倍福 BK9100/BC9000 通过 ModbusTCP 与 GE RX3i 通讯

一、试验目的：

测试 BK9100/BC9000 和 GE RX3i 通过 ModbusTCP 的通讯功能。

二、试验时间和地点：

2007 年 4 月 17 日，4 月 25 日，北京西通电子有限公司

三、试验人员：

王宁强（技术支持工程师，德国倍福自动化有限公司北京代表处）

陈明辉（技术工程师，北京西通电子有限公司（GE 授权分销商））

四、试验设备：

倍福产品清单：

BK9100，1 块

BC9000，1 块

KL1002，2 块

KL2012，2 块

KL9010，2 块

KL3132，1 块

KL4132，1 块

GE 产品清单：

IC695CPU310-EL(RX3i)，1 块

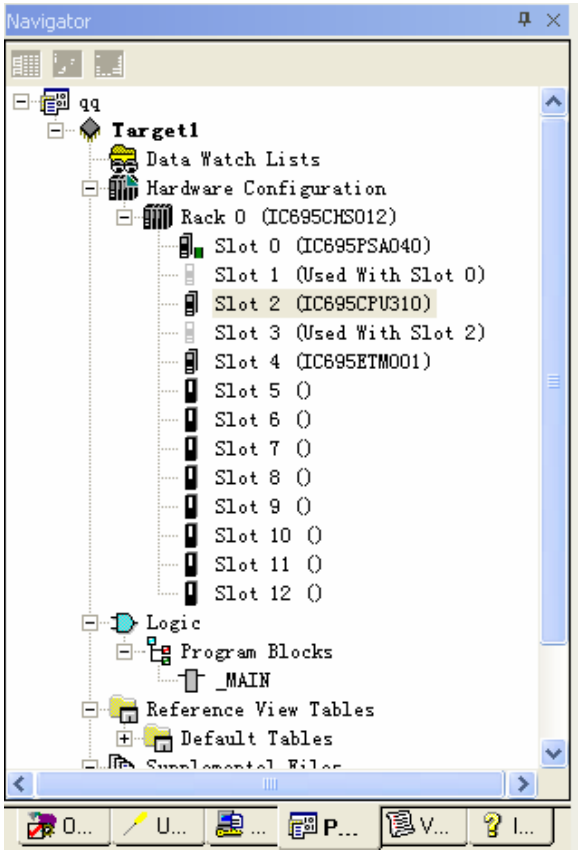
IC695ETM001-CC(以太网模块，10/100MBITS)，1 块

IC695PSA040F(电源)，1 块

五、试验步骤：

（一）设置

1．打开当前项目的 Hardware Configuration



2．双击 Rack 0 下的 CPU（Slot2），并将右侧窗口中 Settings 标签下的最后一项设置如下图中红色框中所示：

Settings | Scan | Memory | Faults | Port 1 | Port 2 | Scan Sets | Modbus TCP Address Map | Power

Parameters	Values
Passwords	Enabled
Stop-Mode I/O Scanning	Disabled
Watchdog Timer (ms)	200
Logic/Configuration Power-up Source	Always RAM
Data Power-up Source	Always RAM
Run/Stop Switch	Enabled
Memory Protection Switch	Disabled
Power-up Mode	Last
Modbus Address Space Mapping Type	Standard Modbus Addressing

3．双击 Rack 0 下的以太网模块（Slot4），并将右侧窗口中 Settings 标签下的 IP 地址设置为与 BC9000/BK9100 在同一网段内，如图所示（同时 BK9100 的 IP 地址为 192.168.0.7）：

Settings | RS-232 Port (Station Manager) | Power Consumption

Parameters	Values
Configuration Mode	TCP/IP
Adapter Name	0.4
Use BOOTP for IP Address	False
IP Address	192.168.0.10
Subnet Mask	255.255.255.0
Gateway IP Address	0.0.0.0
Name Server IP Address	0.0.0.0
Max FTP Server Connections	2
Network Time Sync	None
Status Address	%I00001
Length	80
I/O Scan Set	1

