Modbus 应用实战之第 001 篇

三菱 FX5U 和西门子 200 Smart 实现 Modbus TCP 以太网通信

作者：关普，中华工控网 ID:guanyumou

一、必备软件和硬件：

1、三菱 GX Works3 编程软件

2、西门子 STEP 7-MicroWIN SMART 编程软件

3、Modbus Poll 调试软件

4、Modbus Slave 调试软件

5、以太网调试助手

6、三菱 FX5U 系列 PLC

7、西门子 200 Smart 系列 PLC

8、无线路由器

9、网线

二、通信实现目的：

1、三菱 FX5U 读取西门子 200 Smart 数字量输入 I0.0~I0.7 并映射到自己的数字量输出 Y0~Y7 上，使用 Modbus 之 02 功能码实现；

2、三菱 FX5U 使用自己的数字量输入 X0~X7 控制西门子 200 Smart 数字量输出 Q0.0~Q0.7，使用 Modbus 之 15 功能码实现；

3、三菱 FX5U 读取西门子 200 Smart 保持寄存器 VW0~VW6 并保存到自己的保持寄存器 D0~D3 里，使用 Modbus 之 03 功能码实现；

4、三菱 FX5U 使用自己的保持寄存器 D4~D7 控制西门子 200 Smart 保持寄存器 VW8~VW14，使用 Modbus 之 16 功能码实现。

三、通信连接说明：

1、三菱 FX5U 本体自带以太网口通过网线连接至无线路由器 LAN 接口；

2、西门子 200 Smart 本体自带以太网口通过网线连接至无线路由器 LAN 接口。

四、Modbus TCP 服务器通信参数：

1、Modbus TCP 服务器：西门子 200 Smart

2、Modbus TCP 服务器 IP 地址：192.168.1.150

3、Modbus TPC 服务器子网掩码：255.255.255.0

4、Modbus TCP 服务器默认网关：192.168.1.1

5、Modbus TCP 服务器端口号：502

五、Modbus TCP 客户端通信参数：

1、Modbus TCP 客户端：三菱 FX5U

2、Modbus TCP 客户端 IP 地址：192.168.1.140

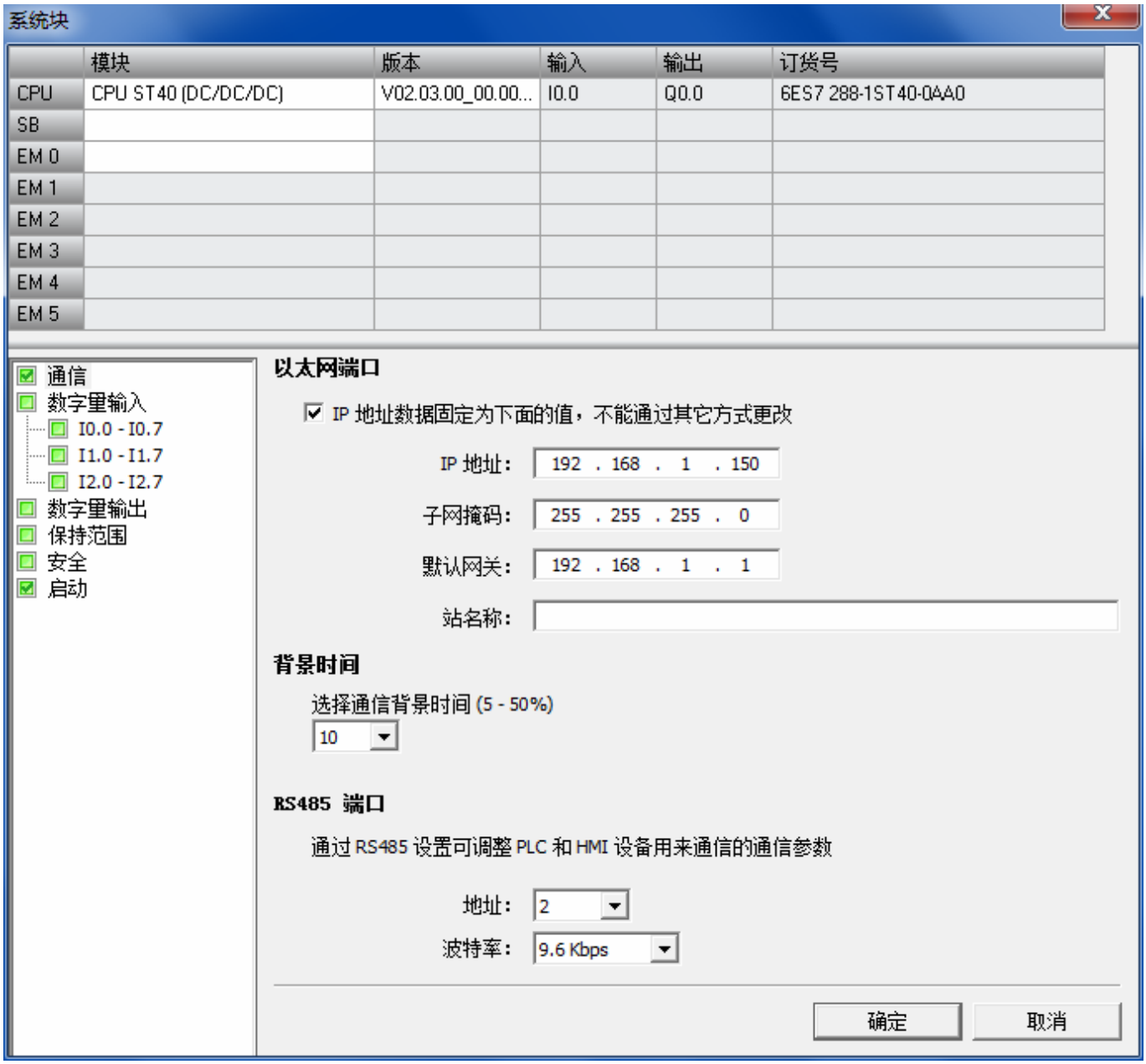
3、Modbus TPC 客户端子网掩码：255.255.255.0

4、Modbus TCP 客户端默认网关：192.168.1.1

5、Modbus TCP 客户端端口号：502

六、西门子 200 Smart 通信参数设置：

1、西门子 200 Smart 通信参数设置如下所示：

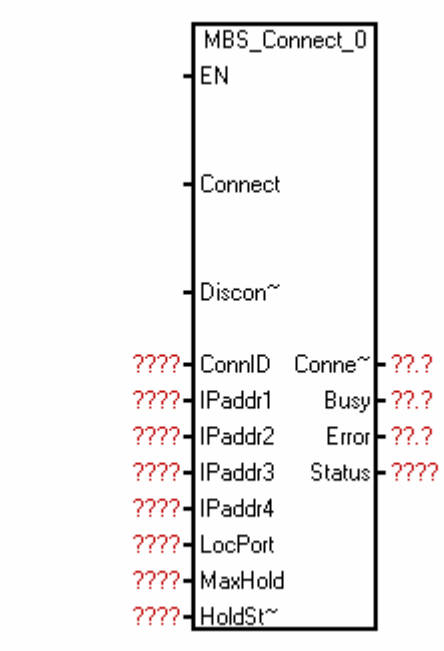


2、西门子 200 Smart 设备编号和 Modbus 寄存器编号、地址对应表如下所示：

类型	设备编号	Modbus 寄存器编号	Modbus 寄存器地址	支持功能码
数字量输入	I0.0~I0.7	1x00001~1x00008	16#0000~16#0007	02
数字量输出	Q0.0~Q0.7	0x00001~0x00008	16#0000~16#0007	01/05/15
保持寄存器	VW0~VW6	4x00001~4x00004	16#0000~16#0003	03/06/16
保持寄存器	VW8~VW14	4x00005~4x00008	16#0004~16#0007	03/06/16

七、西门子 200 Smart 实现 Modbus TCP 服务器相关指令：

1、MBS_Connect 指令：



MBS_Connect 指令各个参数定义如下所示：

(1)、EN 使能：必须保证每一扫描周期都被使能；

(2)、Connect：启动 TCP 连接建立操作；

(3)、Disconnect：断开 TCP 连接操作；

(4)、ConnID：TCP 连接标识；

(5)、IPaddr1~IPaddr4：Modbus TCP 客户端的 IP 地址，IPaddr1 是 IP 地址的最高有效字节，IPaddr4 是 IP 地址的最低有效字节。如果不指定客户端 IP 地址，则可以设置为 0.0.0.0；

(6)、LocPort： 本地设备上端口号；

(7)、MaxHold：用于设置 Modbus 地址 4xxxx 或 4yyyyy 可访问的 V 存储器中的字保持寄存器数；

(8)、HoldStart：间接地址指针，指向 CPU 中 V 存储器中保持寄存器的起始地址；

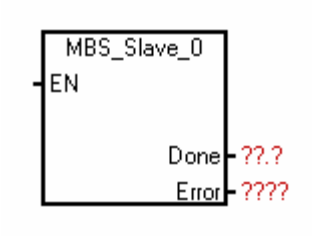
(9)、ConnectDone：Modbus TCP 连接已经成功建立；

(10)、Busy：连接操作正在进行时；

(11)、Error：建立或断开连接时，发生错误；

(12)、Status：如果指令置位“Error”输出，Status 输出会显示错误代码。

2、MBS_Slave 指令：



MBS_Slave 指令各个参数定义如下所示：

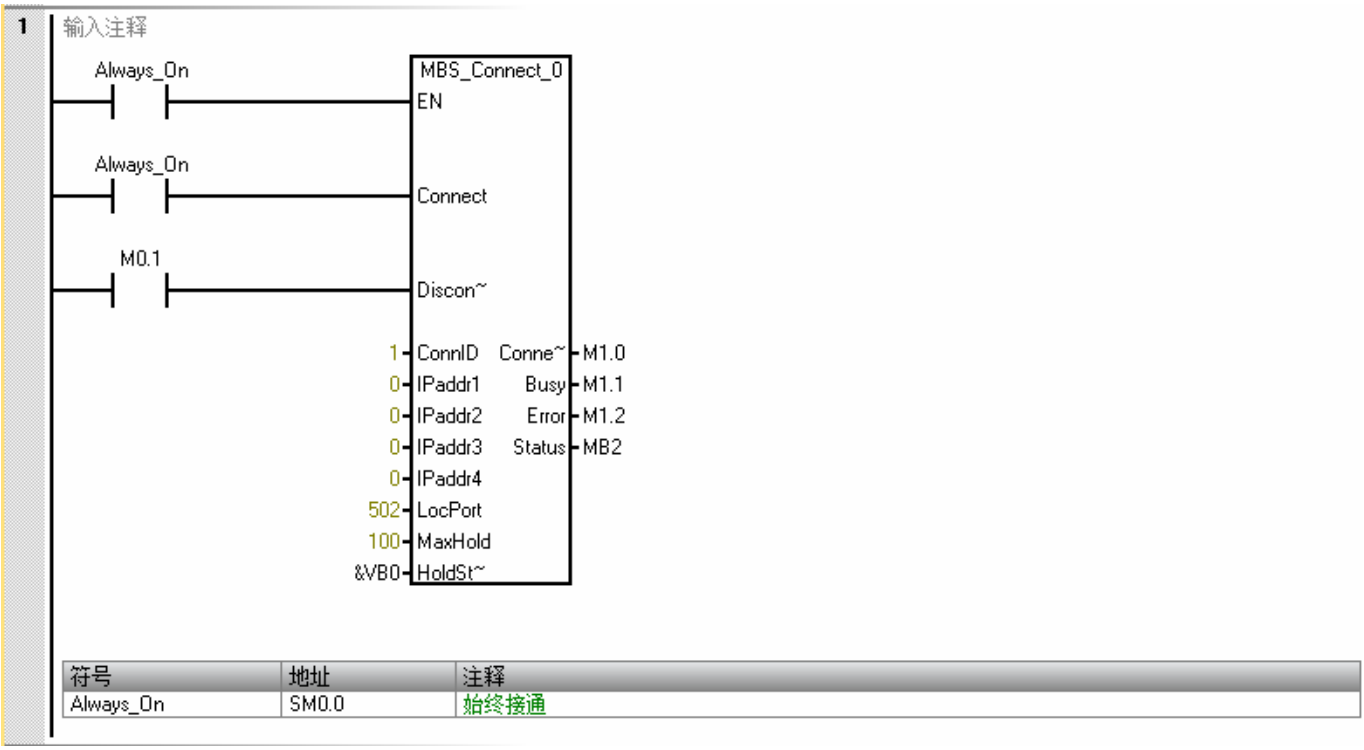
(1)、EN 使能：必须保证每一扫描周期都被使能；

(2)、Done：当 MB_Server 指令响应 Modbus 请求时，Done 完成位在当前扫描周期被设置为 1；如果未处理任何请求，Done 完成位为 0；

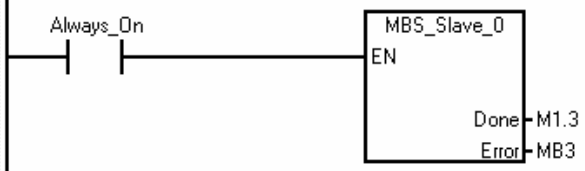
(3)、Error：错误代码，只有在 Done 位为 1 时错误代码有效。

八、西门子 200 Smart 实现 Modbus TCP 服务器编程：

调用 MB_Server0 指令库编制的程序如下所示：

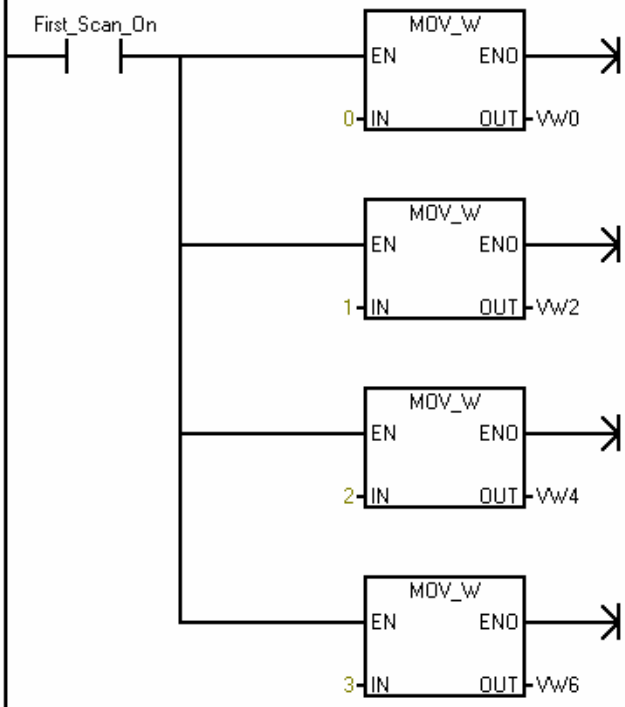


2 输入注释



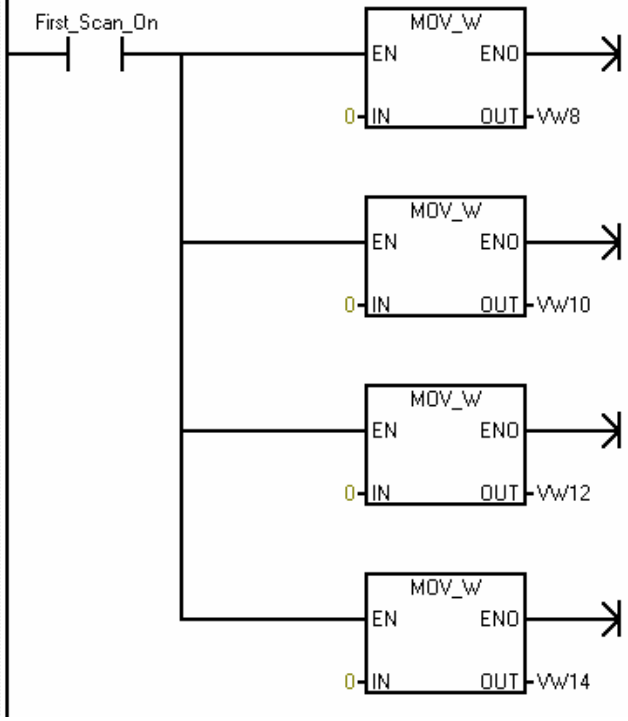
符号	地址	注释
Always_On	SM0.0	始终接通

3 输入注释



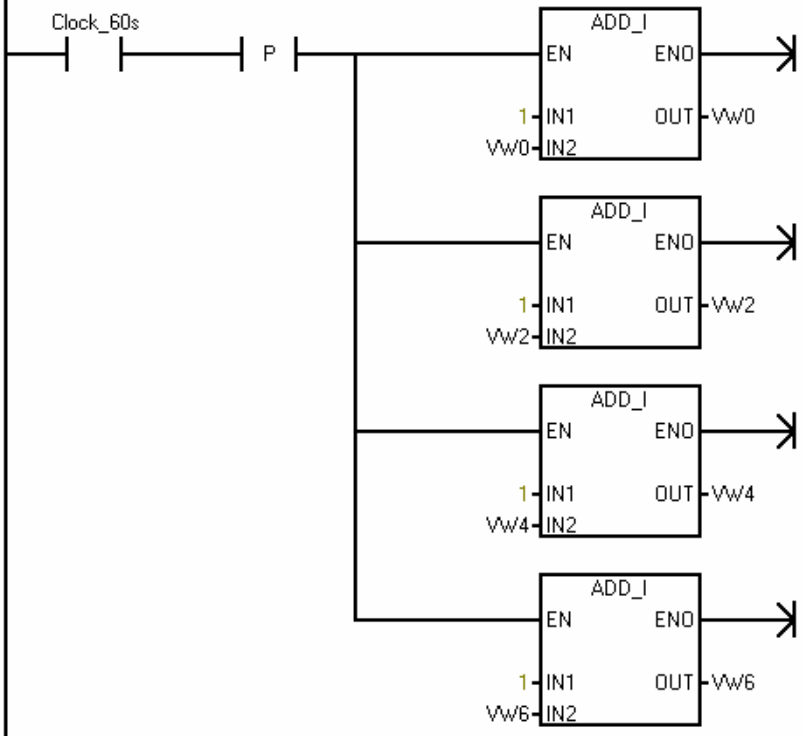
符号	地址	注释
First_Scan_On	SM0.1	仅在第一个扫描周期时接通

