TURCK

Industrial Automation

BLXX-PG-EN for GE 90-30 Series

SARTUP GUIDE

G1024
Published 8/28/2013

Content

1 Introduction

The purpose of this document is to guide you through the configuration of Programmable Modbus TCP/IP BL20 and BL67 gateways and RFID modules, ***BLident*** system and integration with GE Series 90™-30 PLC. The RFID interface logic is handled by the Programmable gateway and control, status, and data are mapped to the higher level controller, GE Series 90™-30. Tools used to configure and program this system are CoDeSys 3S, IO Assistant, and GE Machine Edition Logic Developer-PLC.

2 Reference Documentation

Following publications provide related information and technical description of the system components used by the system:

- BLident Modular RFID System
- BL67 User Manual for Modbus TCP/IP
- BL67 IO User Manual
- BL20 User Manual for Modbus TCP/IP
- BL20 IO User Manual
- IOAssistant – configuration software
- TCP/IP Ethernet Commications for Series 90™ PCL User's Manual (GFK-1541B)

3 System Overview

The system presented in this example consists of BL67 Programmable Modbus TCP/IP gateway with 2RFID-A module and Series 90™-30 PLC. The BL67 gateway act as a Modbus TCP/IP slave and implement RFID interface internally via Proxy Indent Block, ***BLident*** function block. Series 90™-30 PLC implement Modbus TCP/IP master functionality by periodically polling BL67 slave device. See *Diagram 3.1* for graphical representation of system control flow.

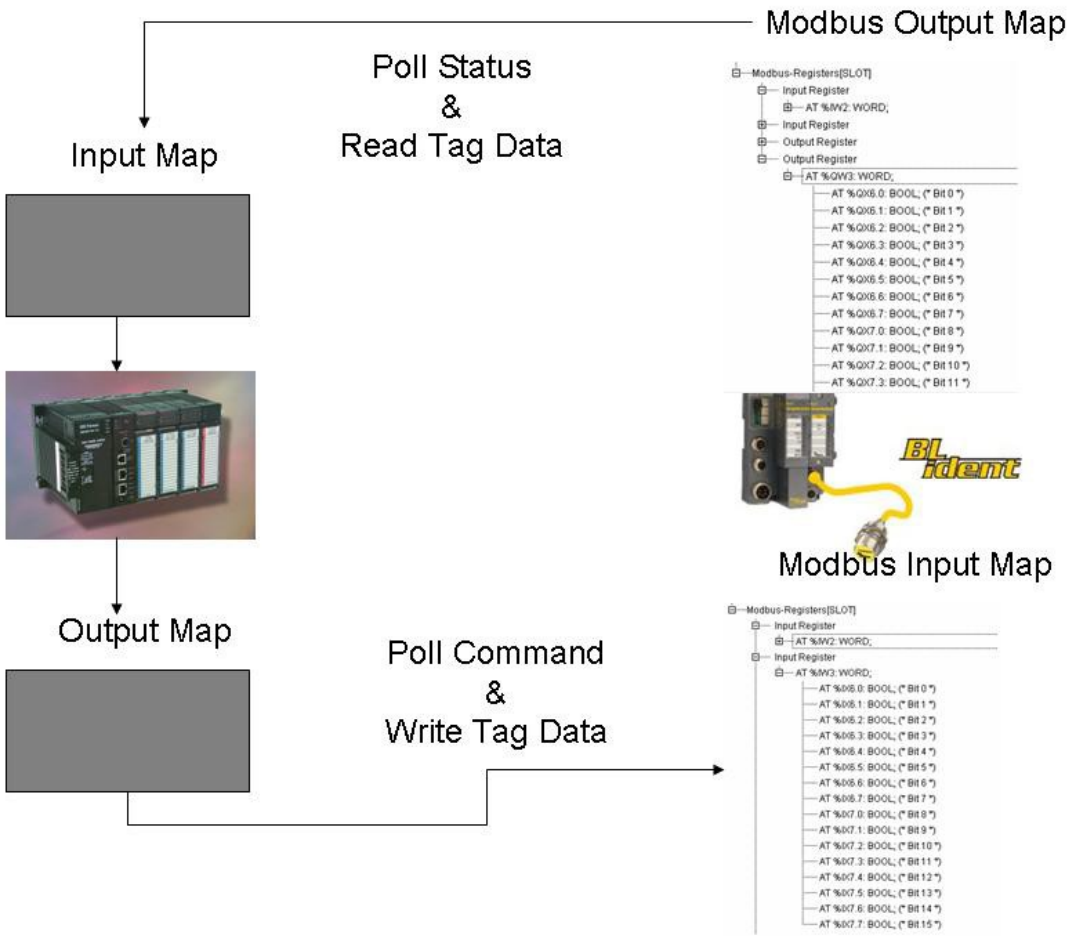


Diagram 3.1: Data and control flow of BLident system and GE Series 90™-30 PLC

4 BL67 Configuration

4.1 Hardware Requirements

- BL67 Programmable Modbus TCP/IP Gateway (BL67-PG-EN) firmware rev. 1.0.0.1 or later
- RFID Module (2RFID-A)
- BL67 base module (BL67-B-4M12)
- RFID transceiver head TB-M18
- RFID tag TW-R16-B128
- Ethernet 4 pin eurofast cable
- 24 VDC power supply

4.1.1 Hardware Setup

1. Assemble BL67 gateway with base module and RFID module.
2. Set the IP address of the gateway. The rotary switches should be set to a value between 000 and 254. In this case the first three bytes of the IP address are always 192.168.1. The last byte of the IP address is set using the rotary switches. For this example, set IP address to 192.168.1.1
3. Power up the programmable gateway.

4. Push the SET button for 10 seconds to store the gateway configuration.

4.2 Software Requirements

- IO Assistant Software
- CoDeSys v 2.3.5.8 (www.turck-usa.com/Support/Downloads ~ Software/)

4.2.1 IO Assistant

Using IO Assistance, confirm that BL67 system is configured as displayed on *Diagram 4.2*.