（二）GE RX3i 作为主站（Client）读写 BK9100/BC9000 (Server）

1．BC9000/BK9000 的 ModbusTcp 过程映像

- 输入过程映像从地址从0x0000开始（对应于GE 功能块中的地址0x0001）。所有面向字节的总线端子优先输入到过程映像区中。面向位的总线端子紧随其后。
- 输出过程映像从地址从0x0800开始(对应于GE 功能块中的地址0x0801)。所有面向字节的总线端子优先输入到过程映像区中。面向位的总线端子紧随其后。
- Memory Flag过程映像从地址 0x4000开始。(对应于GE 功能块中的地址 0x4001)
- 所有的数字量信号可以直接使用功能 1，2，5 和 15 寻址。(直接寻址意味着使用这几个功能时，数字量端子的映像地址从 0x0000 开始(对应于 GE 功能块中的地址 0x4001)，不考虑面向字节的总线端子。

2．功能代码

在GE PLC中，通过COMM_REQ功能块来发送请求：（COMM_REQ功能块的使用参见GE文档GFK2224D）

功能代码1：

- 读取线圈状态，地址从0x0001开始(COMM_REQ的Command Block中的 word7＝3003，word9＝1，word12＝线圈地址，word13＝线圈数量(bits))

功能代码2：

- 读取DI状态，地址从0x0001开始(COMM_REQ的Command Block中的 word7＝3003，word9＝2，word12＝DI地址，word13＝DI数量(bits))

功能代码3：

- 读取寄存器（Memory Flag），地址从0x4001开始(COMM_REQ的Command Block中的 word7＝3003，word9＝3，word12＝M寄存器地址，word13＝M寄存器数量(words))
- 读取 AO，地址从 0x0801 开始，数字输出的地址排在所有面向字节的总线端子之后(COMM_REQ 的 Command Block 中的 word7＝3003，word9＝3，word12＝输出寄存器地址，word13＝输出寄存器数量(words))

功能代码4：

- 读取 AI，地址从 0x0001 开始，数字输入的地址排在所有面向字节的总线端子之后(COMM_REQ 的 Command Block 中的 word7＝3003，word9＝4，word12＝输入寄存器地址，word13＝输入寄存器数量(words))

功能代码5：

- 写单个线圈，地址从0x0001开始(COMM_REQ的Command Block中的 word7＝3004，word9＝5，word12＝线圈地址，word13＝1线圈数量(bits))

功能代码15：

- 写多个线圈，地址从0x0000开始(COMM_REQ的Command Block中的 word7＝3004，word9＝15，word12＝线圈地址，word13＝线圈数量(bits))