4 输入注释

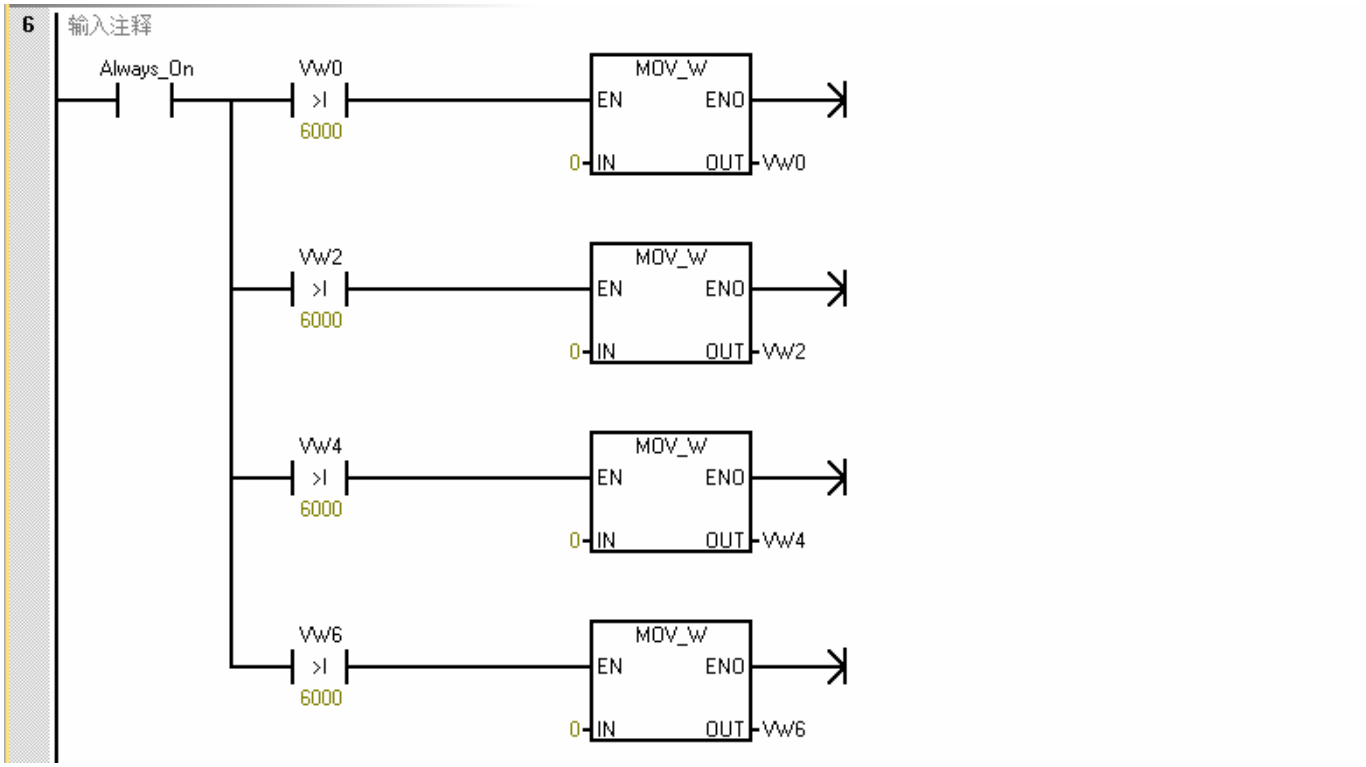


符号	地址	注释
First_Scan_On	SM0.1	仅在第一个扫描周期时接通

5 输入注释



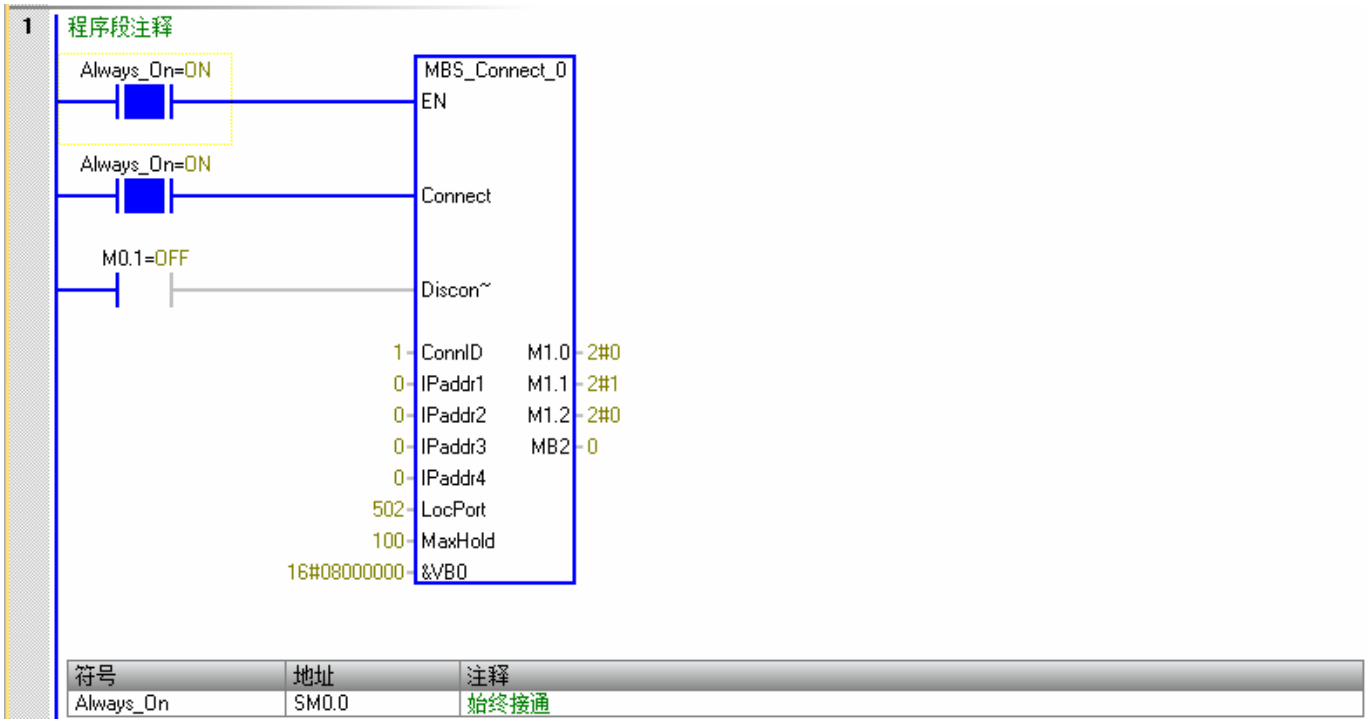
符号	地址	注释
Clock_60s	SM0.4	针对 1 分钟的周期时间，时钟脉冲接通 30 s，断开 30 s.



符号	地址	注释
First_Scan_On	SM0.1	仅在第一个扫描周期时接通

九、以太网调试助手和西门子 200 Smart 通信连接：

1、下载程序，并打开程序状态监控、图标状态监控，如下所示：

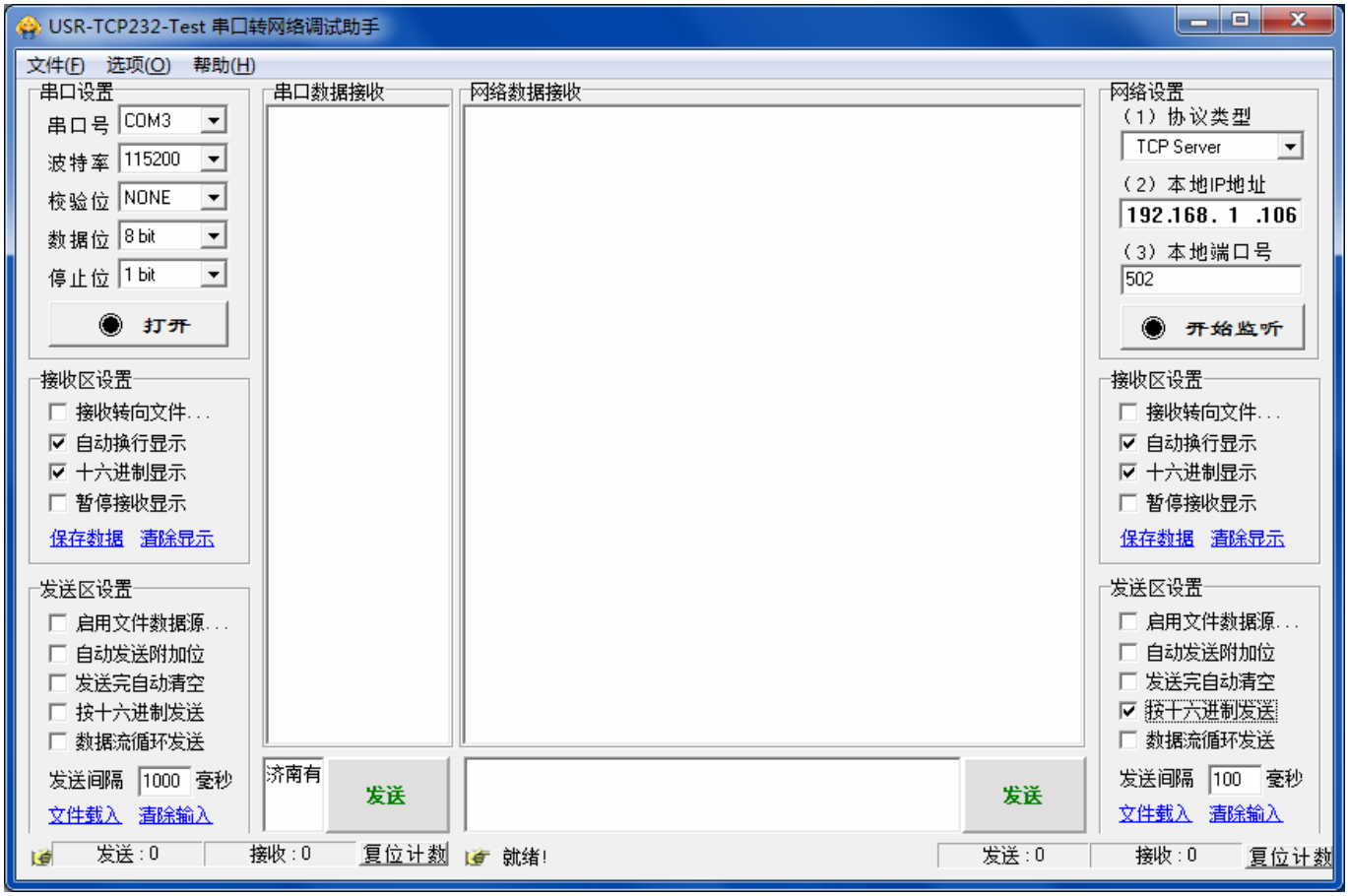


符号	地址	注释
Always_On	SM0.0	始终接通

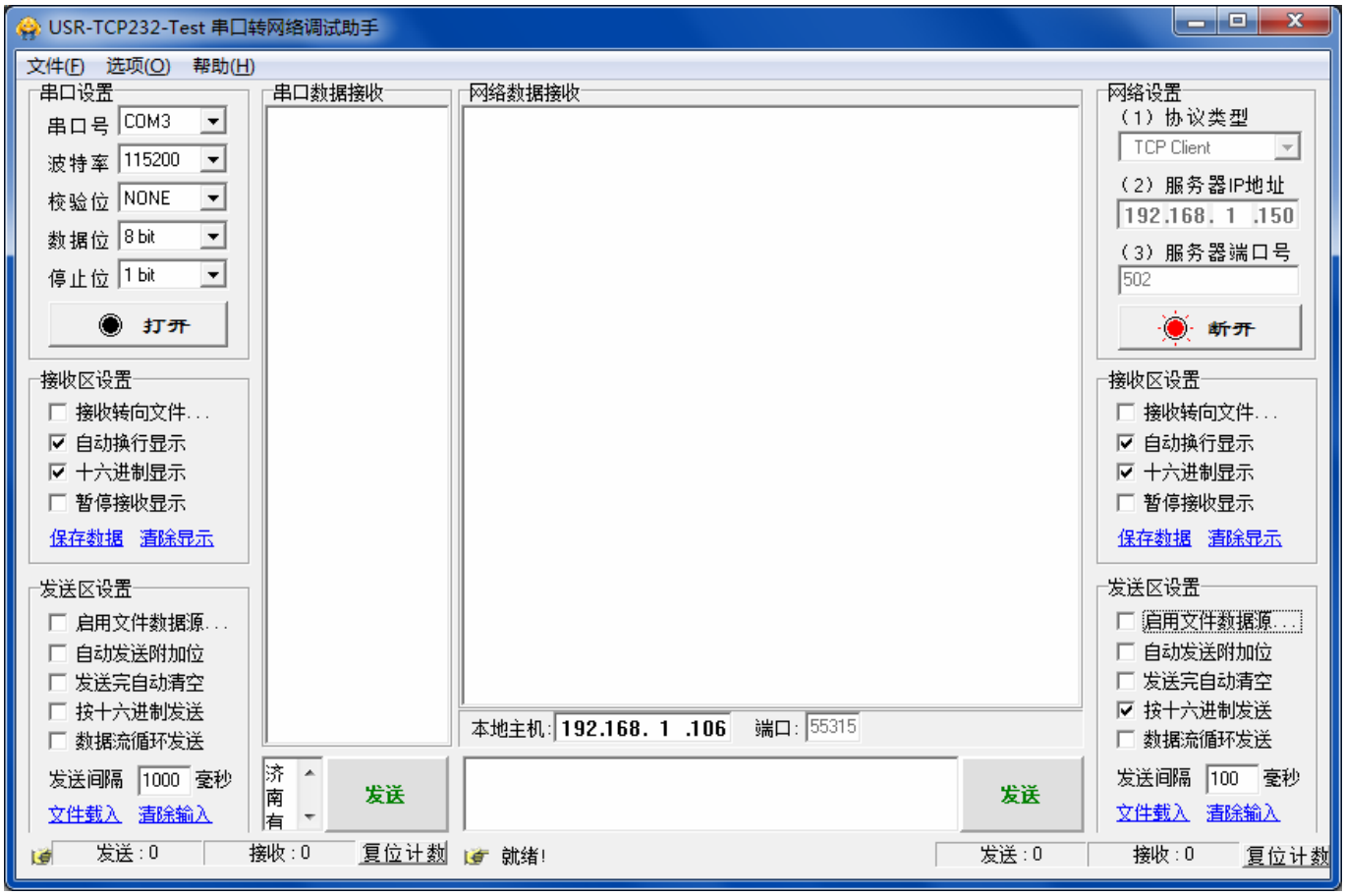
表明暂未有 Modbus TCP 客户端去连接。

2、使用以太网调试助手连接西门子 200 Smart：

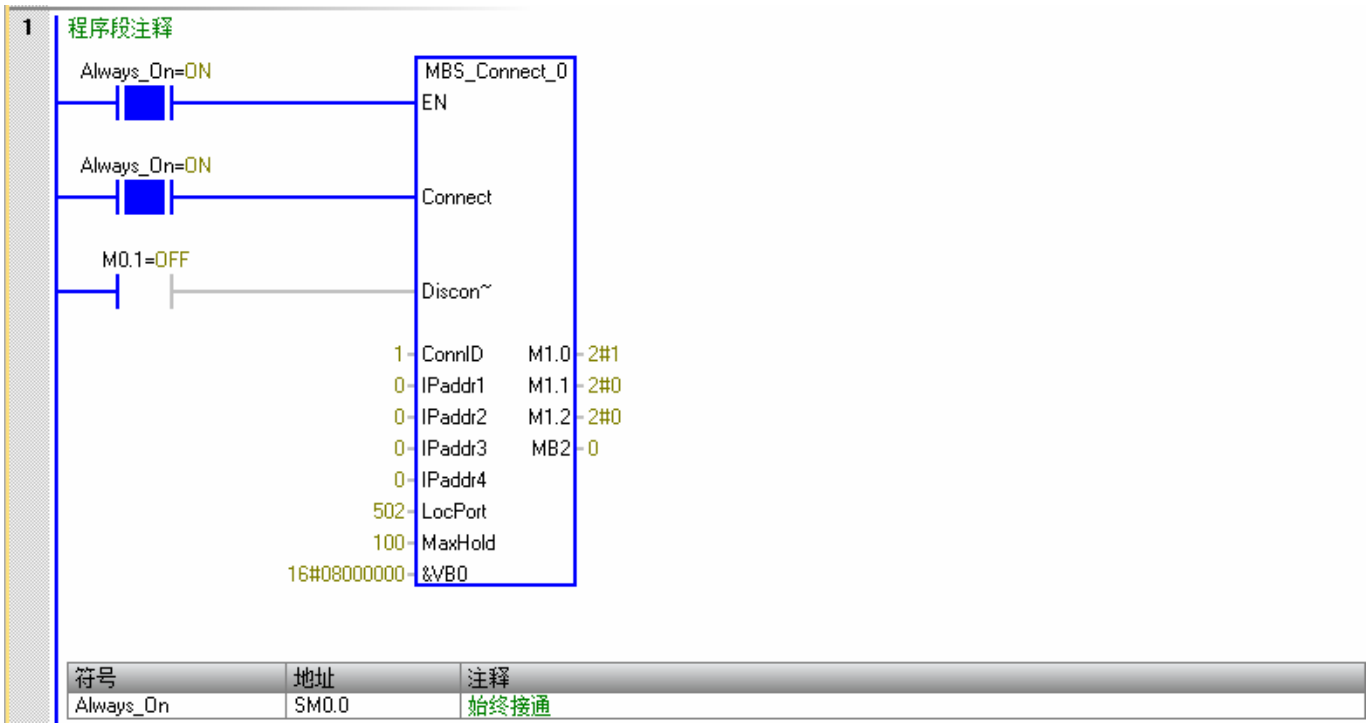
打开以太网调试助手，如下所示：

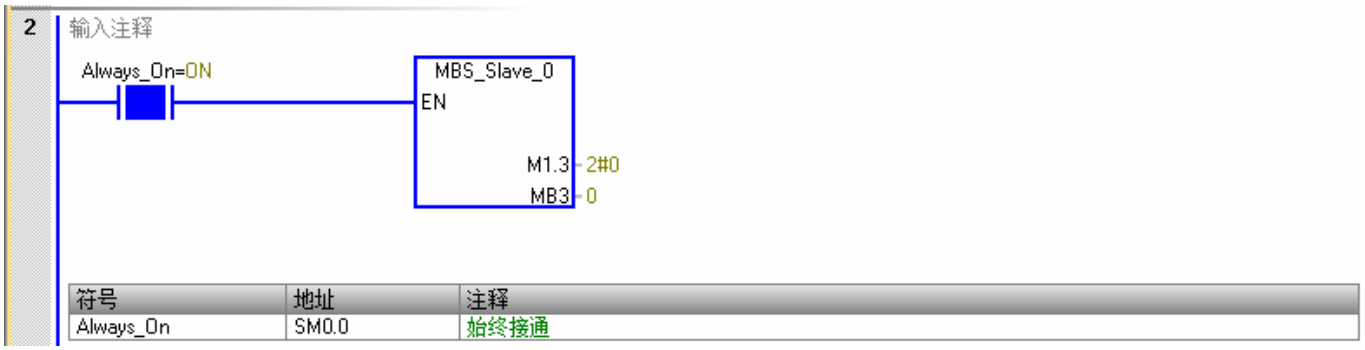


协议类型选择“TCP Client”、服务器 IP 地址设置为 192.168.1.150、服务器端口号设置为 502，点击连接，如下所示：



表明此时以太网调试助手已经成功连接到西门子 200 Smart。此时再去监控西门子 200 Smart 程序，如下所示：





表明已经成功连接。

十、使用以太网调试助手测试西门子 200 Smart 之 Modbus TCP 服务器程序：

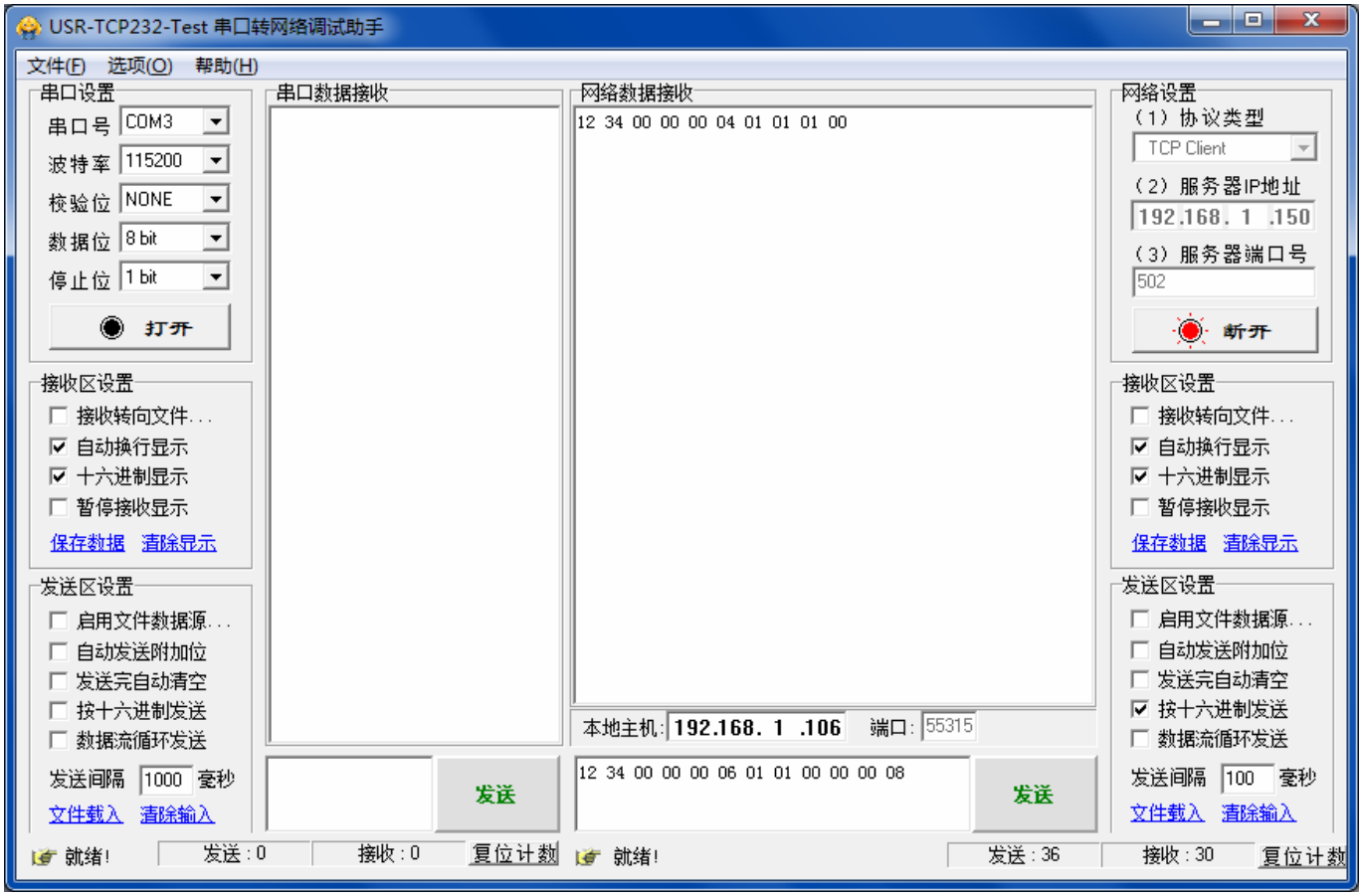
1、Modbus 之 01 功能码测试，读取西门子 200 Smart 之 Q0.0~Q0.7 状态：