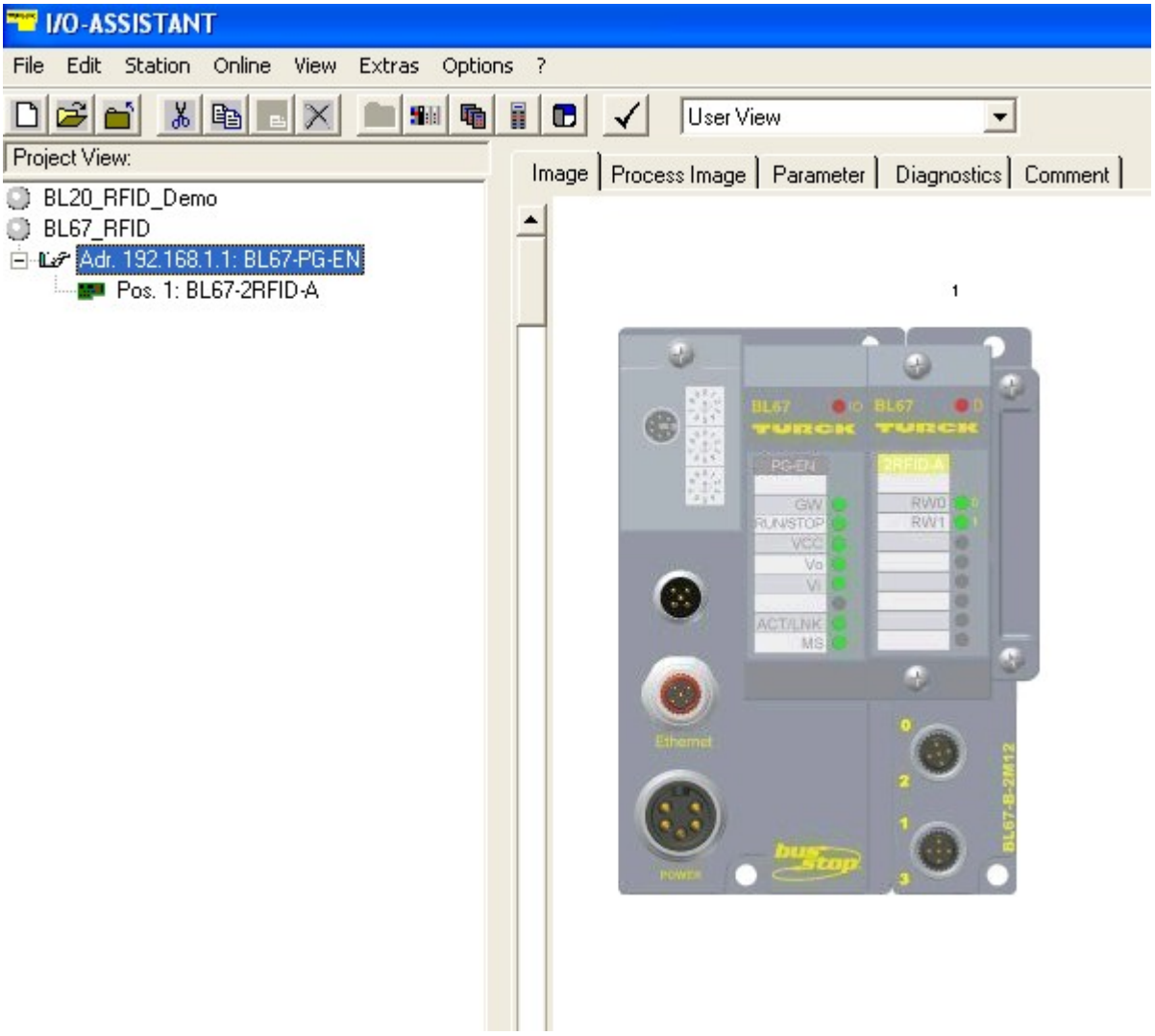
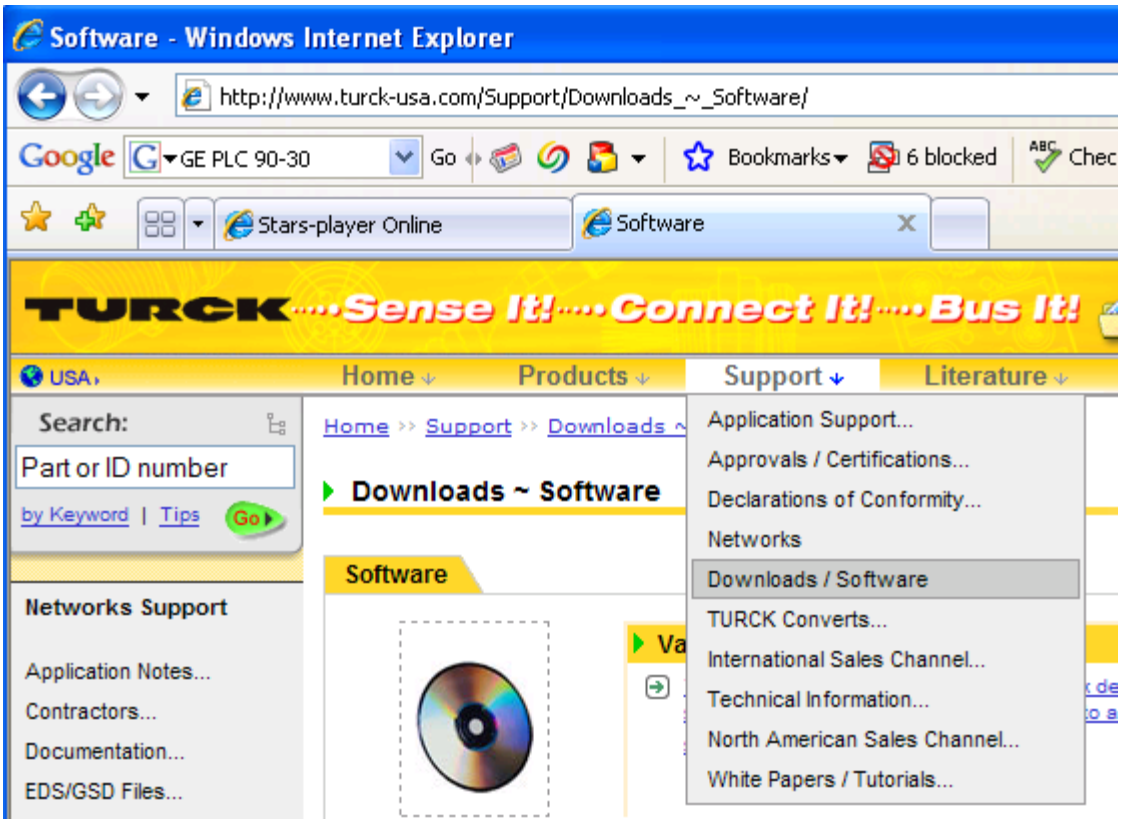