功能代码6：

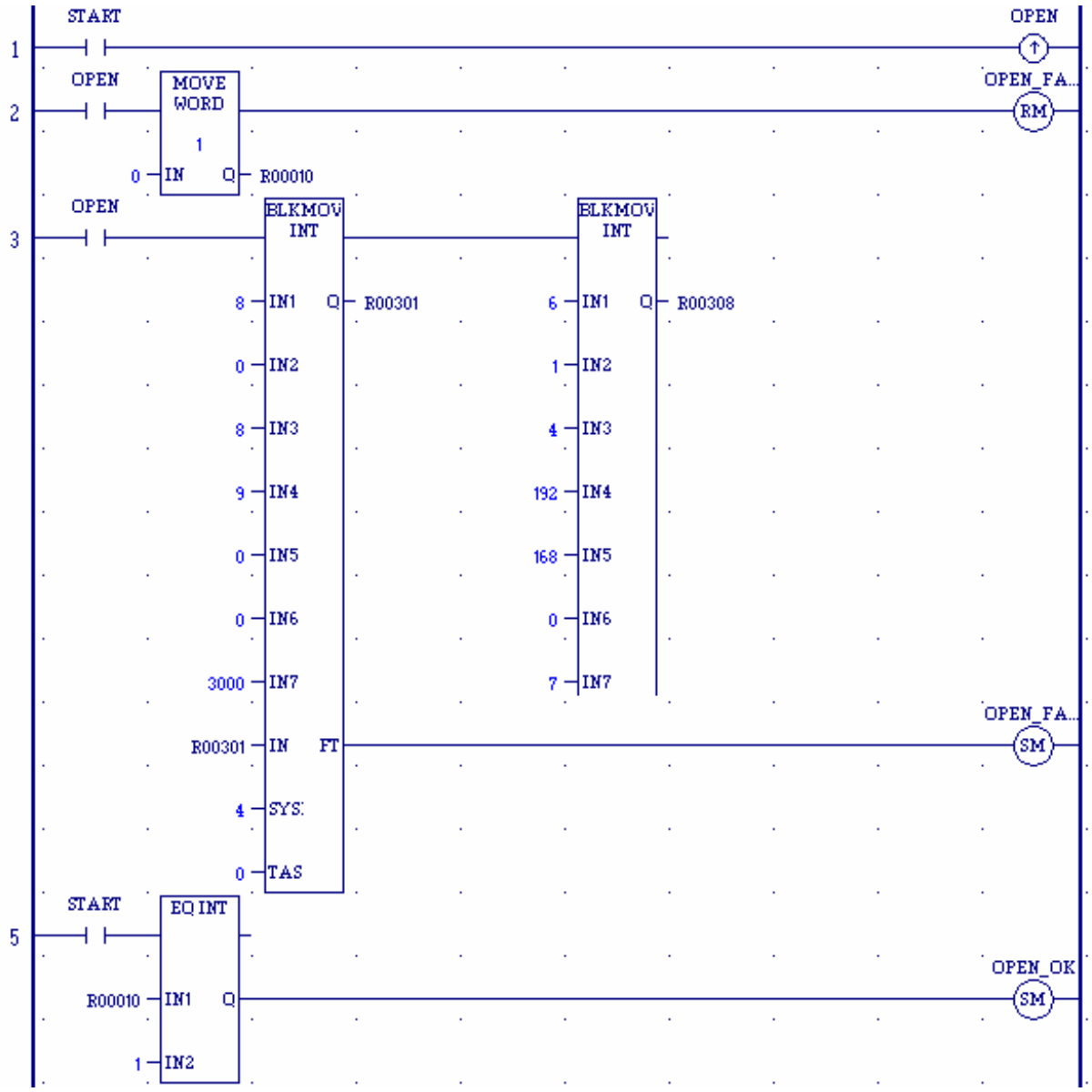
- 写单个寄存器，地址从0x4001开始(COMM_REQ的Command Block中的 word7＝3004，word9＝6，word12＝M寄存器地址，word13＝1M寄存器数量(bits))
- 写单个AO寄存器，地址从0x0801开始，数字输出的地址排在所有面向字节的总线端子之后(COMM_REQ的Command Block中的 word7＝3004，word9＝6，word12＝输出寄存器地址，word13＝1输出寄存器数量(words))