以太网调试助手发送：12 34 00 00 00 06 01 01 00 00 00 08

西门子 200 Smart 返回：12 34 00 00 00 04 01 01 01 00

返回数据为 00，表明此时 Q0.0~Q0.7 状态全为 0

以太网调试收发数据、西门子 200 Smart 图表监控如下所示，Modbus 之 01 功能码测试完成：



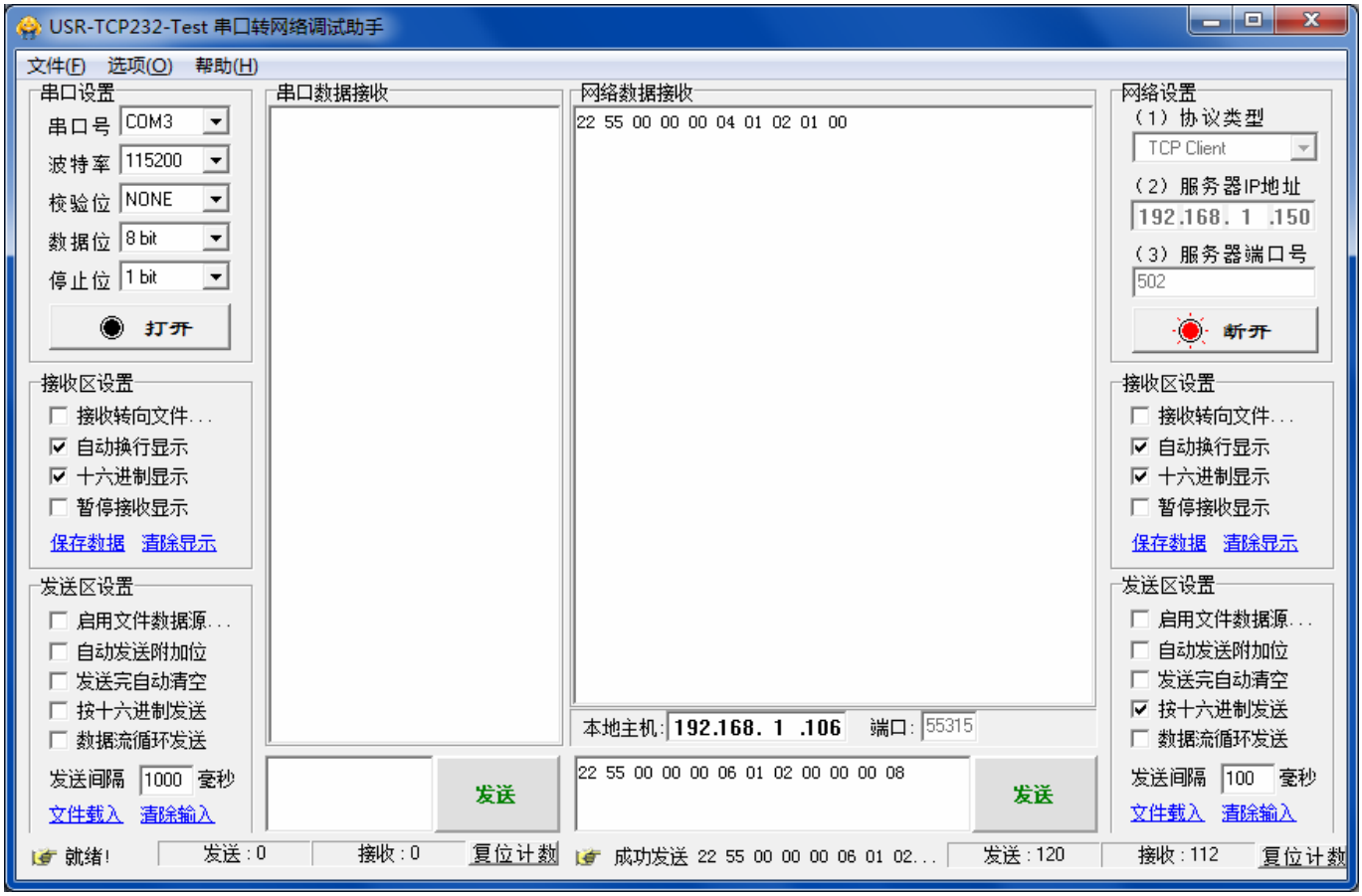
2、Modbus 之 02 功能码测试，读取西门子 200 Smart 之 I0.0~I0.7 状态：

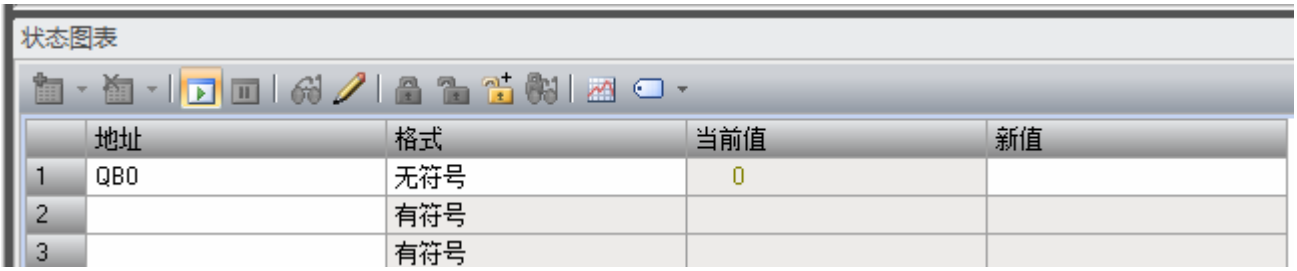
以太网调试助手发送：**22 55 00 00 00 06 01 02 00 00 00 08**

西门子 200 Smart 返回：**22 55 00 00 00 04 01 02 01 00**

返回数据为 00，表明此时 I0.0~I0.7 状态全为 0

以太网调试收发数据、西门子 200 Smart 图表监控如下所示，Modbus 之 02 功能码测试完成：





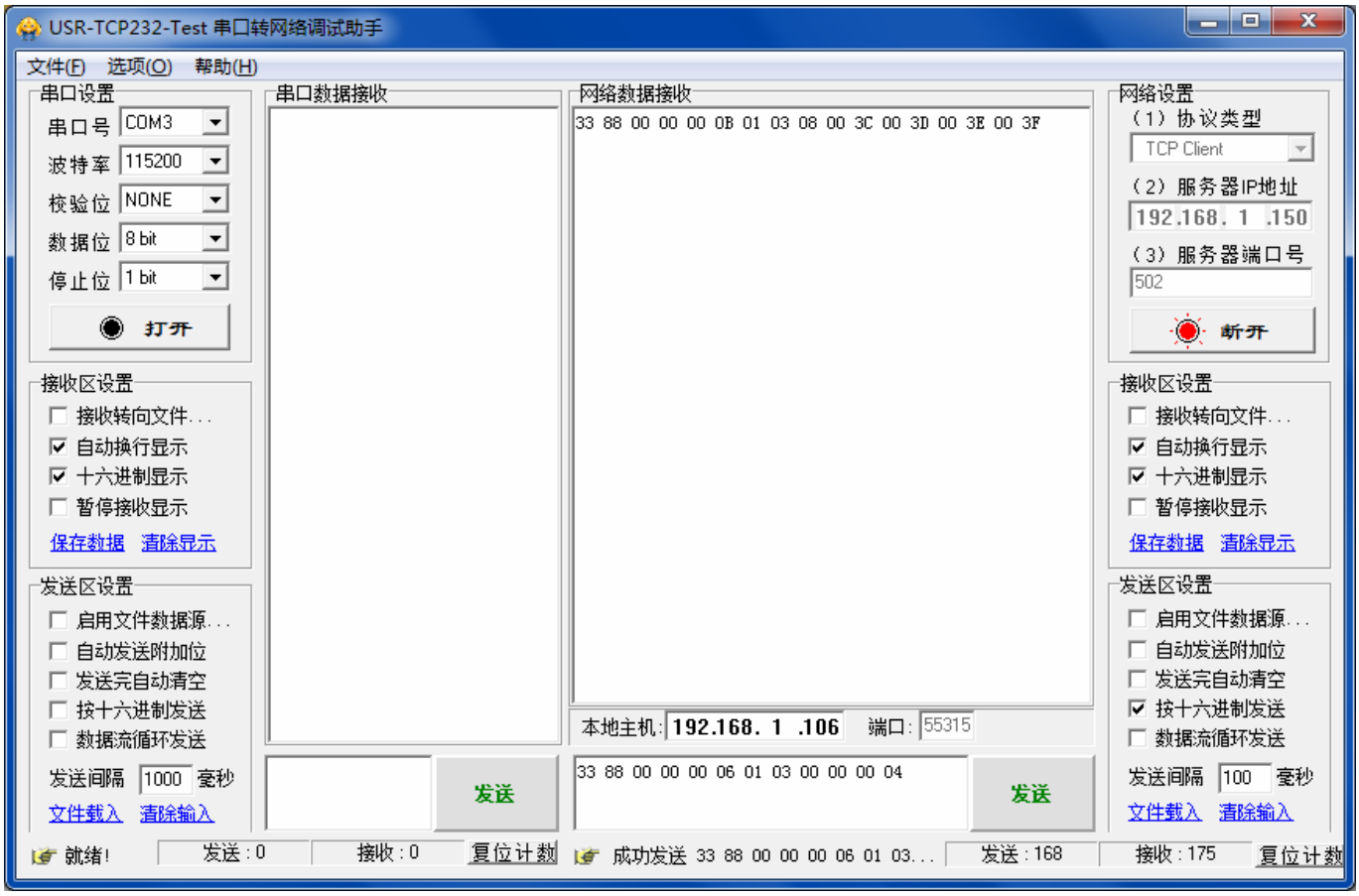
3、Modbus 之 03 功能码测试，读取西门子 200 Smart 之 VW0~VW6 状态：

以太网调试助手发送：33 88 00 00 00 06 01 03 00 00 00 04

西门子 200 Smart 返回：33 88 00 00 00 0B 01 03 08 00 3C 00 3D 00 3E 00 3F

返回数据为依次为 00 3A 00 3B 00 3C 00 3D，表明 VW0、VW2、VW4、VW6 的值依次为 16 进制 003C、003D、003E、003F

以太网调试收发数据、西门子 200 Smart 图表监控如下所示，Modbus 之 03 功能码测试完成：



状态图表

	地址	格式	当前值	新值
1	VW0	十六进制	16#003C	
2	VW2	十六进制	16#003D	
3	VW4	十六进制	16#003E	
4	VW6	十六进制	16#003F	
5		有符号		
6		有符号		

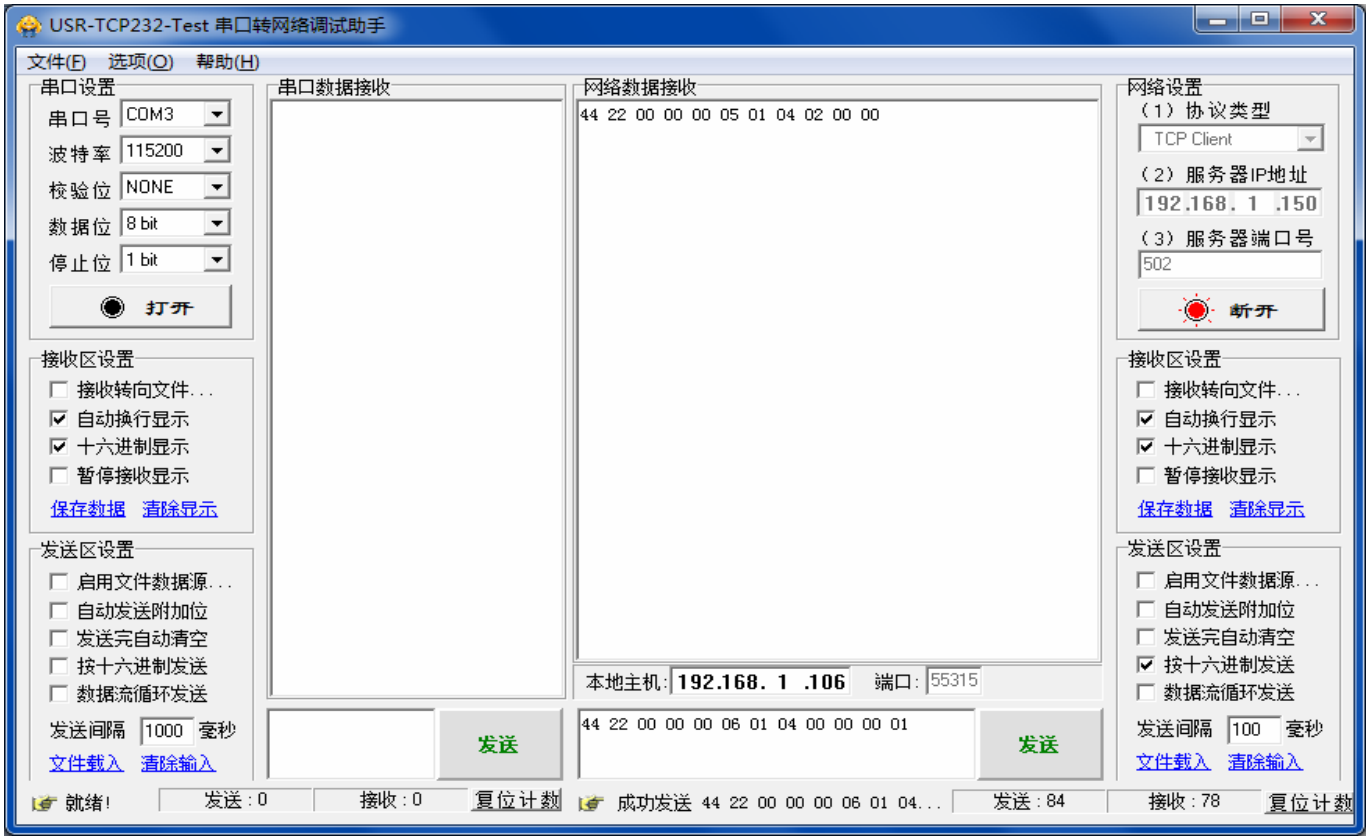
4、Modbus 之 04 功能码测试，读取西门子 200 Smart 之 AIW0 状态：

以太网调试助手发送：44 22 00 00 00 06 01 04 00 00 00 01

西门子 200 Smart 返回：44 22 00 00 00 05 01 04 02 00 00

返回数据为依次为 00 00，表明 AIW0 为 16 进制 0000

以太网调试收发数据、西门子 200 Smart 图表监控如下所示，Modbus 之 04 功能码测试完成：



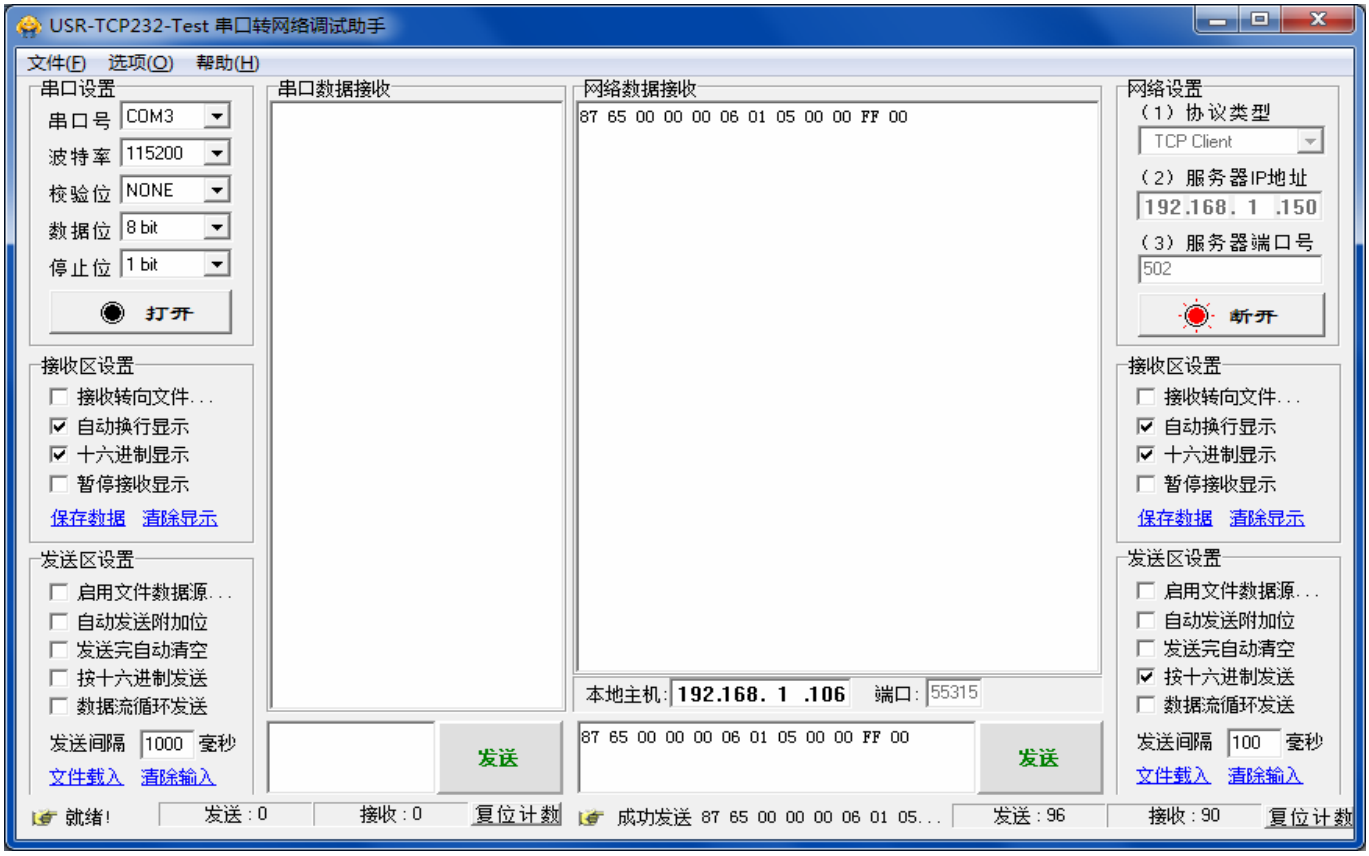
5、Modbus 之 05 功能码测试，将西门子 200 Smart 之 Q0.0 置位

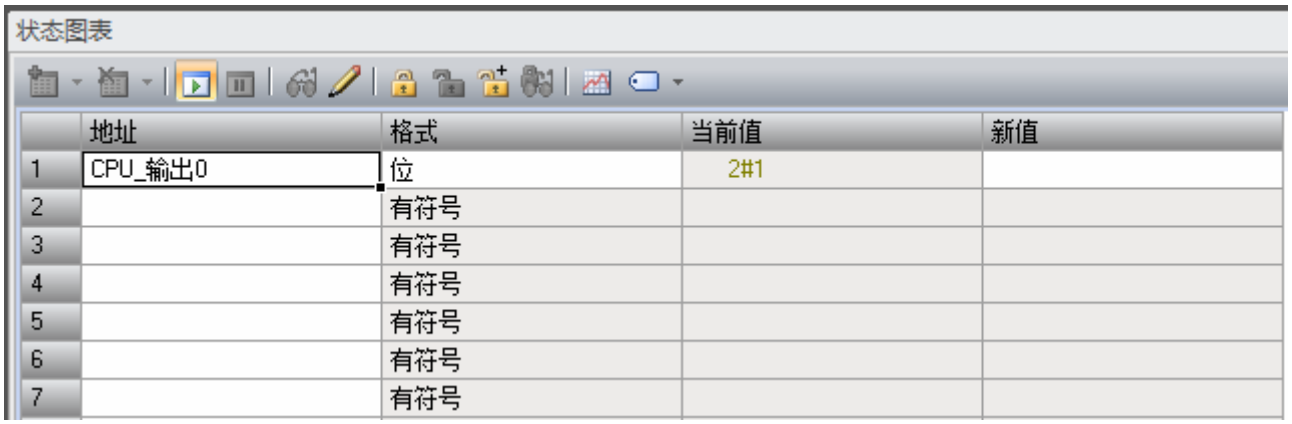
以太网调试助手发送：87 65 00 00 00 06 01 05 00 00 FF 00

西门子 200 Smart 返回：87 65 00 00 00 06 01 05 00 00 FF 00

西门子 200 Smart 原样返回，Q0.0 置位成功

以太网调试收发数据、西门子 200 Smart 图表监控如下所示，Modbus 之 05 功能码测试完成：





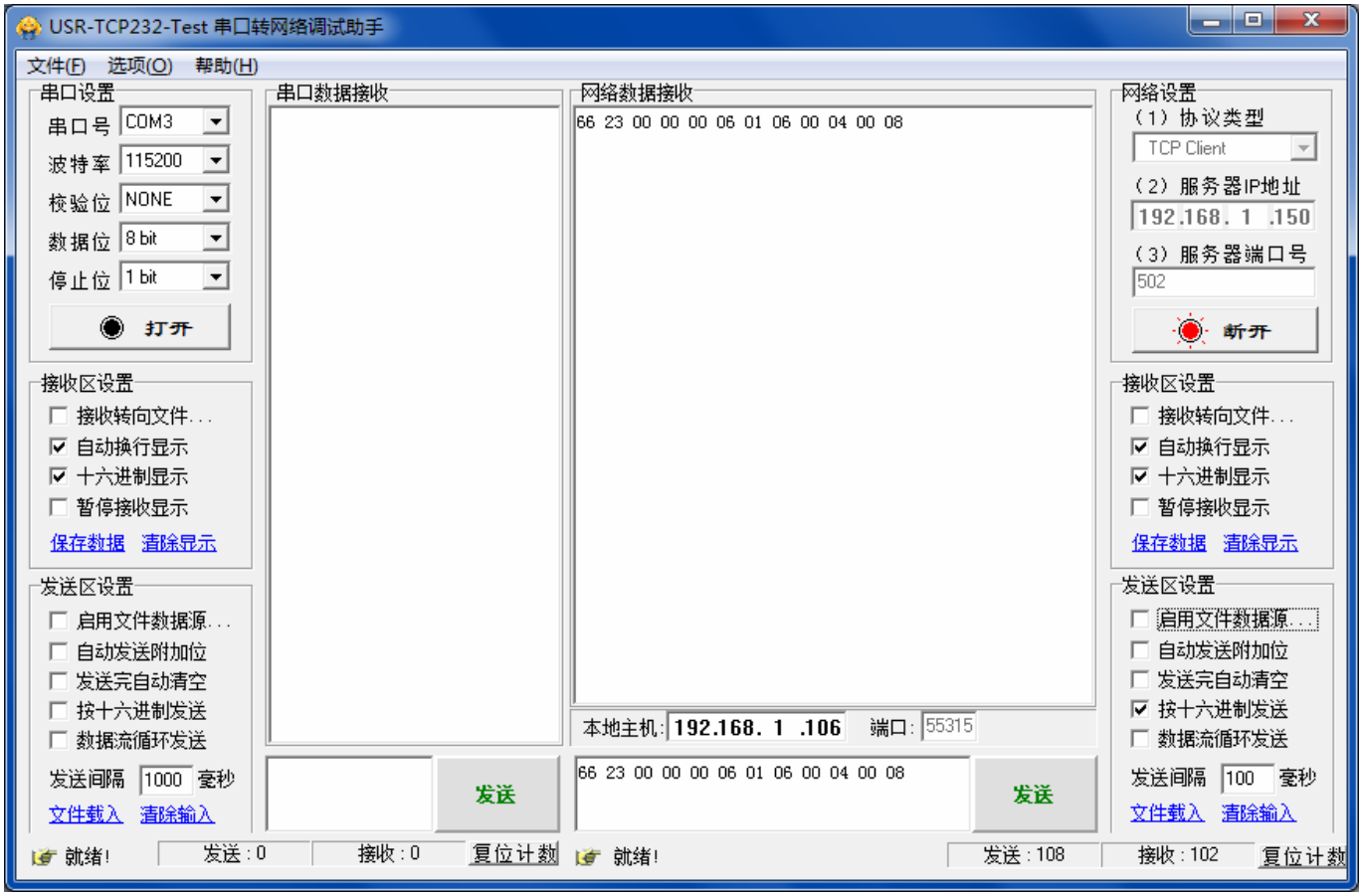
6、Modbus 之 06 功能码测试，将西门子 200 Smart 之 VW8 数据设置为 8