Diagram 4.2: IO Assistant BLident system configuration

4.2.2 BL67 Program

To implement ***BLident*** system on Programmable gateway, you need to write program including BLident function block to handle Proxy Identification with the RFID read/write heads. Following is the list of required libraries and supplementary files to create BL67 program:

- BL67_PG_EN_Target.zip – Target file for CoDeSys (www.turck-usa.com/Support/Downloads ~ Software/)

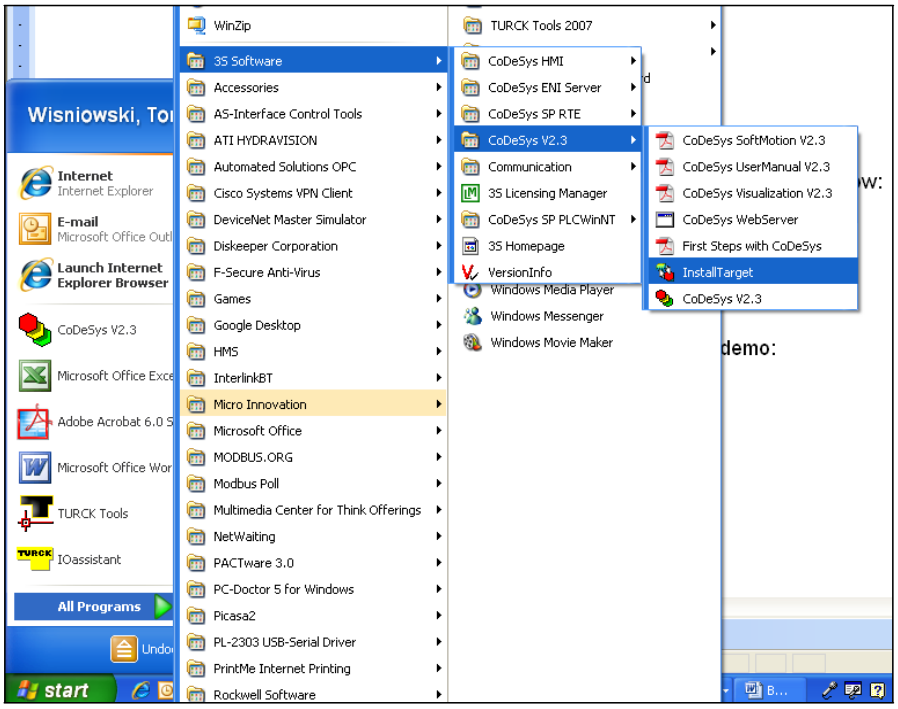


- BL_ident v0.4.lib – Blident library version 0.4 or higher.

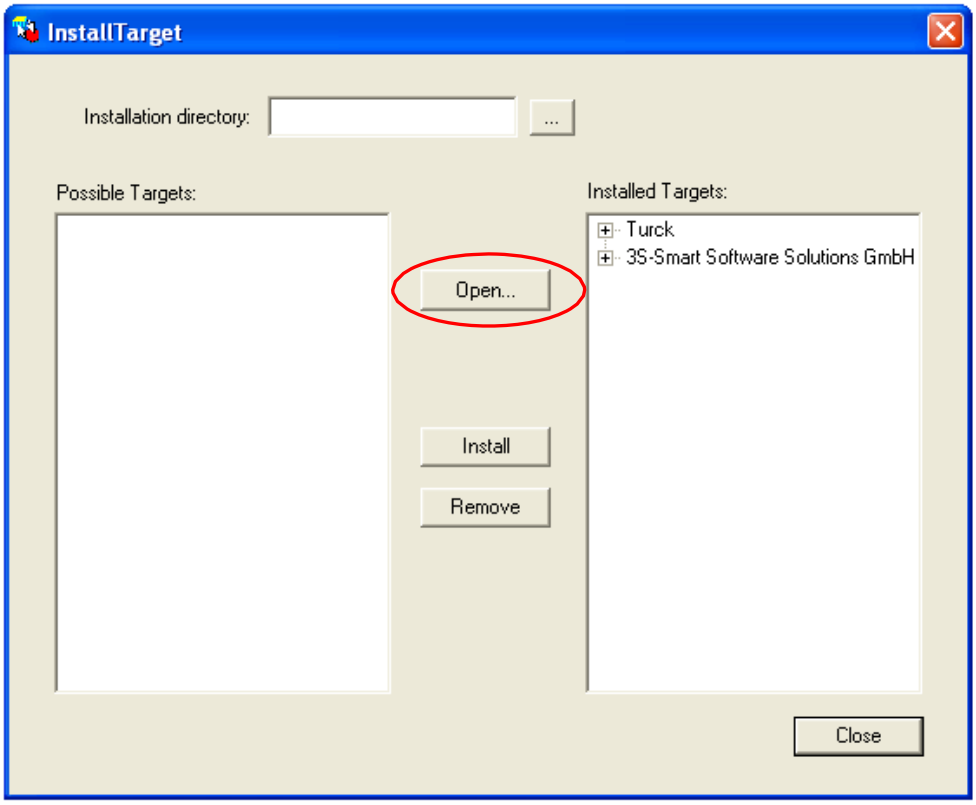
4.2.2.1 Install Target

Download the BL67_PG_EN_target.zip. There is no specific folder that these files need to be saved in. To install the target files into the CoDeSys software follow the instructions below.