功能代码16：

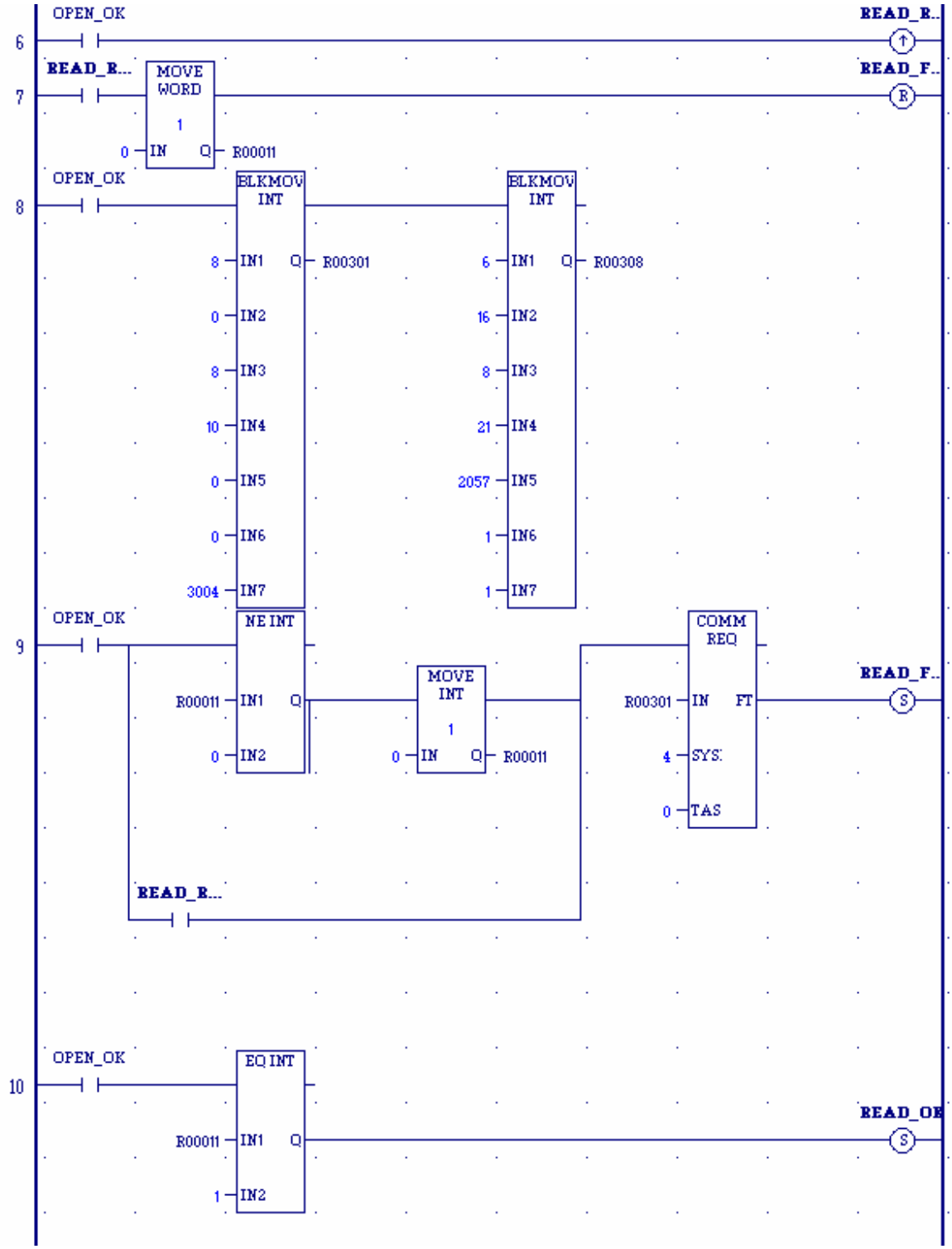
- 写多个寄存器，地址从0x4001开始(COMM_REQ的Command Block中的 word7＝3004，word9＝16，word12＝M寄存器地址，word13＝M寄存器数量(bits))
- 写多个AO寄存器，地址从0x0801开始，数字输出的地址排在所有面向字节的总线端子之后(COMM_REQ的Command Block中的 word7＝3004，word9＝16，word12＝输出寄存器地址，word13＝输出寄存器数量(words))

3．程序示例：

- Open a Modbus/TCP Client Connection (3000)
- Send Request to Modbus/TCP Server(3003,3004)