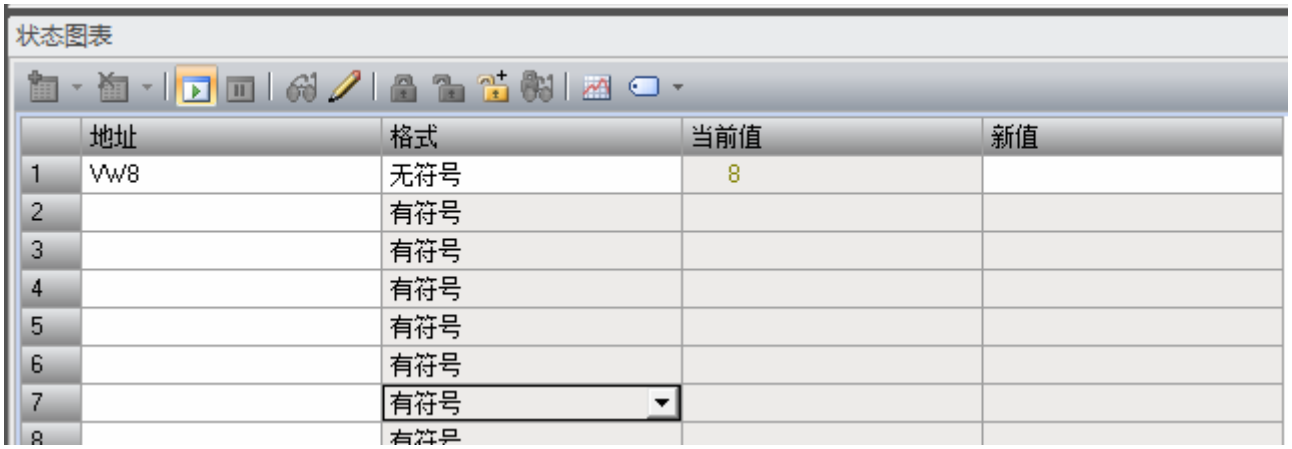
以太网调试助手发送：66 23 00 00 00 06 01 06 00 04 00 08

西门子 200 Smart 返回：66 23 00 00 00 06 01 06 00 04 00 08

西门子 200 Smart 原样返回，VW8 数据设置成功

以太网调试收发数据、西门子 200 Smart 图表监控如下所示，Modbus 之 06 功能码测试完成：





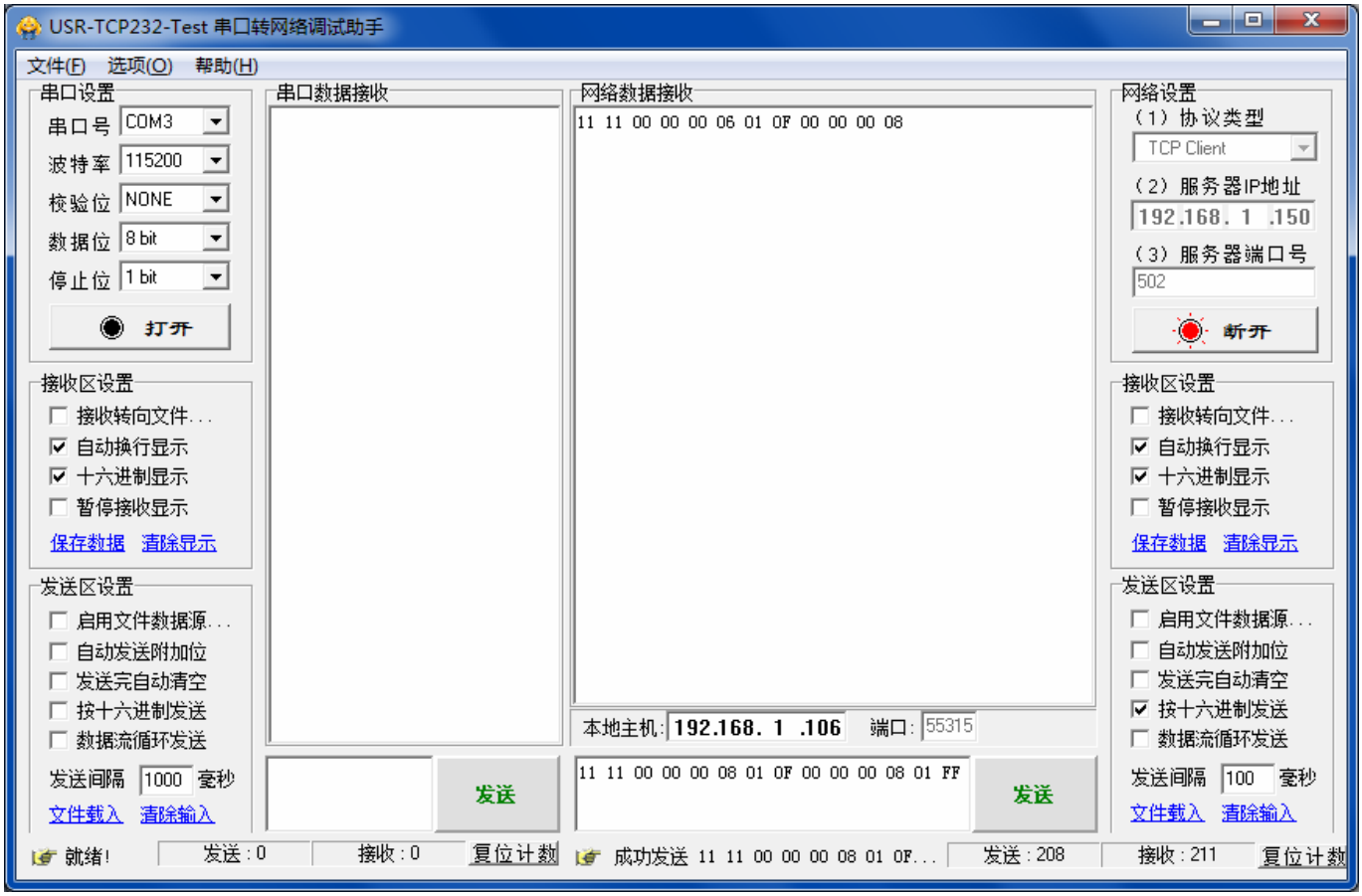
7、Modbus 之 15 功能码测试，将西门子 200 Smart 之 Q0.0~Q0.7 全部置位

以太网调试助手发送：11 11 00 00 00 08 01 0F 00 00 00 08 01 FF

西门子 200 Smart 返回：11 11 00 00 00 06 01 0F 00 00 00 08

西门子 200 Smart 返回数据表明 Q0.0~Q0.7 置位操作设置成功

以太网调试收发数据、西门子 200 Smart 图表监控如下所示，Modbus 之 15 功能码测试完成：



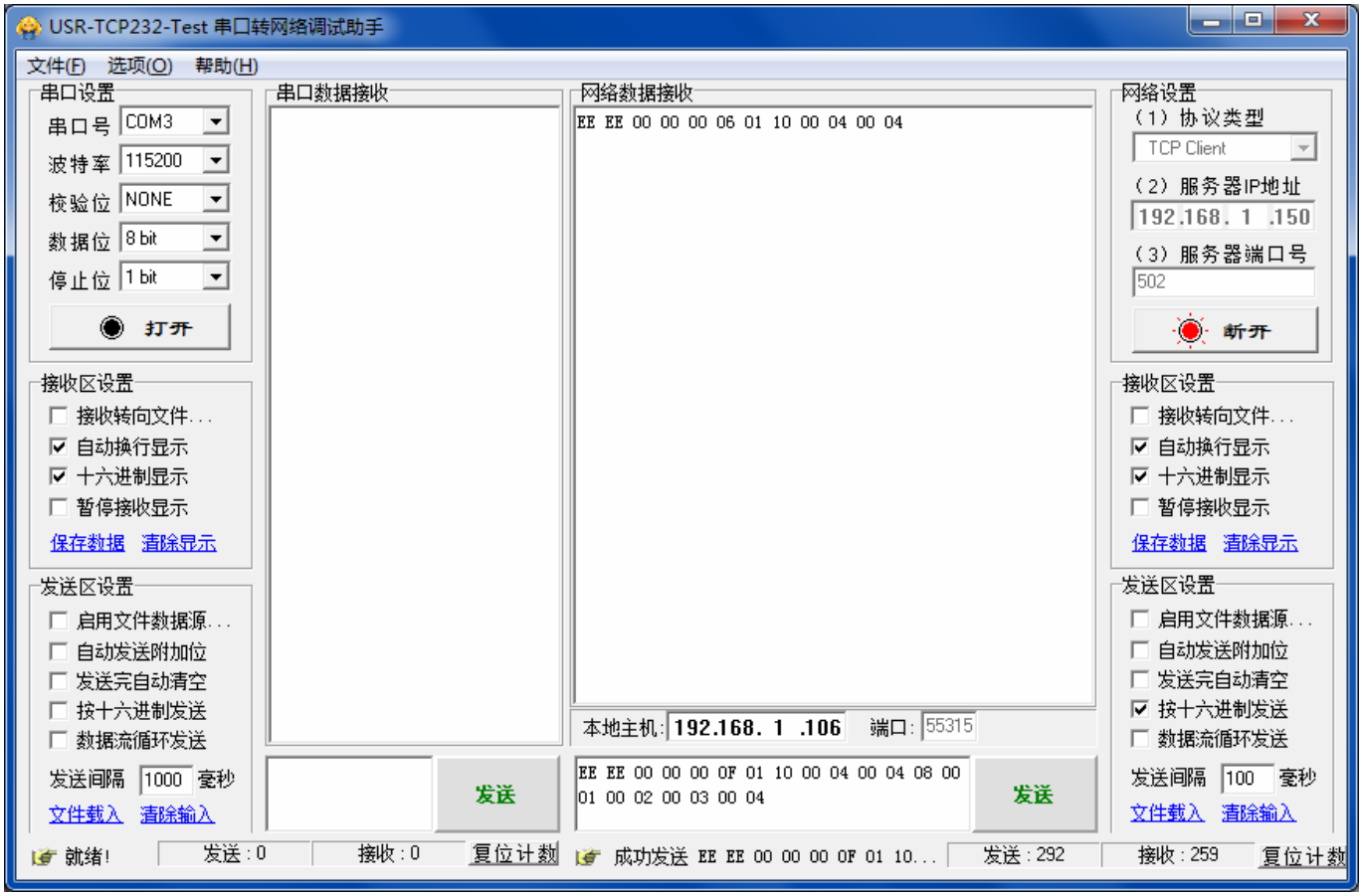
8、Modbus 之 16 功能码测试，将西门子 200 Smart 之 VW8~VW14 依次设置为 1、2、3、4

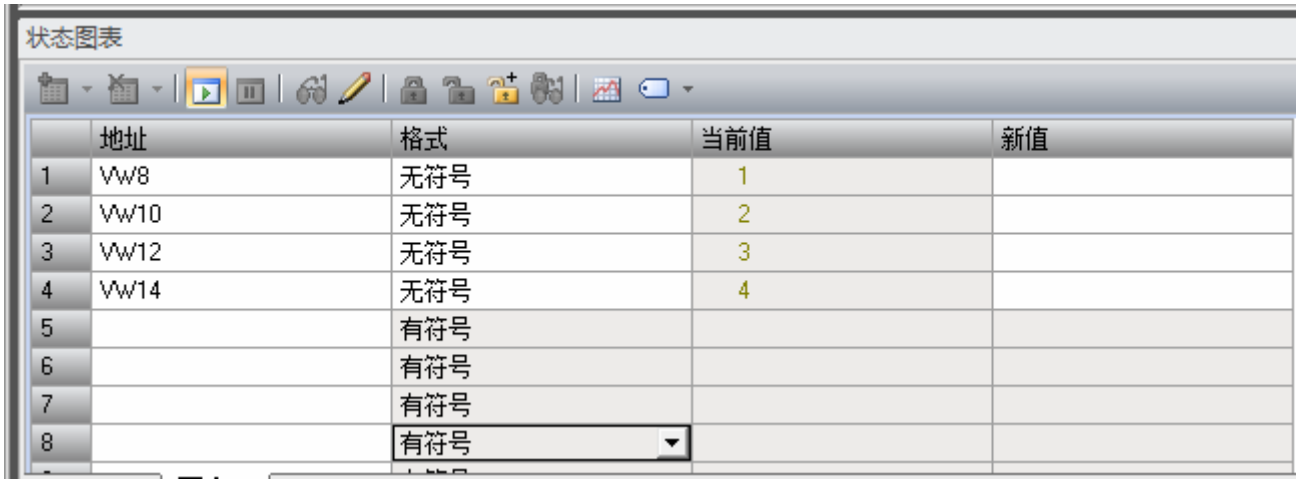
以太网调试助手发送：EE EE 00 00 00 0F 01 10 00 04 00 04 08 00 01 00 02 00 03 00 04

西门子 200 Smart 返回：EE EE 00 00 00 06 01 10 00 04 00 04

西门子 200 Smart 返回数据表明 VW8~VW14 数据设置成功

以太网调试收发数据、西门子 200 Smart 图表监控如下所示，Modbus 之 16 功能码测试完成：



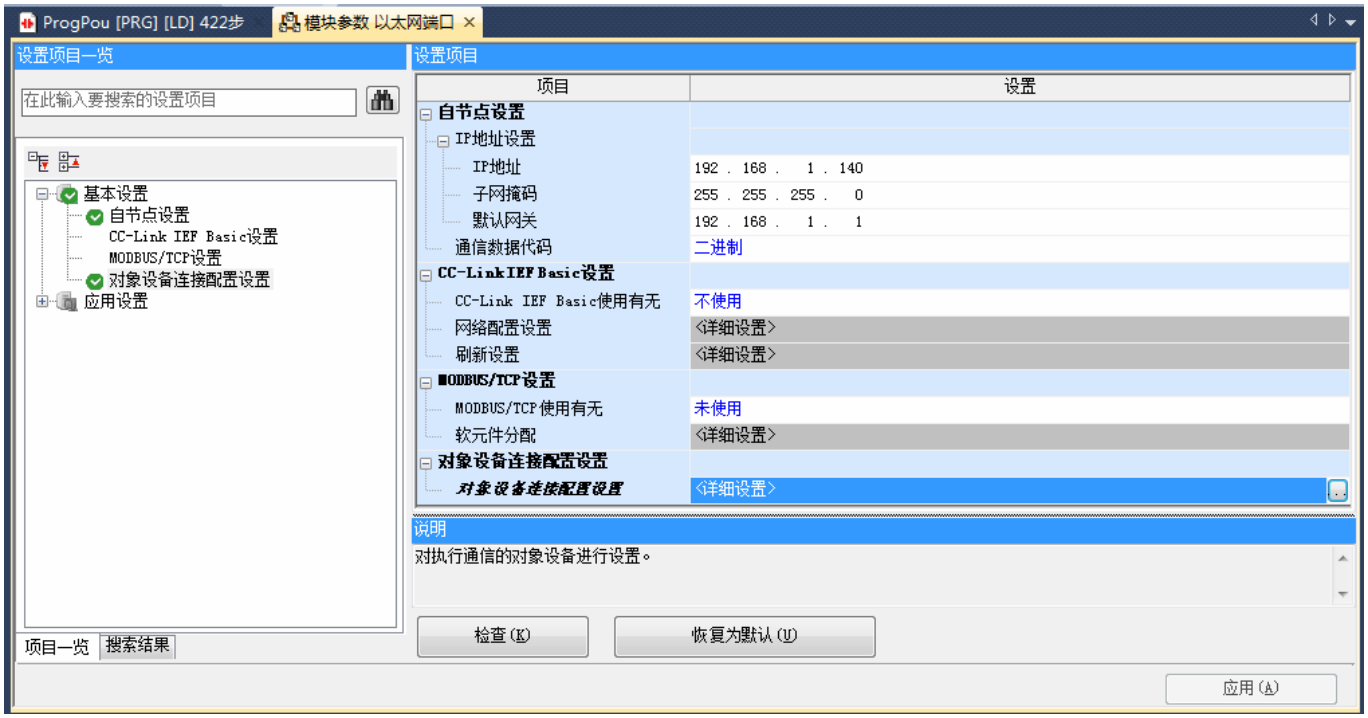


9、测试总结：

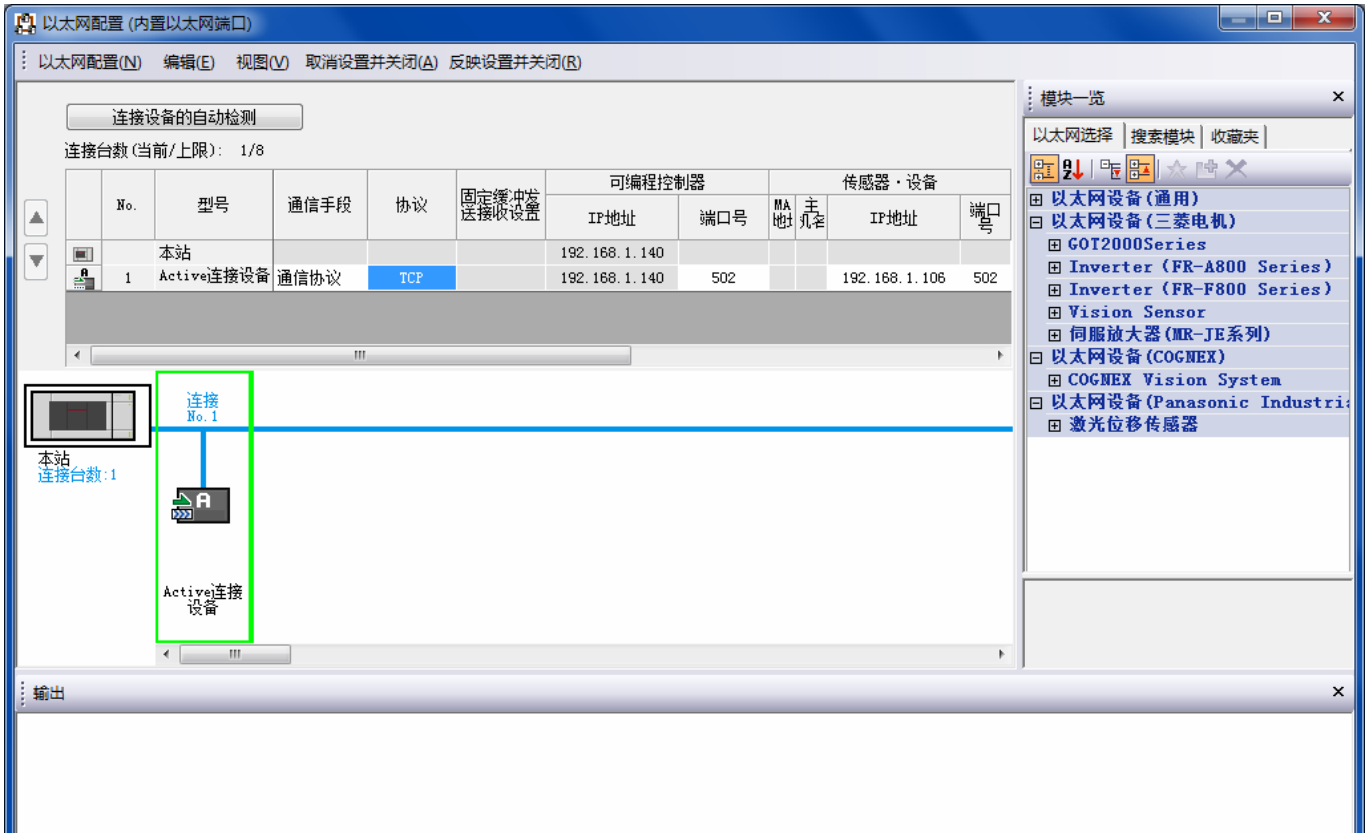
以上 Modbus 共计 8 个功能码测试通过，亦表明西门子 200 Smart 之 Modbus TCP 服务器程序正确无误。此时可以断开以太网调试助手和西门子 200 Smart 的通信连接。