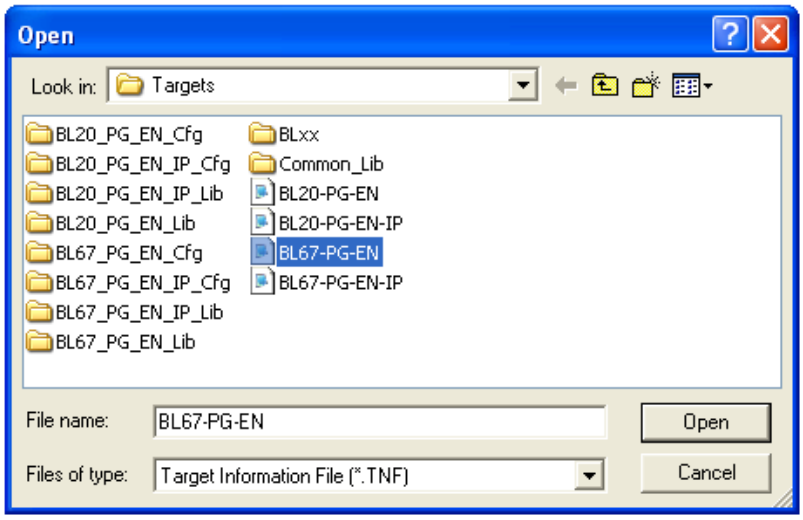
1. Open the InstallTarget program.



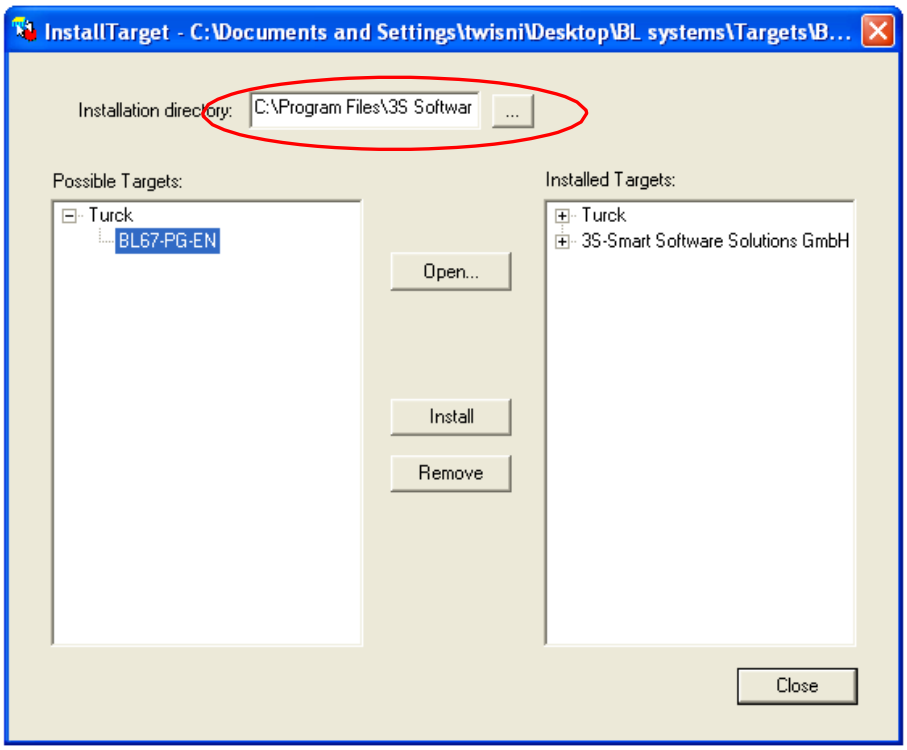
2. Click on the “Open…” button.



3. Find the BL67-PG-EN.tnf file in the folder where the target files have been saved.



4. The BL67-PG-EN target will be located in the "Possible Targets:" window.
   a. The "Installation directory:" will be filled in automatically. To avoid possible errors while opening, compiling and downloading projects into the gateways the default directory should be used. The default directory should be C:\Program Files\3S Software\CoDeSys V2.3\Targets\Turck\.

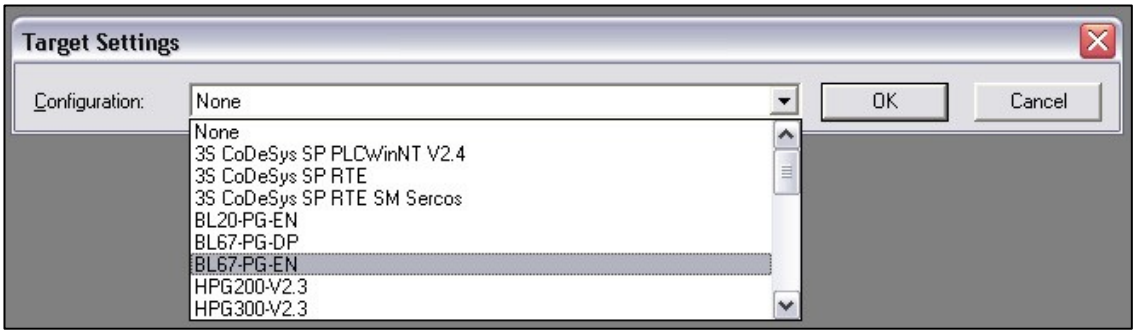


5. Highlight BL67-PG-EN in the "Possible Targets:" and click on "Install" button. The BL67-PG-EN target can now be seen in the "Installed targets:

