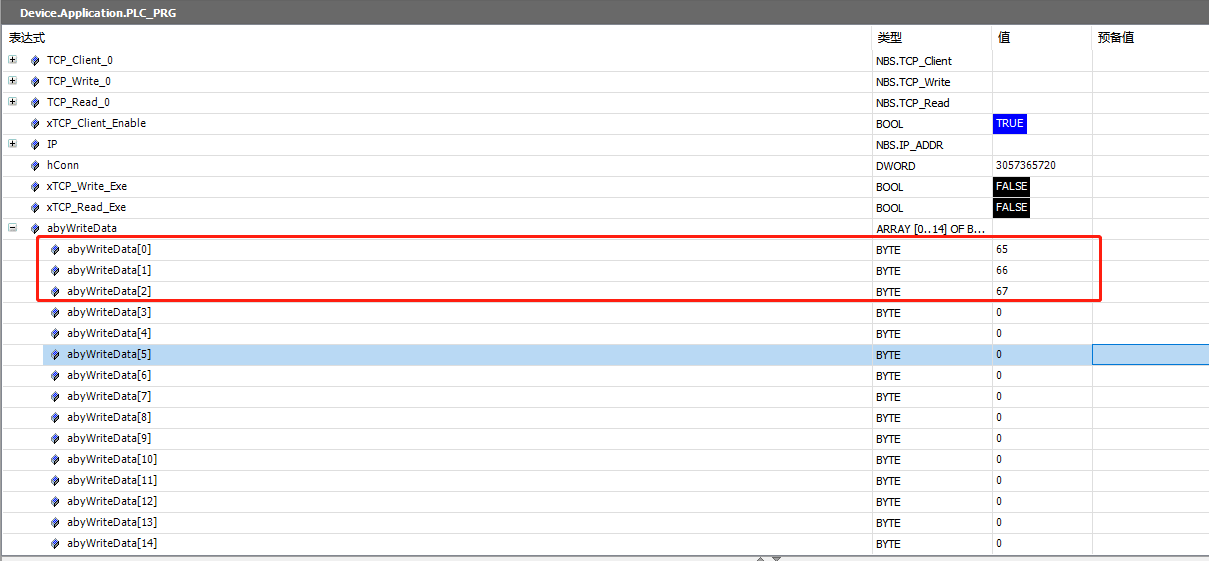
① 调用TCP\_Client指令，ipAddr为NBS.IP\_Client结构体变量，需要对结构体变量成员sAddr进行赋值（Server端的IP地址，为字符串），即IP.sAddr:='192.168.1.88';（Server 端IP地址为192.168.1.88）

② TCP\_Client的xEnable为TRUE后开始连接，当hConnection输出句柄，说明连接成功。

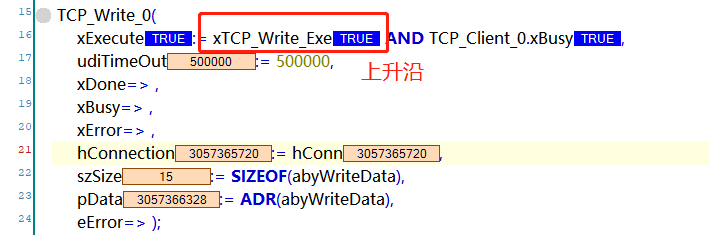


3.4客户端（SC-2C）发送数据

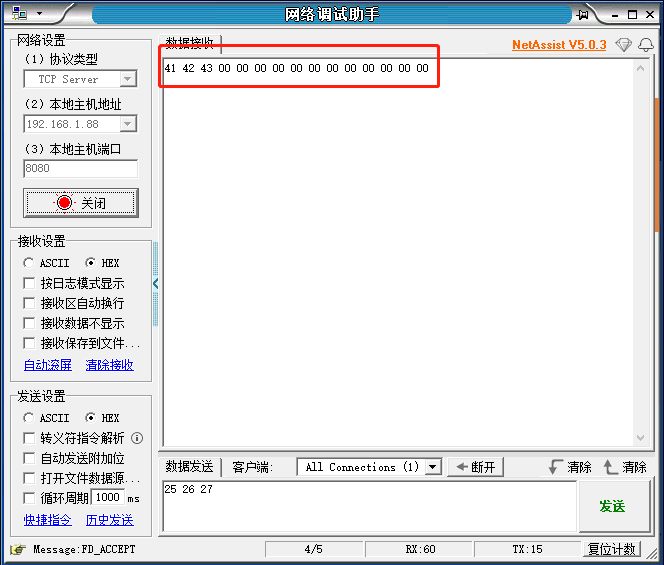
① 将abyWriteData[0]、abyWriteData[1]及abyWriteData[2]赋值十进制数值65、66及67；



② 触发TCP\_Write指令xExecute上升沿；



③ 查看NetAssist软件接收数据结果。 (65D=41H，66D=42H，66D=43H)

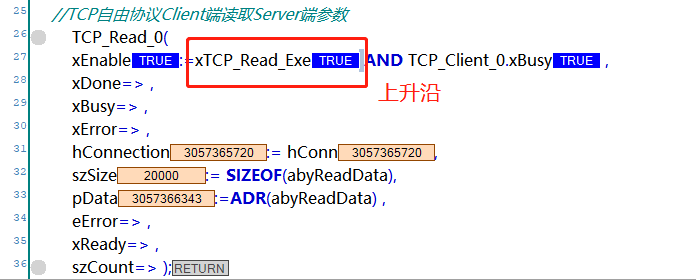


3.5服务器（NetAssist软件）发送数据

①使用 NetAssist 网络调试助手作为Server 端发送16进制数值12 23 34（十进制为37、38及39）；



②触发TCP\_Read的xExecute上升沿；



③查看SC2-C接收数据结果。abyReadData[0]、abyReadData[1]及abyReadData[2]分别的数值为37、38及39，说明读取客户端发送的数据成功。