十一、三菱 FX5U 通信参数设置：

1、以太网端口自接点设置，需要设置 IP 地址、子网掩码、默认网关、通信数据代码等诸多选型，如下所示：



2、对象设备连接配置设置，拖入一个 Active 连接设备，通信手段选择“通信协议”、可编程控制的 IP 地址设置为 192.168.1.140、可编程控制器端口号设置为 502；传感器设备 IP 地址暂时设置为 192.168.1.106（此为电脑 IP 地址，方便测试时使用以太网调试助手测试三菱 FX5U 程序，待测试 OK 后再修改为西门子 200 Smart 的 IP 地址）、传感器设备端口号设置为 502，如下所示：



3、三菱 FX5U 以太网端口通信协议支持功能数据包建立：

协议号 1，功能码为 02，用来读取 Modbus TCP 服务器多路输入

协议号 2，功能码为 15，用来写入 Modbus TCP 服务器多路线圈

协议号 3，功能码为 03，用来读取 Modbus TCP 服务器多路保持寄存器

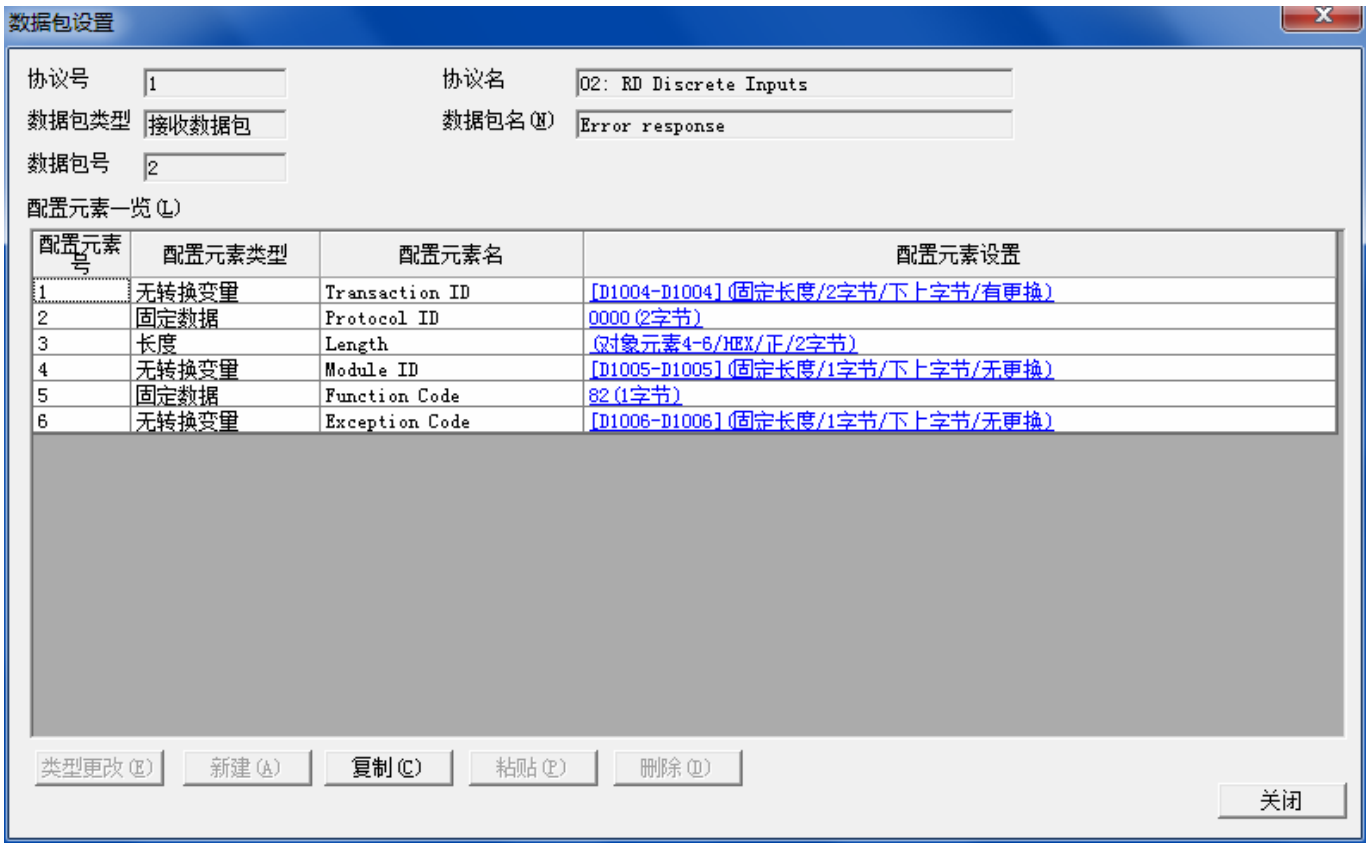
协议号 4，功能码为 16，用来写入 Modbus TCP 服务器多路保持寄存器

(1)、协议号 1 详细设置如下所示：

发送，占用寄存器 D1000~D1003，如下所示：

正确返回，占用寄存器 D1007~D1010，如下所示：

错误返回，占用寄存器 D1004~D1006，如下所示：



(2)、协议号 2 详细设置如下所示：

发送，占用寄存器 D1107~D1112，如下所示：

正确返回，占用寄存器 D1100~D1103，如下所示：

错误返回，占用寄存器 D1104~D1106，如下所示：

(3)、协议号 3 详细设置如下所示：

发送，占用寄存器 D1200~D1203，如下所示：

正确返回，占用寄存器 D1207~D1213，如下所示：

错误返回，占用寄存器 D1204~D1206，如下所示：

(4)、协议号 4 详细设置如下所示：

发送，占用寄存器 D1307~D1315，如下所示：

正确返回，占用寄存器 D1300~D1303，如下所示：

错误返回，占用寄存器 D1304~D1306，如下所示：

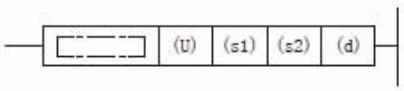
十二、三菱 FX5U 实现 Modbus TCP 客户端相关指令：

1、SP.SOCOPEN 指令：

指令格式如下所示

SP.SOCOPEN

建立连接。

梯形图	ST
	ENO:=SP_SOCOPEN(EN, U0, s1, s2, d);

指令参数说明如下所示

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(U)*1	虚拟(应输入字符串“‘U0’”。)	—	字符串	ANYSTRING_SINGLE
(s1)	连接编号	1~8	无符号BIN16位	ANY16
(s2)	存储控制数据的软元件起始编号	请参考控制数据 (☞ 80页)	字	ANY16_ARRAY (要素数：10)
(d)	指令结束时，1个扫描为ON的软元件起始编号 异常完成时(d)+1也变为ON。	—	位	ANYBIT_ARRAY (要素数：2)

其中操作数 S2 含义如下所示

■控制数据

软元件	项目	内容	设置范围	设置侧*1
(s2)+0	执行型/结束型	指定在连接的开放处理时，是使用通过工程工具设置的参数设置值还是使用控制数据 (s2)+2～(s2)+9的设置值。 0000H: 通过工程工具的“对方设备连接构成设置”中设置的内容进行开放处理。 8000H: 通过在控制数据(s2)+2～(s2)+9中指定的内容进行开放处理。	0000H 8000H	用户
(s2)+1	结束状态	存储完成时的状态。 0000H：正常结束 0000H以外：异常结束 (出错代码) 关于出错代码，请参考□ 118页 出错代码	—	系统
(s2)+2	使用用途设置区域	b15 b14 b13～b11 b10 b9 b8 b7 ～ b0 (s2)+2 [4] 0 [3][2][1] 0 [1]通信方式 (协议) 0：TCP/IP 1：UDP/IP [2]套接字通信功能的有序无序 0：通信协议 1：套接字通信 (无顺序) [3]通信协议设置 0：不使用通信协议功能 (使用套接字通信功能) 1：使用通信协议功能 [4]开放方式 00：Active开放或UDP/IP 10：Unpassive开放 11：Fullpassive开放	如左所示	用户
(s2)+3	本站端口编号	指定本站的端口编号。	1～5548，5570～65534 (0001H～15ACH，15C2H～FFFEH) *3	
(s2)+4 (s2)+5	对方设备IP地址*2	指定对方设备的IP地址。	1～3758096382 (00000001H～DFFFFFFEH)	
(s2)+6	对方设备端口编号*2	指定对方设备的端口编号。	1～65534 (0001H～FFFEH)	
(s2)+7～ (s2)+9	—	禁止使用	—	系统

编程举例如下所示：

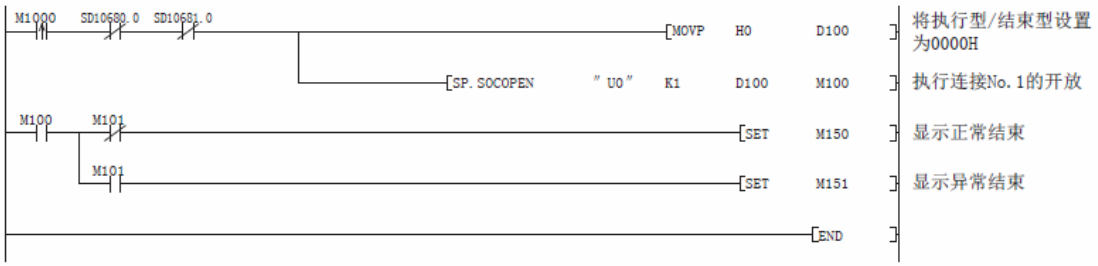
■使用参数设置值执行开放时

将M1000置ON时，使用“对象设备连接配置设置”开放连接No.1的程序。

• 使用的软元件

软元件编号	用途
SD10680	开放结束信号
SD10681	开放请求信号
D100	SP.SOCOPEN指令控制数据
M100	SP.SOCOPEN指令结束软元件

• 程序

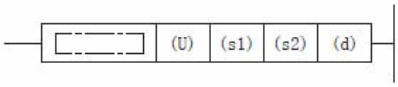


2、SP.SOCCLOSE 指令：

指令格式如下所示：

SP. SOCCLOSE

切断连接。

梯形图	ST
	ENO:=SP_SOCCLOSE(EN, U0, s1, s2, d);

指令参数说明如下所示

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(U)*1	虚拟(应输入字符串“‘U0’”。)	—	字符串	ANYSTRING_SINGLE
(s1)	连接编号	1～8	无符号BIN16位	ANY16
(s2)	存储控制数据的软元件起始编号	请参考控制数据 (☞83页)	字	ANY16_ARRAY (要素数：2)
(d)	指令结束时，1个扫描为ON的软元件起始编号 异常结束时，(d)+1也为ON。	—	位	ANYBIT_ARRAY (要素数：2)

其中操作数 S2 的含义如下所示

软元件	项目	内容	设置范围	设置侧*1
(s2)+0	系统区域	—	—	—
(s2)+1	结束状态	存储结束时的状态。 0000H：正常结束 0000H以外：异常结束（出错代码） 关于出错代码，请参考☞118页 出错代码	—	系统

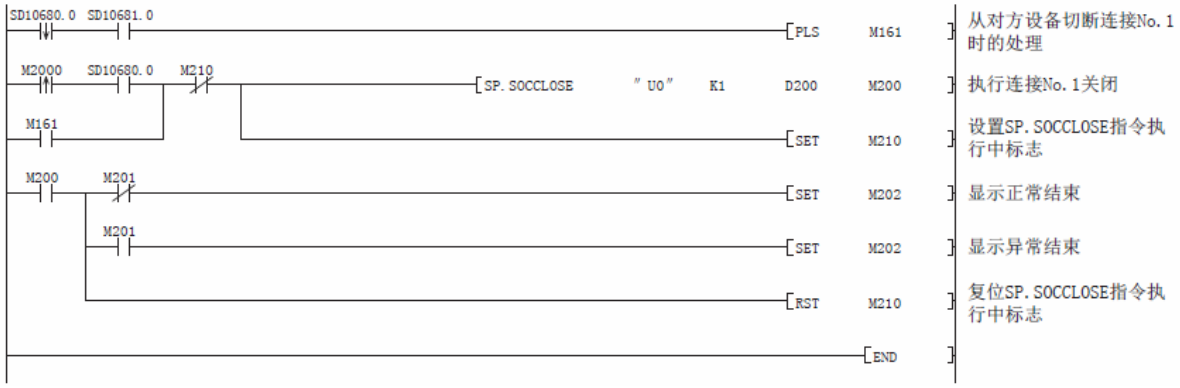
编程举例如下所示：

在将M2000置ON或从对方设备切断了连接No.1时对连接No.1进行切断的程序。

- 使用的软元件

软元件编号	用途
SD10680	开放结束信号
SD10681	开放请求信号
D200	SP.SOCCLOSE指令控制数据
M200	SP.SOCCLOSE指令结束软元件

- 程序

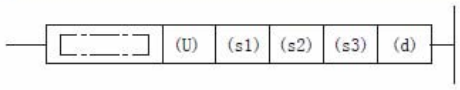


3、SP.ECPRTCL 指令：

指令格式如下所示

SP. ECPRTCL

通过内置以太网执行工程工具中登录的通信协议。

梯形图	ST
	ENO:=SP_ECPRTCL(EN,U0,s1,s2,s3,d);

指令参数说明如下所示

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(U)*1	虚拟(应输入字符串“‘U0’”。)	—	字符串	ANYSTRING_SINGLE
(s1)	连接编号	1～8	无符号BIN16位	ANY16
(s2)	连续执行的协议数	1～8	无符号BIN16位	ANY16
(s3)	存储控制数据的软元件起始编号	参阅控制数据 (☞60页)	字	ANY16_ARRAY (要素数：18)
(d)	通过指令完成使1个扫描ON的软元件起始编号 异常完成时(d)+1也变为ON。	—	位	ANYBIT_ARRAY (要素数：2)

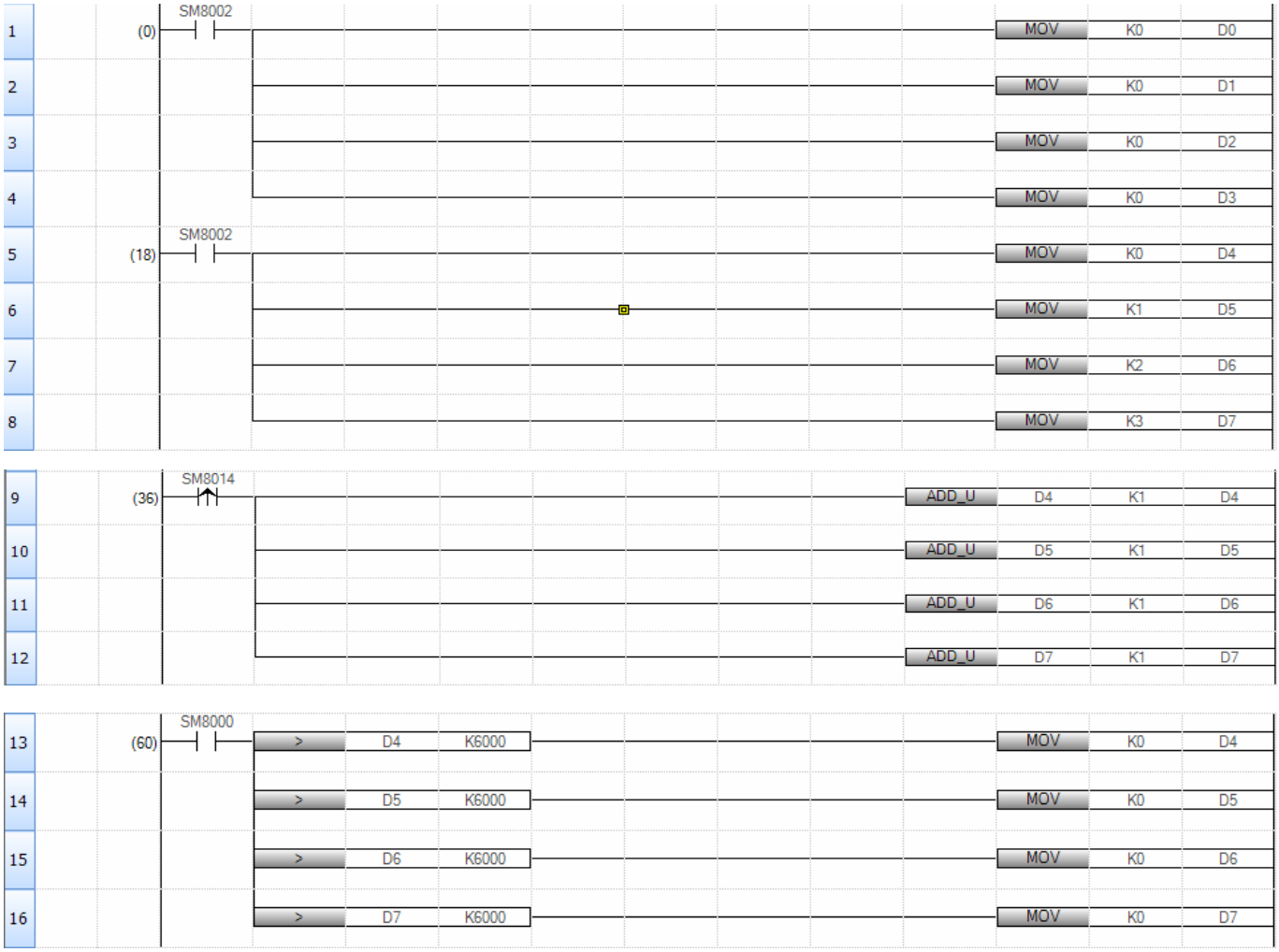
其中操作数 S3 的含义如下所示

软元件	项目	内容	设置范围	设置方*1
(s3)+0	执行数结果	存储通过SP.ECPRTCL指令执行的协议数。发生出错的协议也包含在执行数中。 设置数据、控制数据的设置有错误的情况下将存储“0”。	0、1～8	系统
(s3)+1	完成状态	存储SP.ECPRTCL指令的执行结果。执行多个协议的情况下，最后执行的协议的执行结果将被存储。 0：正常 0以外：异常结束(出错代码)	—	系统
(s3)+2	执行协议编号指定1	指定第1个执行的协议的协议编号。	1～64	用户
(s3)+3	执行协议编号指定2	指定第2个执行的协议的协议编号。	0、1～64	
(s3)+4	执行协议编号指定3	指定第3个执行的协议的协议编号。	0、1～64	
(s3)+5	执行协议编号指定4	指定第4个执行的协议的协议编号。	0、1～64	
(s3)+6	执行协议编号指定5	指定第5个执行的协议的协议编号。	0、1～64	
(s3)+7	执行协议编号指定6	指定第6个执行的协议的协议编号。	0、1～64	
(s3)+8	执行协议编号指定7	指定第7个执行的协议的协议编号。	0、1～64	
(s3)+9	执行协议编号指定8	指定第8个执行的协议的协议编号。	0、1～64	
(s3)+10	校验一致 接收数据包编号1	第1个执行的协议的通信类型中包含接收的情况下，将存储校验一致的接收数据包编号。通信类型为“仅发送”的情况下，将存储“0”。 执行第1个协议时发生了出错的情况下，将存储“0”。	0、1～16	系统
(s3)+11	校验一致 接收数据包编号2	第2个执行的协议的通信类型中包含接收的情况下，将存储校验一致的接收数据包编号。通信类型为“仅发送”的情况下，将存储“0”。 执行第2个协议时发生了出错的情况下，将存储“0”。 执行的协议数不足2个时，将存储“0”。	0、1～16	
(s3)+12	校验一致 接收数据包编号3	第3个执行的协议的通信类型中包含接收的情况下，将存储校验一致的接收数据包编号。通信类型为“仅发送”的情况下，将存储“0”。 执行第3个协议时发生了出错的情况下，将存储“0”。 执行的协议数不足3个时，将存储“0”。	0、1～16	

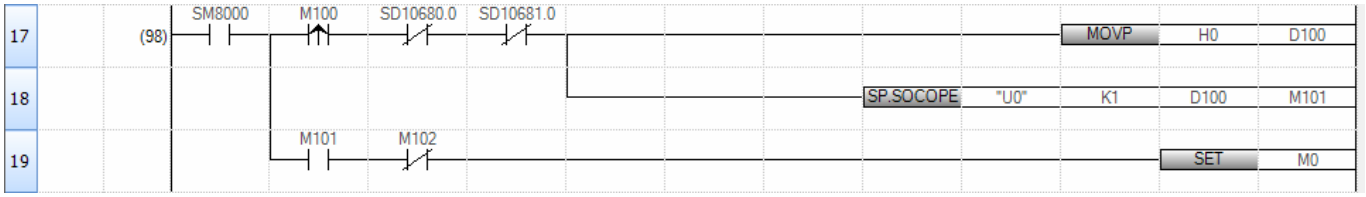
(s3)+13	校验一致 接收数据包编号4	第4个执行的协议的通信类型中包含接收的情况下，将存储校验一致的接收数据包编号。通信类型为“仅发送”的情况下，将存储“0”。 执行第4个协议时发生了出错的情况下，将存储“0”。 执行的协议数不足4个时，将存储“0”。	0、1～16
(s3)+14	校验一致 接收数据包编号5	第5个执行的协议的通信类型中包含接收的情况下，将存储校验一致的接收数据包编号。通信类型为“仅发送”的情况下，将存储“0”。 执行第5个协议时发生了出错的情况下，将存储“0”。 执行的协议数不足5个时，将存储“0”。	0、1～16
(s3)+15	校验一致 接收数据包编号6	第6个执行的协议的通信类型中包含接收的情况下，将存储校验一致的接收数据包编号。通信类型为“仅发送”的情况下，将存储“0”。 执行第6个协议时发生了出错的情况下，将存储“0”。 执行的协议数不足6个时，将存储“0”。	0、1～16
(s3)+16	校验一致 接收数据包编号7	第7个执行的协议的通信类型中包含接收的情况下，将存储校验一致的接收数据包编号。通信类型为“仅发送”的情况下，将存储“0”。 执行第7个协议时发生了出错的情况下，将存储“0”。 执行的协议数不足7个时，将存储“0”。	0、1～16
(s3)+17	校验一致 接收数据包编号8	第8个执行的协议的通信类型中包含接收的情况下，将存储校验一致的接收数据包编号。通信类型为“仅发送”的情况下，将存储“0”。 执行第8个协议时发生了出错的情况下，将存储“0”。 执行的协议数不足8个时，将存储“0”。	0、1～16