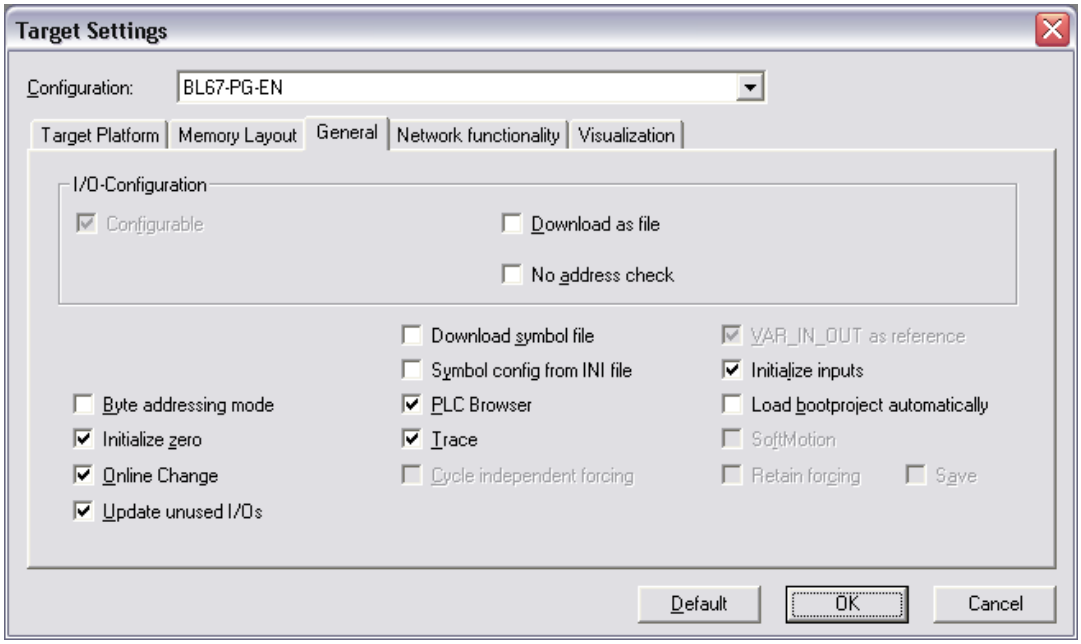
4.2.2.2 Create BLident CoDeSys Project

The following instructions will take you through the steps required to create ***BLident*** CoDeSys project. It is assumed you are familiar with CoDeSys software to complete this instruction set. The CoDeSys software will need to be installed for the following steps. The recommended version is 2.3.5.8. The hardware will not work with versions higher than 3.0.

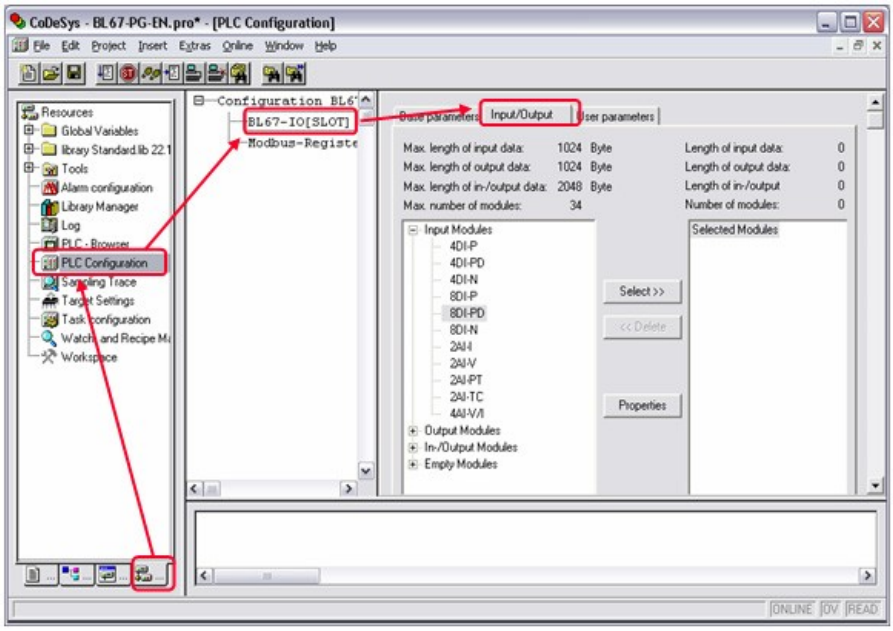
1. Start CoDeSys
2. Open a new project
   a. File >> New
3. Select the BL67-PG-EN target



4. Push OK
   a. Use the default settings
   b. NEVER change the "Byte Addressing Mode" setting



5. Choose a programming language
    a. This example project is written in LD (ladder logic)
6. Use the default Name "PLC_PRG."
    a. PLC_PRG is similar to OB1 in Siemens. This is the program that gets executed automatically. If you change the name, and do not do a TASK configuration, the program will not run.
    b. The Type of the POU should be "Program."
7. Go to Resources tab >> Library Manager and add BL_ident v0.4.lib to the list of available libraries.
8. Go to Resources tab >> PLC Configuration >> Configuration BL67-PG-EN >> BL67-IO[SLOT] >> Input/Output and insert 2RFID-A module



9. For this example we want to be able to see BLident system status, issue commands and send and receive at lease 1 word long register from the GE controller. To accomplish this, we need to create two input and two output Modbus registers in the PLC Configuration.
    a. Resources tab >> PLC Configuration >> Configuration BL67-PG-EN >> Modbus_Registers[SLOT] >> Input/Output tab
    b. Insert two input and two output registers by highlighting the registers and clicking the "Select>>" button
10. The BL67-PG-EN gateway communicates with a Mobdus Master via the Modbus Registers that are created in the PLC configuration. In the previous steps, two input and two output registers were created in the configuration. These registers were aliased as Modbus_In and Modbus_Out respectively and used in the program. The Modbus I/O registers that are created in the PLC configuration are mapped to the PC, PLC or other Modbus masters according to the tables below.