Open a Modbus/TCP Client Connection



Send Request to Modbus/TCP Server

（三）PC/BC9000 作为主站（Client）读写 GE RX3i (Server）

1．GE RX3i 的 ModbusTcp 映像地址

- Q 区（数字输出）：地址从 0x0001 开始（对应于 BC 功能块中的地址 0x0000）
- I 区（数字输入）：地址从 0x0001 开始（对应于 BC 功能块中的地址 0x0000）
- AI 区 (模拟输入)：地址从 0x0001 开始（对应于 BC 功能块中的地址 0x0000）
- R 区（寄存器） ：地址从 0x0001 开始（对应于 BC 功能块中的地址 0x0000）
- 模拟输出不能读写，但是可以通过 R 区间接读写。（GE PLC 中的 Modbus 寄存器表完全映像到 R 区）

2．功能代码

在 BC 中，通过 FB_ModbusTcpRequest 功能块来发送请求：（该功能块的使用参见 TwinCAT PLC 帮助文档）

功能代码1：

- 读取线圈状态，地址从0x0000开始

功能代码2：

- 读取DI状态，地址从0x0000开始

功能代码3：

- 读取寄存器（Memory Flag），地址从0x0000开始

功能代码4：

- 读取 AI，地址从 0x0000 开始

功能代码 5：

- 写单个线圈，地址从0x0000开始

功能代码15：

- 写多个线圈，地址从0x0000开始

功能代码6：

- 写单个寄存器，地址从0x0000开始

功能代码16：

- 写多个寄存器，地址从 0x4000 开始

3．BC9000 程序示例：

```
PROGRAM MAIN
VAR
    ModbusTCP_Open :FB_ModbusTcpOpen;
    ModbusTCP_Request   :FB_ModbusTcpRequest;
    bStartOpen          :BOOL:=FALSE;
    BK_IPAddress        :STRING(15):='192.168.0.1';
    bBusyOpen           :BOOL;
    bErrorOpen          :BOOL;
    iErrorID            :WORD;
    Connection          :WORD:=0;
    bStartRequest       :BOOL       :=FALSE;
    bBusyRequest        :BOOL;
    bErrorRequest       :BOOL;
    iErrorIDRequest     :WORD;
    cbResponse          :WORD;
    SendBuffer          :ARRAY[0..5] OF BYTE    :=1,3,0,0,0,2;
```

```
        (*1: Unit    Identifier *)
        (*3: 功能号，此时为读寄存器 *)
        (*0，0: 起始地址高，低字节 *)
        (*0，2: 数据长度高，低字节 *)
    ReceiveBuffer           :ARRAY[0..255] OF BYTE;
END_VAR

ModbusTCP_Open(     bStart          :=bStartOpen,
                    sRemoteIPAddr   :=BK_IPAddress,
                    tTimeOut        :=t#2S);
bBusyOpen   :=ModbusTCP_Open.bBusy;
bErrorOpen  :=ModbusTCP_Open.bError;
iErrorID    :=ModbusTCP_Open.iErrorId;
Connection  :=ModbusTCP_Open.iMyPortNo;
bStartOpen  :=(Connection=0)AND(NOT bStartOpen)AND(NOT bBusyOpen);

ModbusTCP_Request(      bStart          :=bStartRequest,
                        sRemoteIPAddr   :=BK_IPAddress,
                        iMyPortNo       :=Connection,
                        pReqBuff    :=ADR(SendBuffer),
                        cbReqLen        :=SIZEOF(SendBuffer),
                        pResBuff        :=ADR(ReceiveBuffer),
                        cbResLen    :=SIZEOF(ReceiveBuffer));
bBusyRequest :=ModbusTCP_Request.bBusy ;
bErrorRequest :=ModbusTCP_Request.bError;
iErrorIDRequest     :=ModbusTCP_Request.iErrorId ;
cbResponse          :=ModbusTCP_Request.cbResponse ;
bStartRequest  :=(Connection<>0)AND( NOT bStartRequest) AND (NOT bBusyRequest);
```

六、结论：

通过试验上述的 ModbusTcp 功能均能够正常实现。