十三、三菱 FX5U 实现 Modbus TCP 客户端编程：

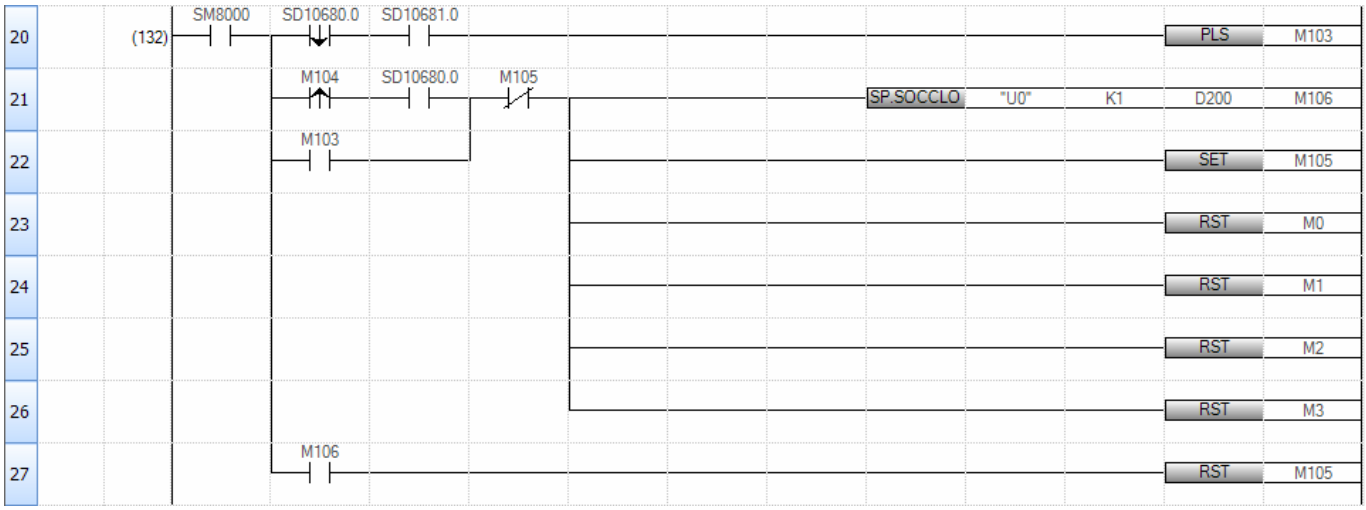
1、初始化部分程序，D0~D3 初始化清零、D4~D7 初始化分别赋值 0~3、每 1 分钟给 D4~D7 做加 1 操作、当 D4~D7 分别大于 6000 时清零，如下所示：



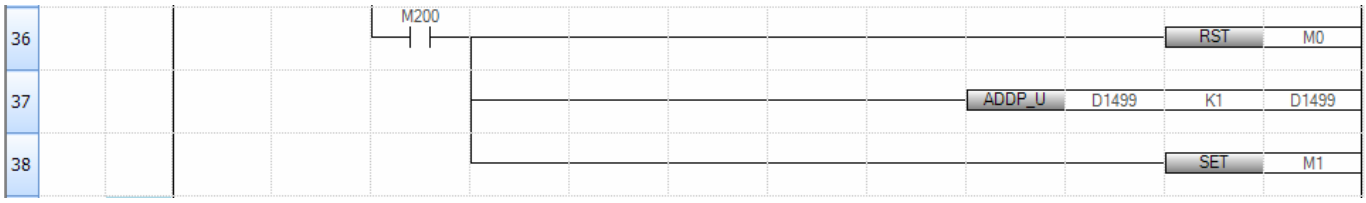
2、和 Modbus TCP 服务器建立连接部分程序，当 Modbus TCP 服务器准备就绪时，置位 M100 即可和 Modbus TCP 服务器建立连接，程序随即置位 M0，即可开始执行 SP.ECPRTCL 指令，如下所示：



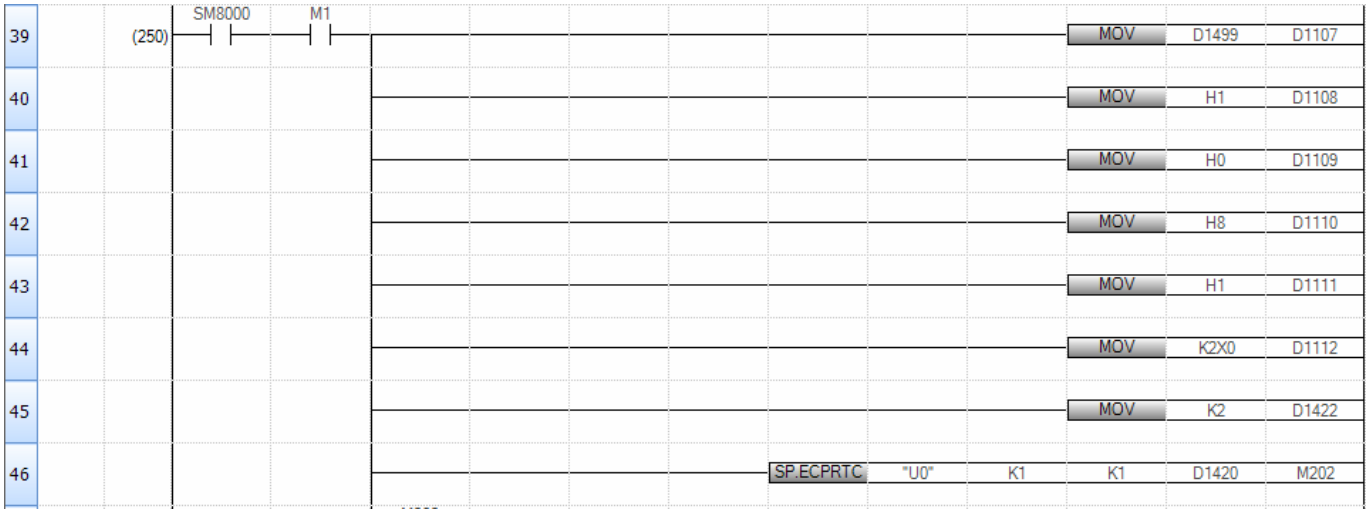
3、和 Modbus TCP 服务器断开连接部分程序，当 Modbus TCP 服务器断开连接或者置位 M104 时，即可执行断开和 Modbus TCP 服务器连接的操作，程序随即复位 M0~M3，终止执行 SP.ECPRTCL 指令，如下所示：

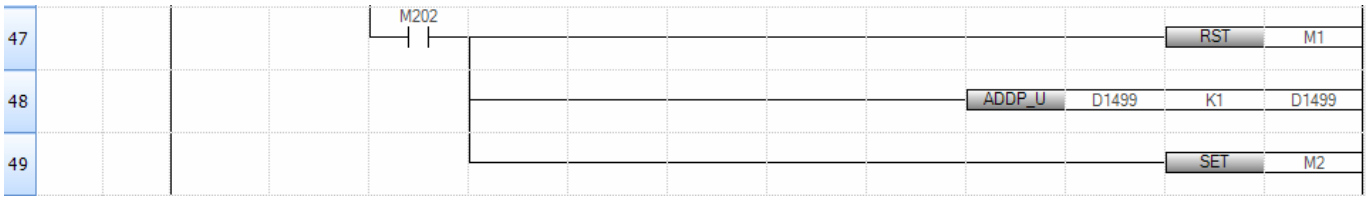


4、读取 Modbus TPC 服务器 8 路输入部分程序，功能码 02，对应协议号 1，M0 置位后开始执行该段程序，读取的 Modbus TCP 服务器 8 路输入映射到三菱 FX5U 的 8 路数字量输出 Y0~Y7 里，该段程序中的 SP.ECPRTCL 指令执行完成之后复位 M0、置位 M1，紧接着去执行下一个 SP.ECPRTCL 指令，如下所示：

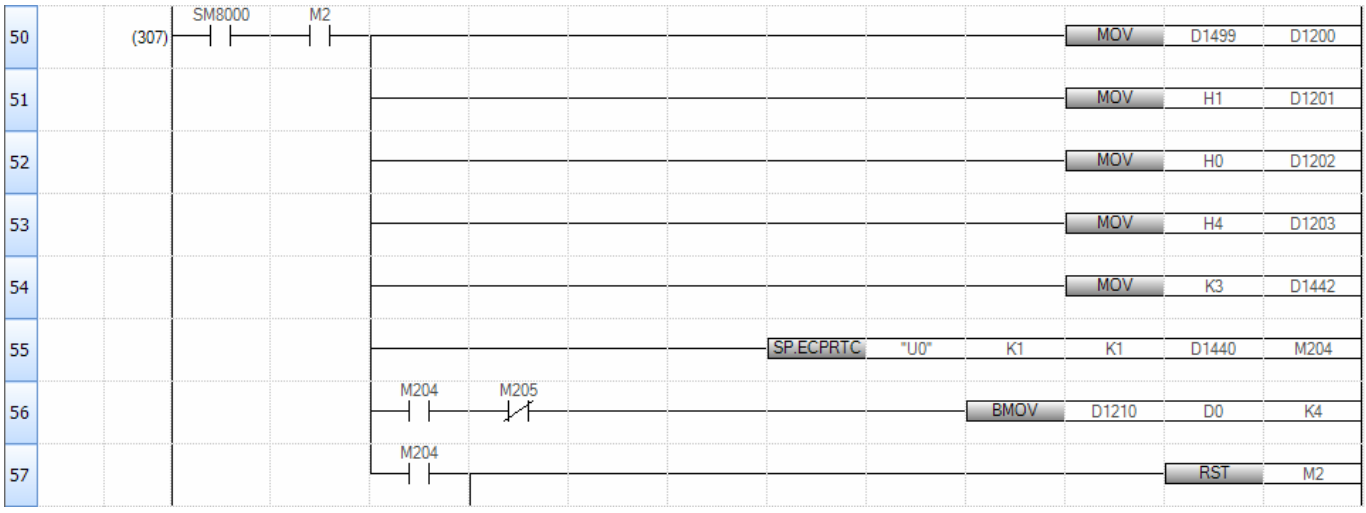


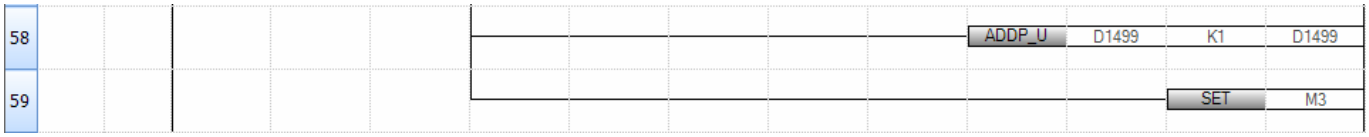
5、写入 Modbus TCP 服务器 8 路线圈部分程序，功能码 15，对应协议号 2，M1 置位后开始执行该段程序，使用三菱 FX5U 的 X0~X7 去控制 Modbus TCP 服务器 8 路线圈，该段程序中的 SP.ECPRTCL 指令执行完成之后复位 M1、置位 M2，紧接着去执行下一个 SP.ECPRTCL 指令，如下所示：



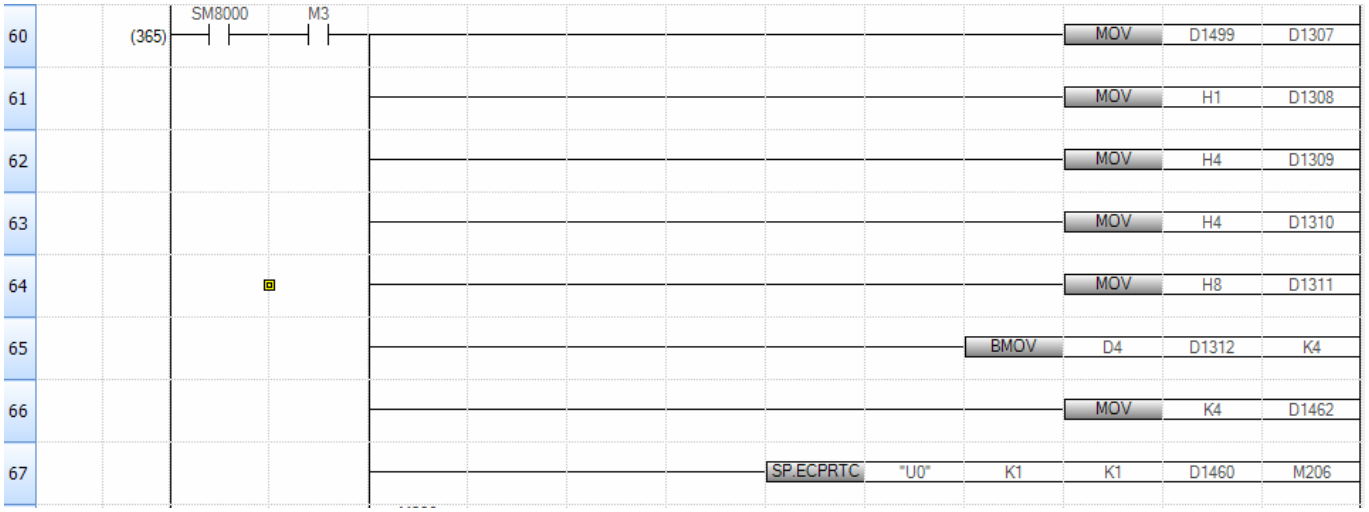


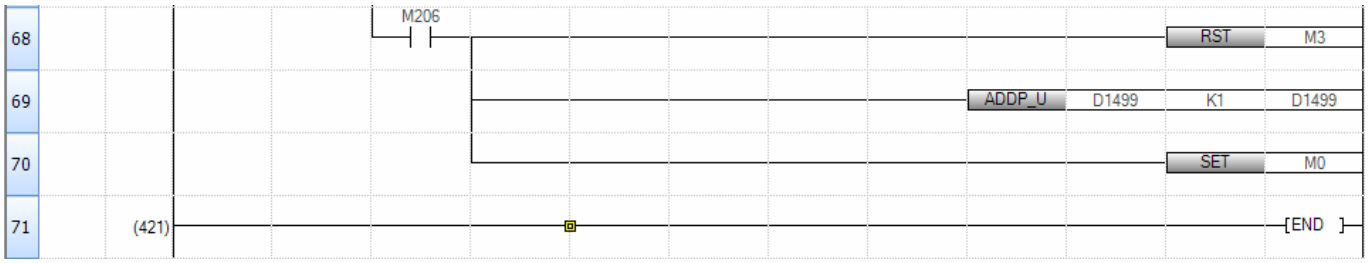
6、读取 Modbus TCP 服务器 4 路保持寄存器部分程序，功能码 03，对应协议号 3，M2 置位后开始执行该段程序，读取 Modbus TCP 服务器 4 路保持寄存器后转存到三菱 FX5U 的 D0~D3 寄存器中，该段程序中的 SP.ECPRTCL 指令执行完成之后复位 M2、置位 M3，紧接着去执行下一个 SP.ECPRTCL 指令，如下所示：





7、写入 Modbus TCP 服务器 4 路保持寄存器部分程序，功能码 16，对应协议号 4，M3 置位后开始执行该段程序，使用三菱 FX5U 的 D4~D7 去控制 Modbus TCP 服务器 4 路保持寄存器，该段程序中的 SP.ECPRTCL 指令执行完成之后复位 M3、置位 M0，紧接着又去执行第一个 SP.ECPRTCL 指令，如下所示：