PC or PLC Modbus Registers	⟺	BL67-PG-EN Modbus Registers
Inputs (Read only)	⟸	Output Registers

16384	Output Register 0
16385	Output Register 1
16386	Output Register 2
16387	Output Register 3
…	…
17407	Output Register 1023

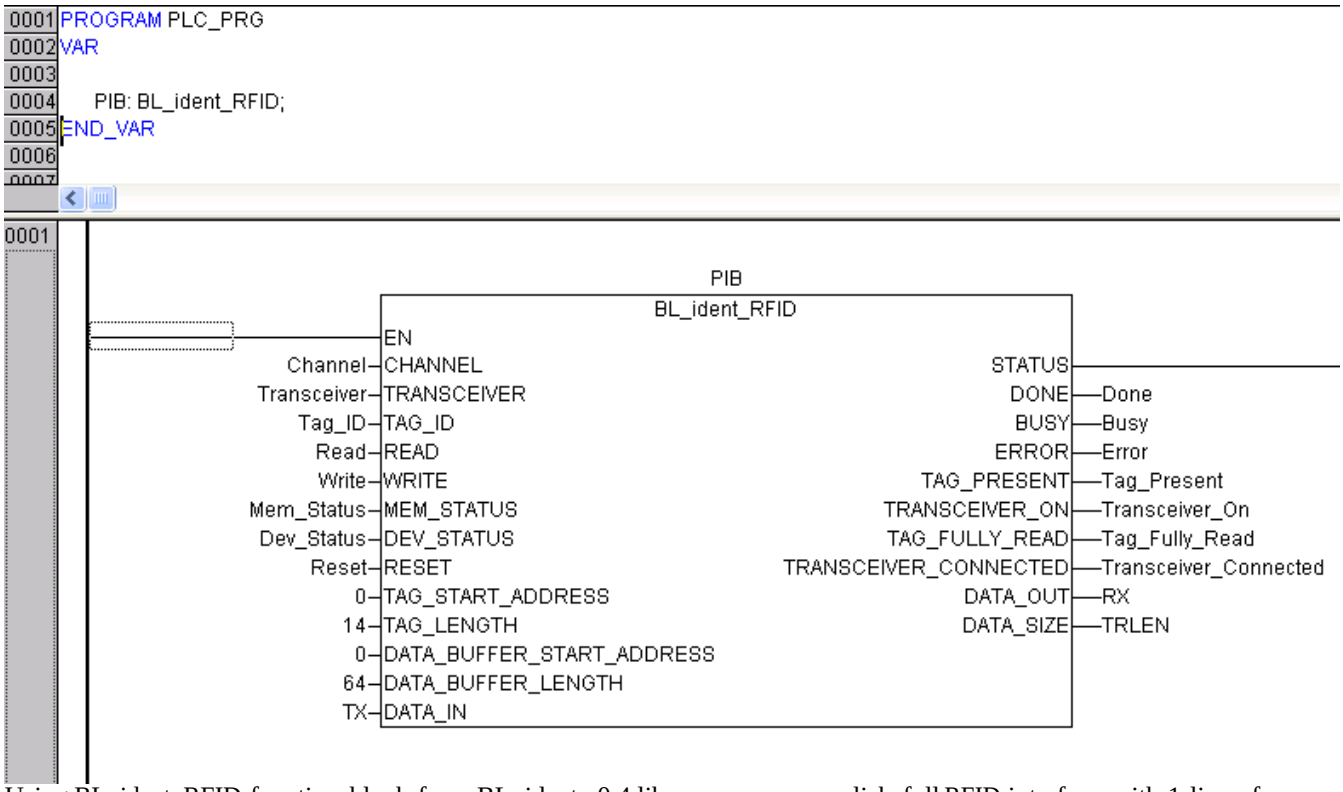
Outputs (Read and Write)		Input Registers
17408	⇨	Input Register 0
17409		Input Register 1
17410		Input Register 2
17411		Input Register 3
…		…
18431		Input Register 1023

Table 4.1: BL67 Modbus Mapping

11. To associate variables in our program with Modbus registers, go to Resources tab >> Global Variables and add the following to the list:

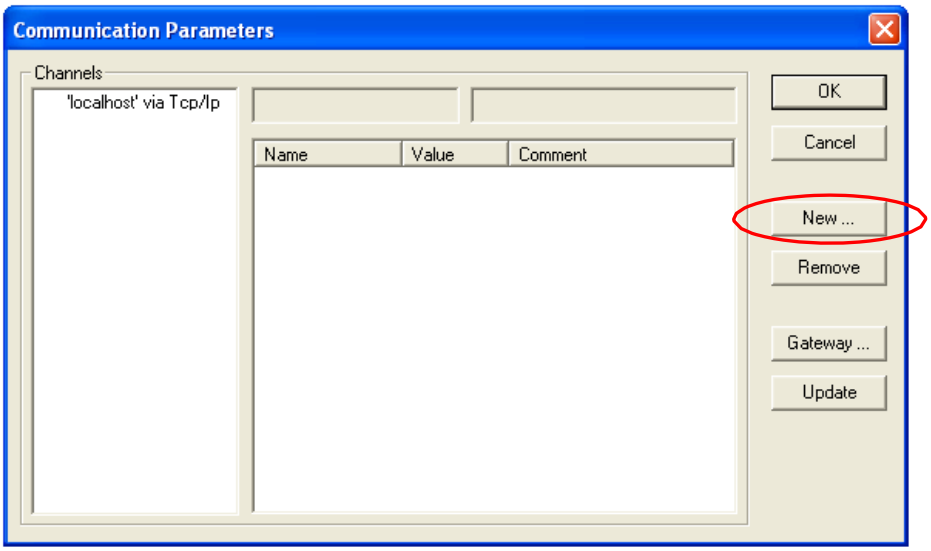
```
Channel: INT:=0;
Transceiver AT %IX4.0: BOOL;
Tag_ID AT %IX4.1: BOOL;
Read AT %IX4.2: BOOL;
Write AT %IX4.3: BOOL;
Mem_Status AT %IX4.4: BOOL;
Dev_Status AT %IX4.5: BOOL;
Reset AT %IX4.6: BOOL;
Start_Address: DINT:=0;
Length: INT:=13;
TX AT %IW3: ARRAY [0..1023] OF BYTE;
Status: DWORD;

Done AT %QX4.0: BOOL;
Busy AT %QX4.1: BOOL;
Error AT %QX4.2: BOOL;
Tag_Present AT %QX4.3: BOOL;
Transceiver_On AT %QX4.4: BOOL;
Tag_Fully_Read AT %QX4.5: BOOL;
Transceiver_Connected AT %QX4.6: BOOL;
RX AT %QW3: ARRAY [0..1023] OF BYTE;
TRLEN: DINT;
```

12. Go to POUs tab and create the following ladder logic program in the PLC_PRG:

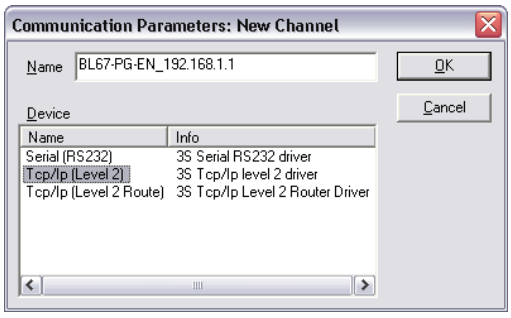


Using BL_ident_RFID function block from BL_ident v0.4.lib you can accomplish full RFID interface with 1 line of ladder logic. Notice use of global variable that were previously declared in step 11.

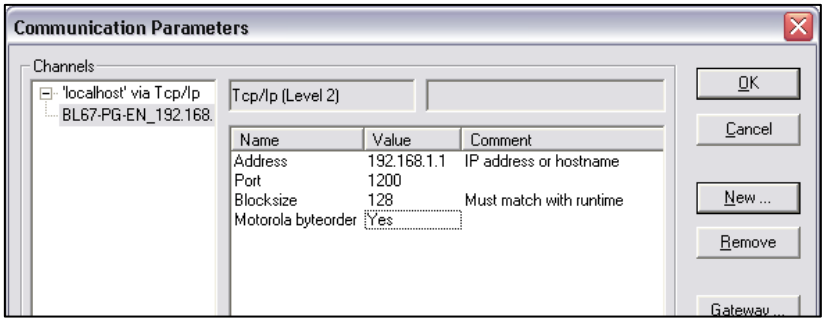
13. Compile the project by going to Project >> Build.
14. Verify Communication Parameters of your BL67 program.
    a. Open the "Communication Parameters" dialogue box.
    b. Online >> Communication Parameters
    c. Click on "New"



d. Enter a name, select “Tcp/Ip (Level 2)” and click “OK”



e. Enter the IP Address of the gateway, change the Motorola byteorder to Yes and click “OK”



15. Login to the BL67-PG-EN gateway, download the run the project

5 GE Fanuc Series 90™-30 PLC Configuration

5.1 Hardware Requirements

- CPU 5 Slot Base part # IC693CHS397
- CPU 366 Module with Profibus DPV1 scanner part # IC693CPU366
- Ethernet Interface TCP/IP Module part # IC693CMM321
- Power Supply part # IC693PWR330
- Horner SNP to RS-232 Programming Cable part # HE693SNPCBL
- Ethernet Cable with RJ45 plug

Assemble all component of the system referring to standard GE operation instructions. Make sure to place Ethernet Interface module in Slot 2 of the Main Rack (right after CPU block).