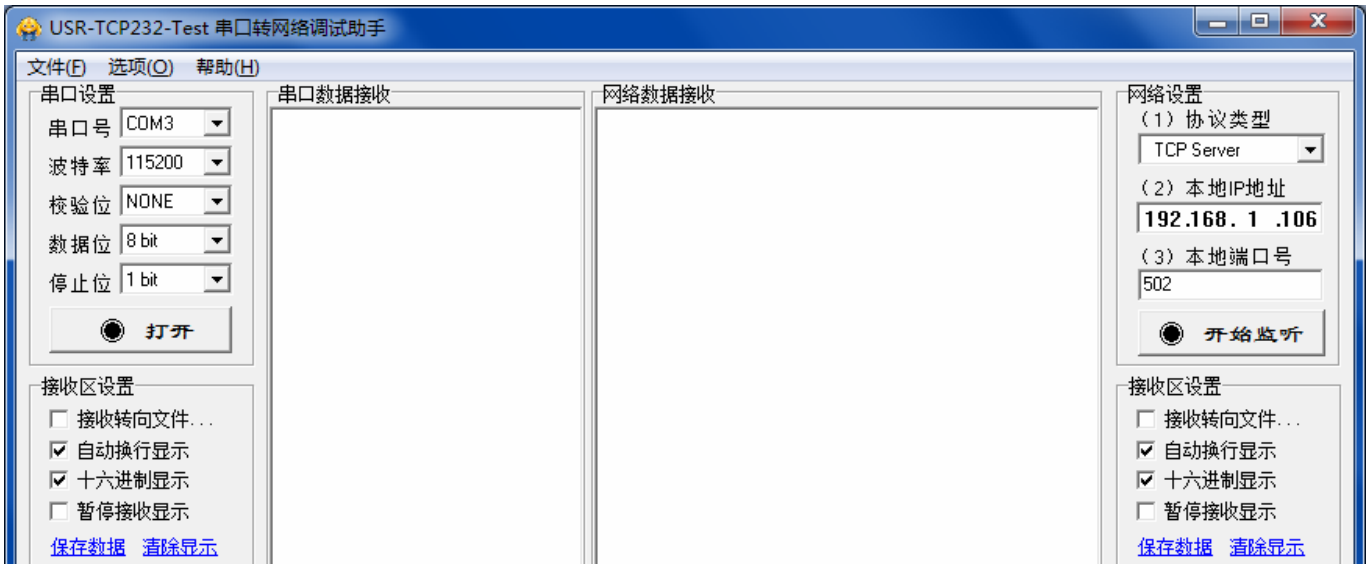




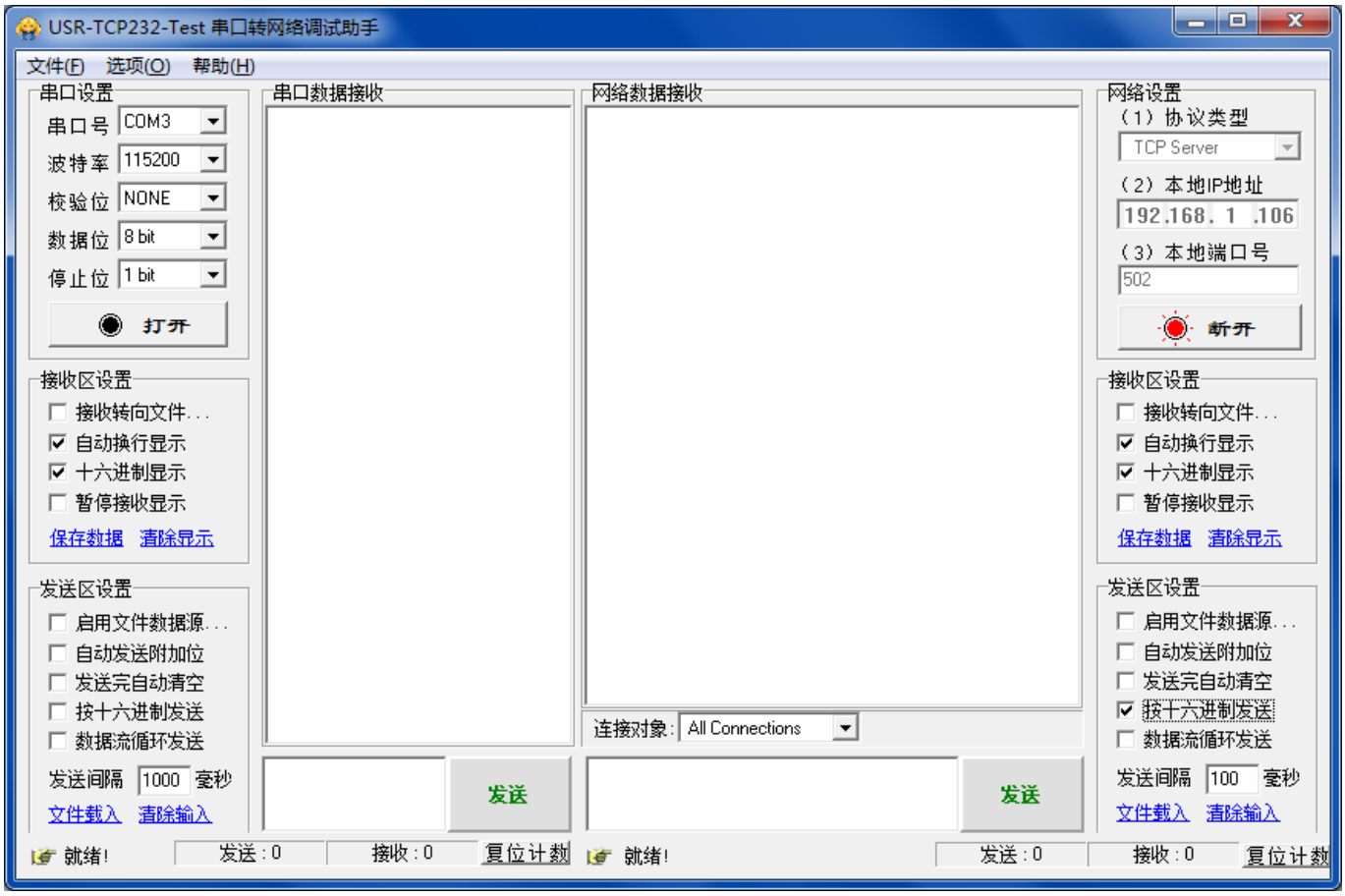
十四、三菱 FX5U 和以太网调试助手通信连接：

1、以太网调试助手做以太网服务器的设置

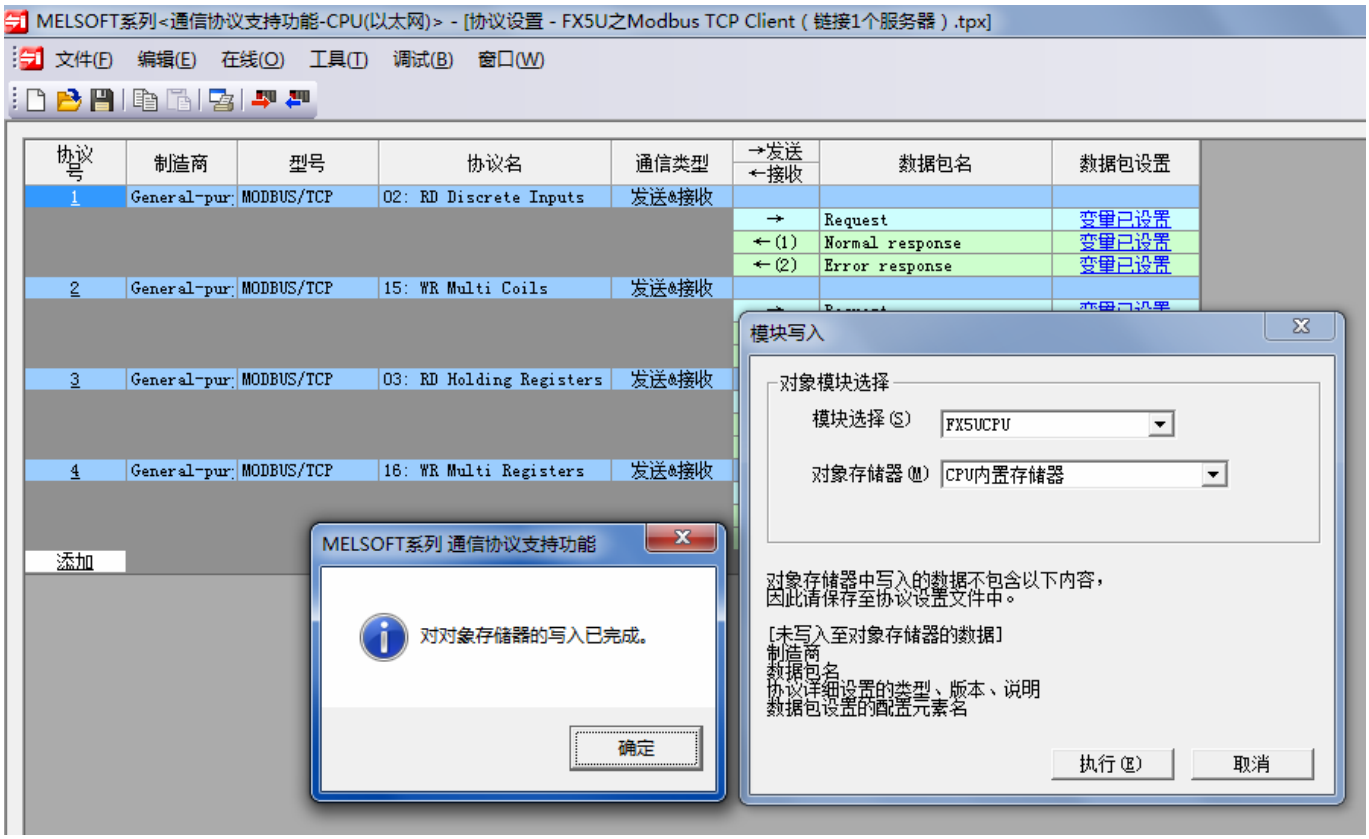
打开以太网调试助手，如下所示：



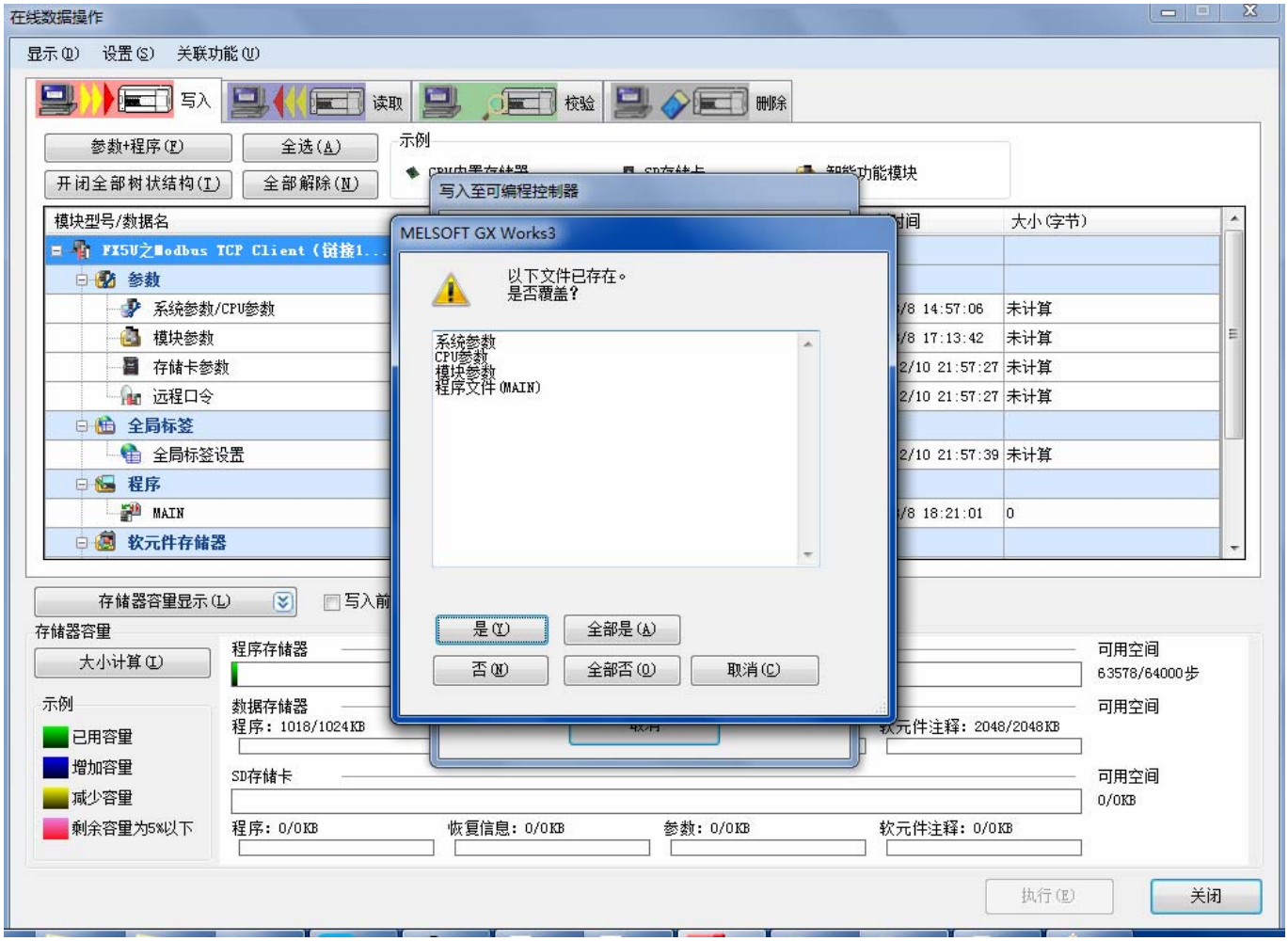
协议类型选择 TCP Server、本地 IP 地址按电脑实际 IP 地址设置为 192.168.1.106、本地端口号设置为 502，点击开始监听，如下所示：



2、下载三菱 FX5U 以太网口通信协议支持功能数据包

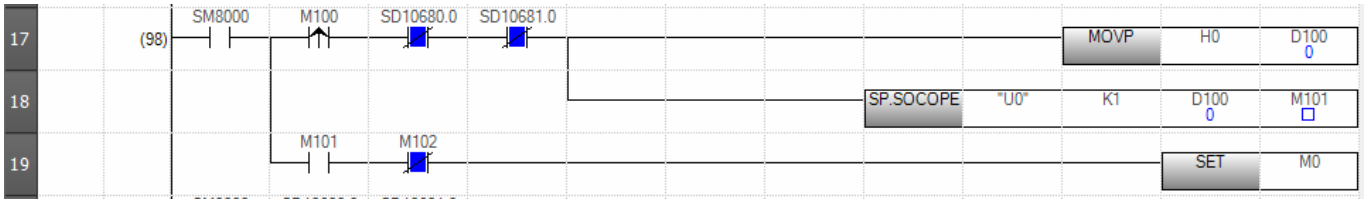


3、下载三菱 FX5U 程序

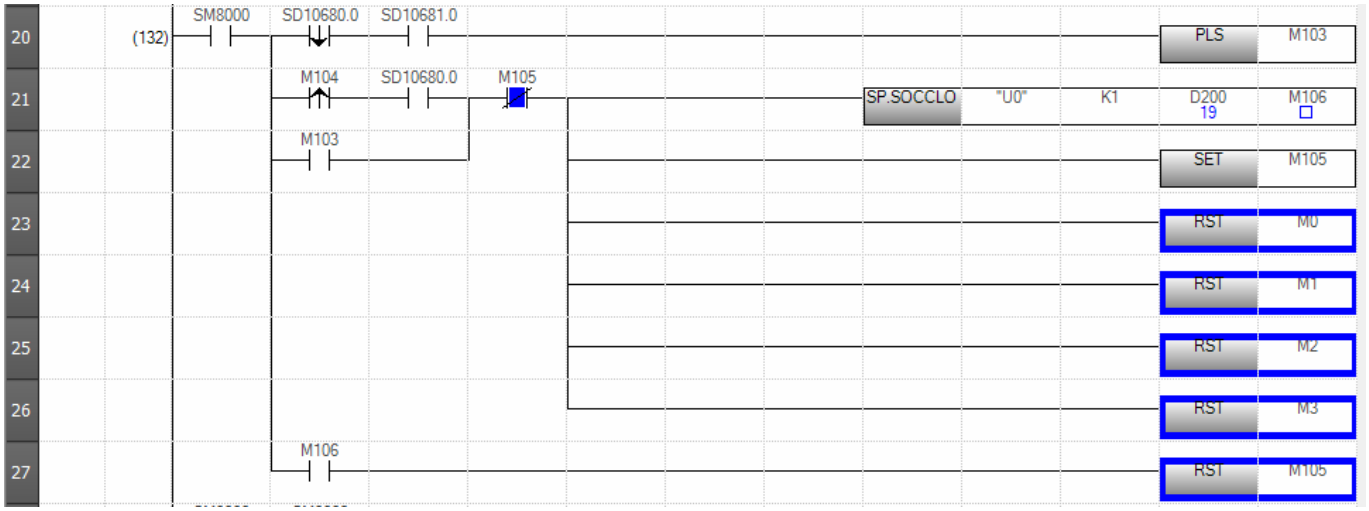


4、监视三菱 FX5U 程序，如下所示

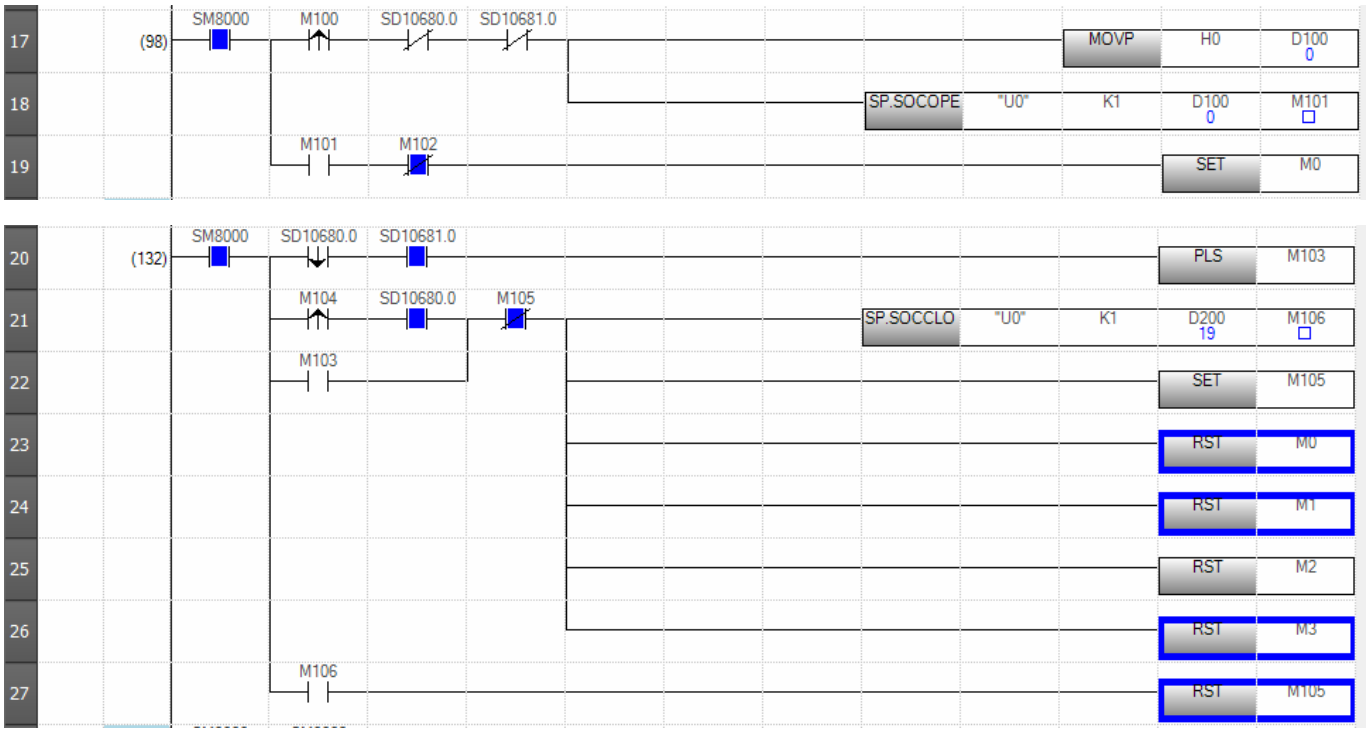
建立连接程序段：



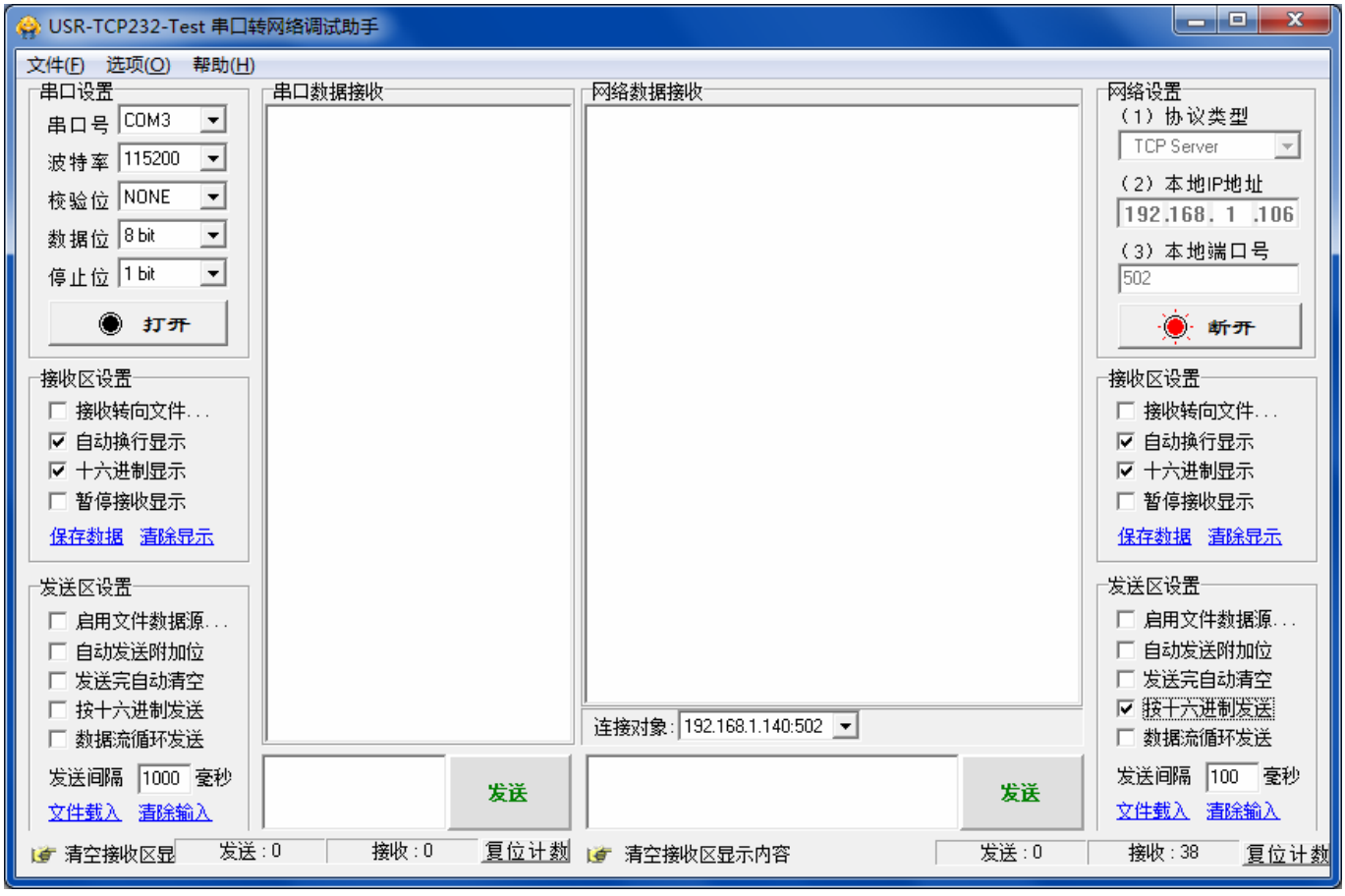
断开连接程序段：



以上可以看出连接 1 开放结束信号 SD10680.0、连接 1 开发请求信号 SD10681.0 均为 0，符合建立和 Modbus TCP 服务器连接条件，此时可置位 M100 建立和以太网调试助手的连接，如下所示：

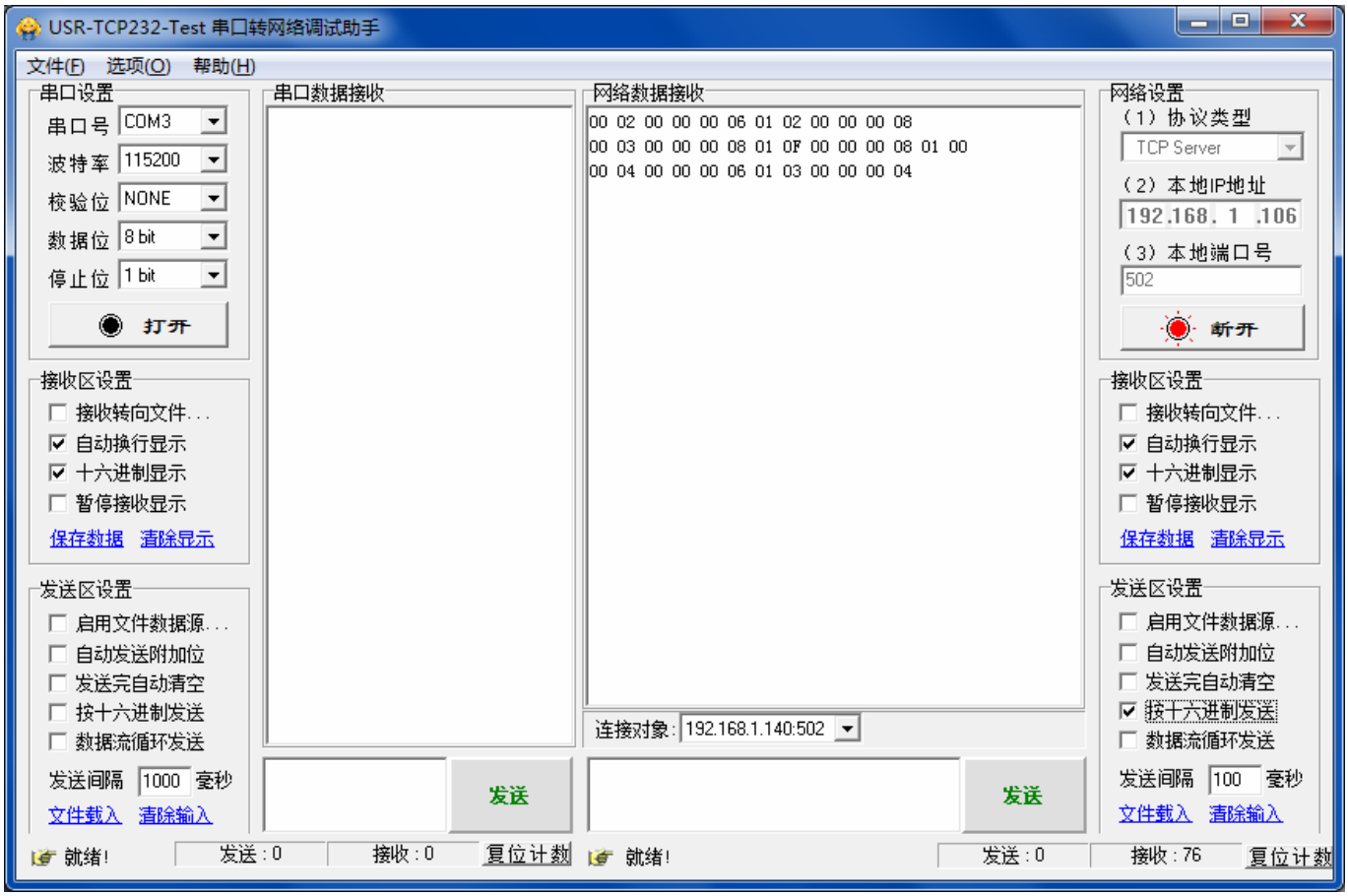


5、以太网调试助手连接建立，如下所示：



十五、三菱 FX5U 之 Modbus TCP 客户端程序的监视：

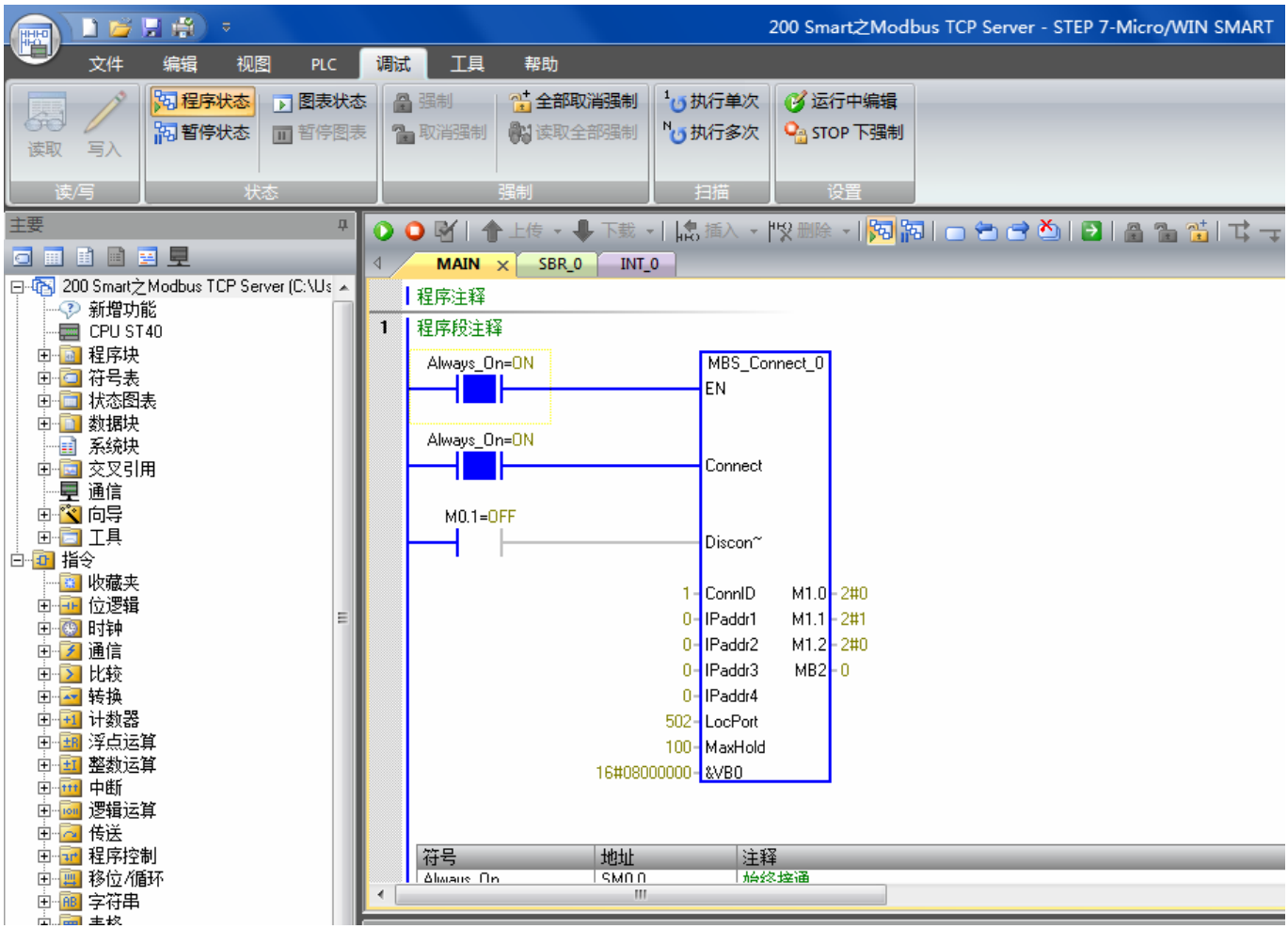
三菱 FX5U 和以太网调试助手建立连接成功后，以太网调试助手即可观察到来自三菱 FX5U 发出的 Modbus TCP 命令，如下所示：



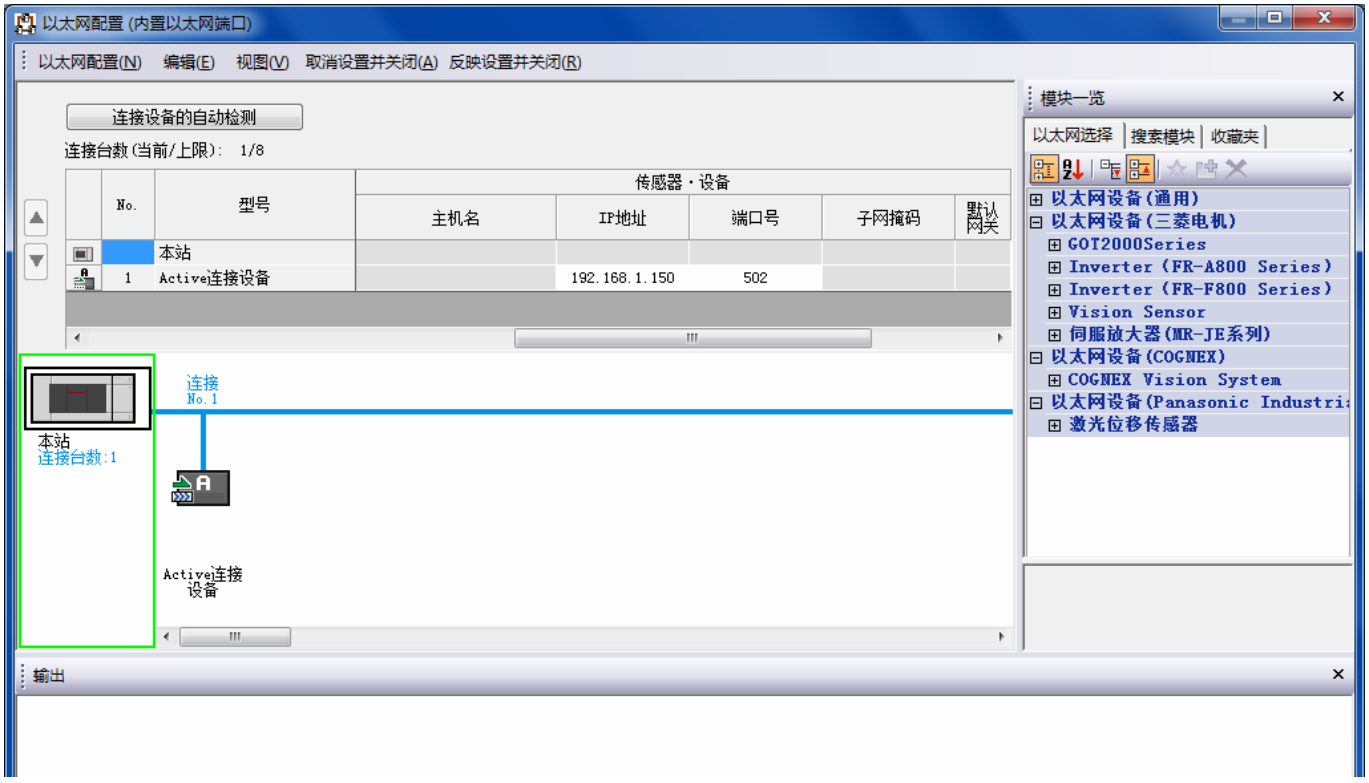
十六、三菱 FX5U 和西门子 200 Smart 通信测试步骤：

1、使用网线将西门子 200 Smart 连接至无线路由器 LAN 口、使用网线将三菱 FX5U 连接至无线路由器 LAN 口，完成硬件连接，如下所示：

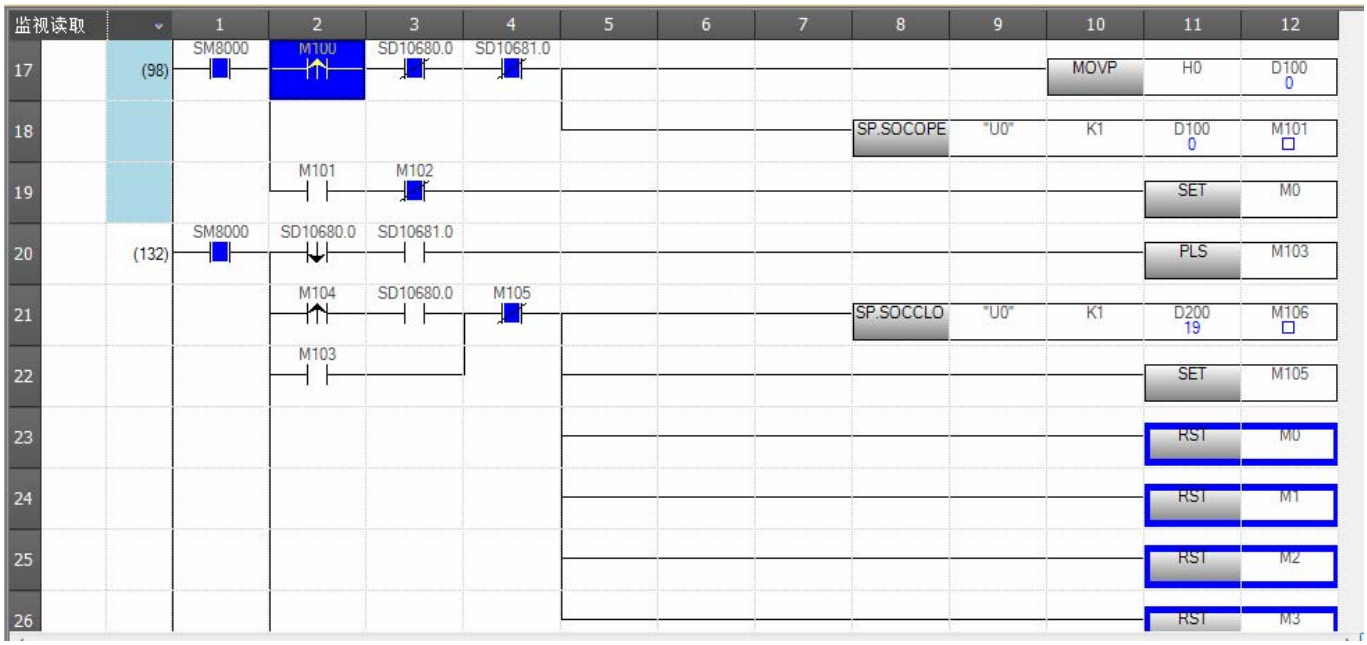
2、下载西门子 200 Smart 之 Modbus TCP 服务器程序，并打开监控，如下所示：



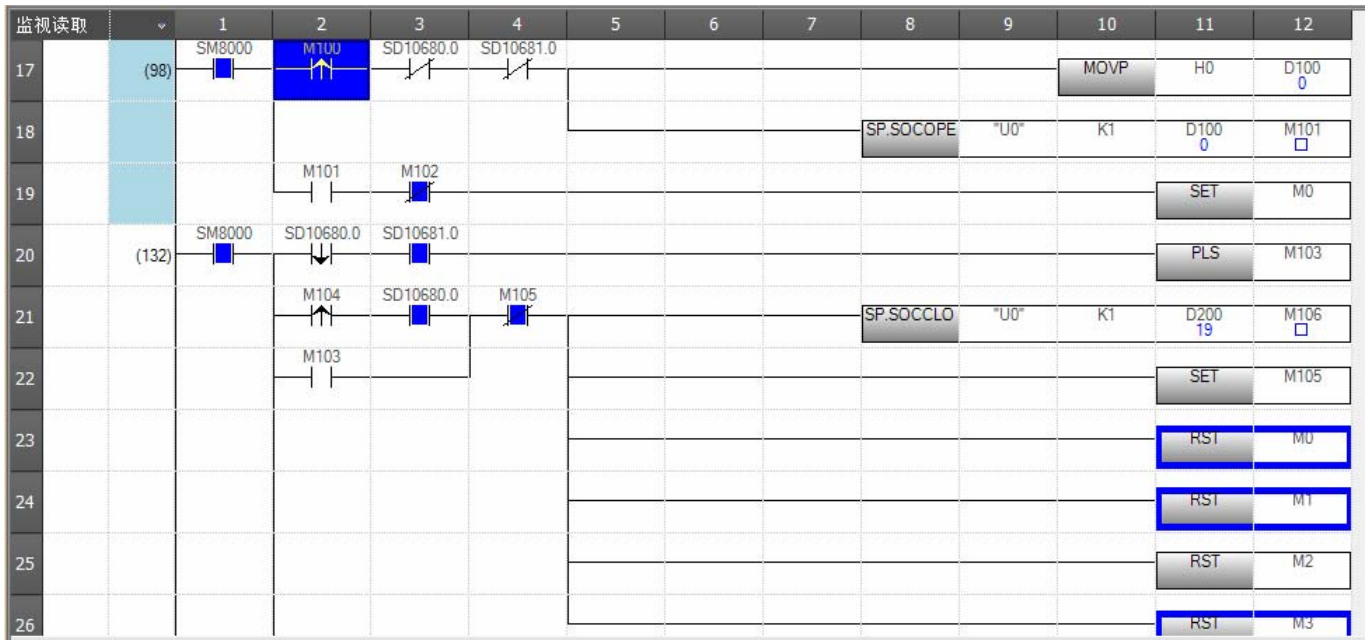
3、将三菱对象设备连接配置设置中连接设备的 IP 地址修改为西门子 200 Smart 的 IP 地址，如下所示：

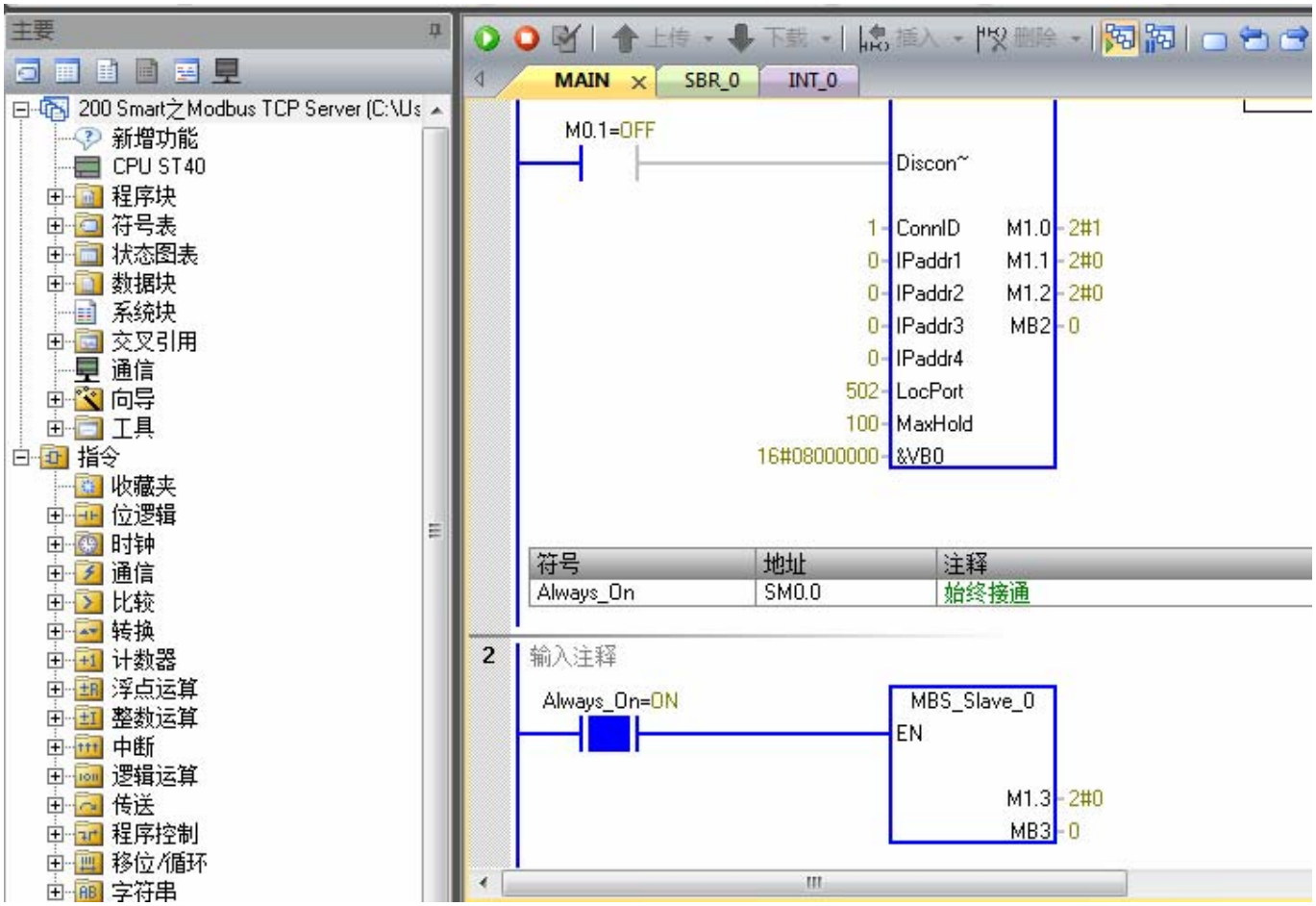


4、重新下载三菱 FX5U 之 Modbus TCP 客户端程序，并打开监控，如下所示：



5、在三菱编程软件中置位 M100，即可建立三菱 FX5U 和西门子 200 Smart 的 TCP 连接，连接建立成功后，三菱 FX5U 随即会发送有关 02、15、03、16 功能码命令，如下所示：





6、Modbus 之 02 功能码的测试，系统实现功能是西门子 200 Smart 的 I0.0~I0.7 控制三菱 FX5U 的 Y0~Y7，在西门子编程软件中强制 I0.0 为 1，同时观察三菱编程软件中 Y0 的状态，如下所示：

7、Modbus 之 15 功能码的测试，系统实现功能是三菱 FX5U 的 X0~X7 控制西门子 200 Smart 的 Q0.0~Q0.7，在三菱编程软件中强制 X1 为 1，同时观察西门子编程软件中 Q0.1 的状态，如下所示：

8、Modbus 之 03 功能码的测试，系统实现功能是三菱 FX5U 读取西门子 200 Smart 的 VW0~VW6 保存到自己的 D0~D3 里，监视西门子编程软件中 VW0~VW6 的值，并监视三菱编程软件中 D0~D3 的值，如下所示：

9、Modbus 之 16 功能码的测试，系统实现功能是三菱 FX5U 的 D4~D7 去控制西门子 200 Smart 的 VW8~VW14，监视三菱编程软件中 D4~D7 的值，并监视西门子编程软件 VW8~VW14 的值，如下所示：

软元件名	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	当前值	字符串
D0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	20	..
D1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	21	..
D2	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	22	..
D3	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	23	..
D4	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	20	..
D5	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	21	..
D6	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	22	..
D7	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	23	..
D8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	..

	地址	格式	当前值	新值
2	CPU_输出1	位	2#0	
3	VW0	有符号	+20	
4	VW2	有符号	+21	
5	VW4	有符号	+22	
6	VW6	有符号	+23	
7	VW8	有符号	+20	
8	VW10	有符号	+21	
9	VW12	有符号	+22	
10	VW14	有符号	+23	

十七、总结：

至此，三菱 FX5U（Modbus TCP 客户端）和西门子 200 Smart（Modbus TCP 服务器）完美实现了 Modbus TCP 以太网通信。

十八、作者简介：

关普，中华工控网串口通信板块版主，就职于西安棋影工作室，专注各种组态软件、触摸屏、PLC、单片机、变频器、伺服控制器、智能仪表等 Modbus TCP 以太网通信、Modbus RTU 串口通信、Modbus ASCII 串口通信和其他协议通信等！联系方式 QQ149034219、微信 guanyumou