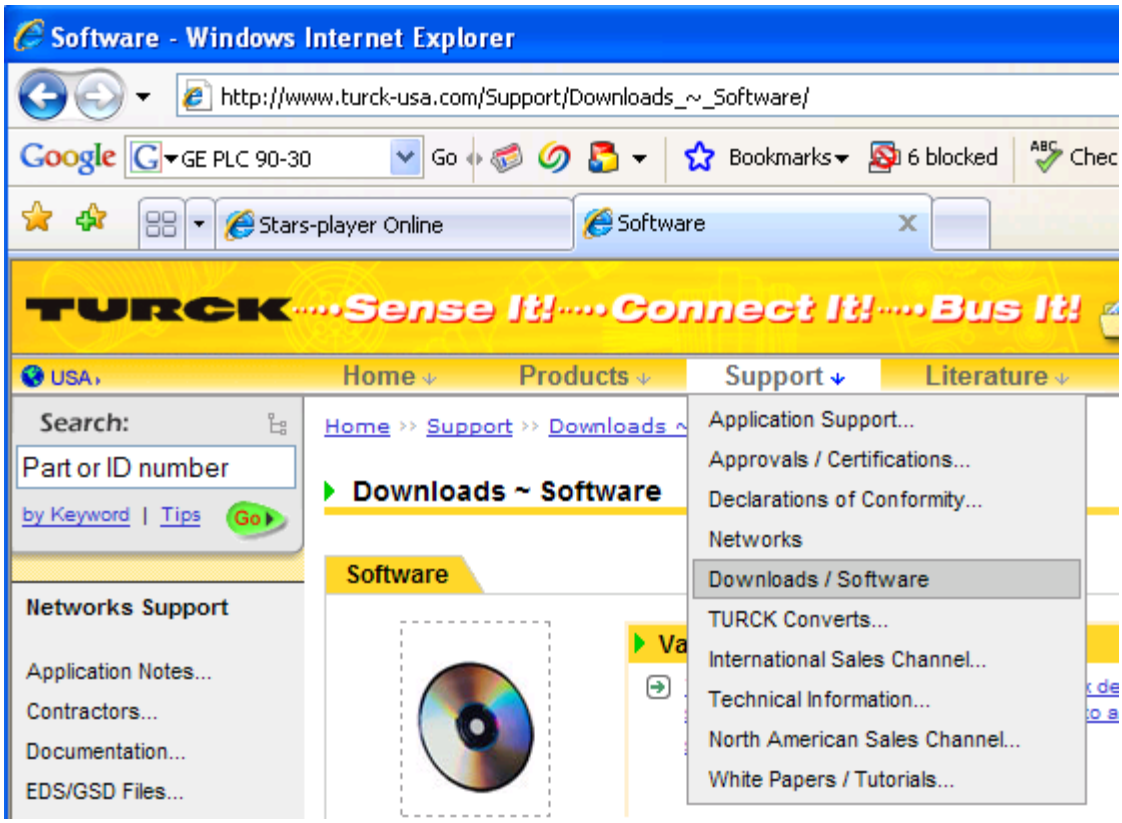
5.2 Software Requirements

- GE Fanuc Logic Developer – PLC version 5.60

5.2.1 Software Setup

The GE Series 90™-30 PLC is going act as a Modbus TCP/IP master to BL67 Programmable gateway. As described previously, all RFID interface logic is realized within BL67 program. Therefore, GE Series 90™-30 PLC program will be responsible for higher level control of the ***BLident*** system. The GE Series 90™-30 PLC program will allow you to see operational status of the ***BLident*** system, issue write and read commands, view 1 receive buffer and send 1 transmit buffer back to the ***BLident*** system.

- The project for GE Series 90™-30 PLC can be downloaded from www.turck.com website and Restored in your Logic Developer – PLC environment.

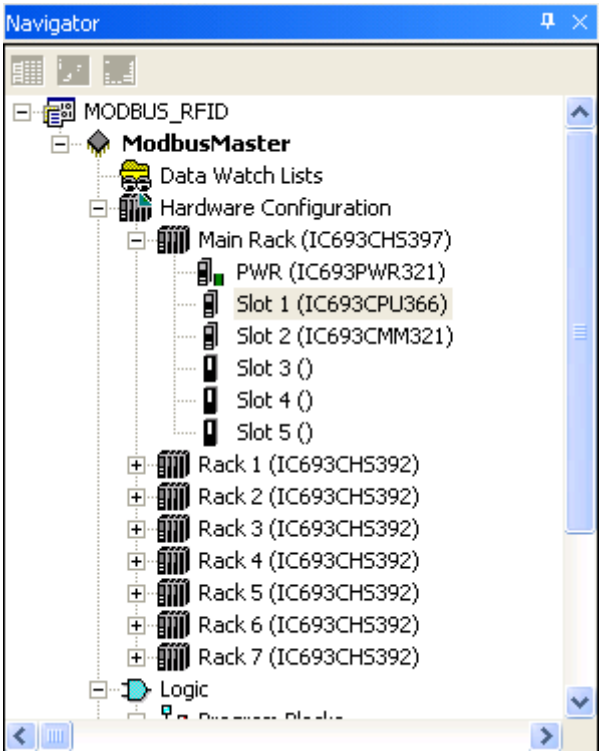


TURCK website Download/Software

- If you wish to write project from scratch, follow instruction set in the next subsection.

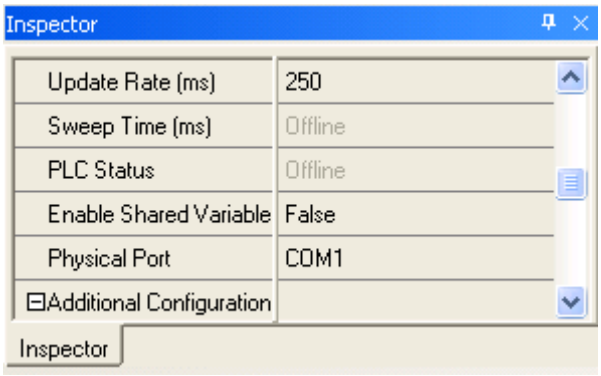
5.2.2 Create GE Series 90™-30 PLC Project

1. Create new project in Logic Developer PLC.
2. Setup Hardware Configuration in the Navigator window. When all done, the Hardware Configuration should look like this:



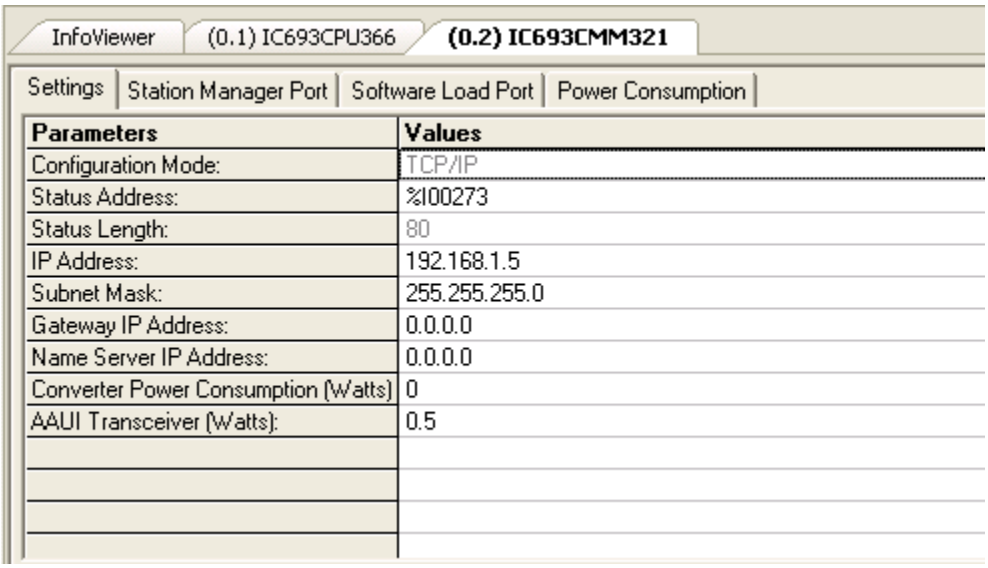
Logic Developer PLC Hardware Configuration

3. Right click on Target and select Properties. In the Inspector window configure the Physical Port parameter.

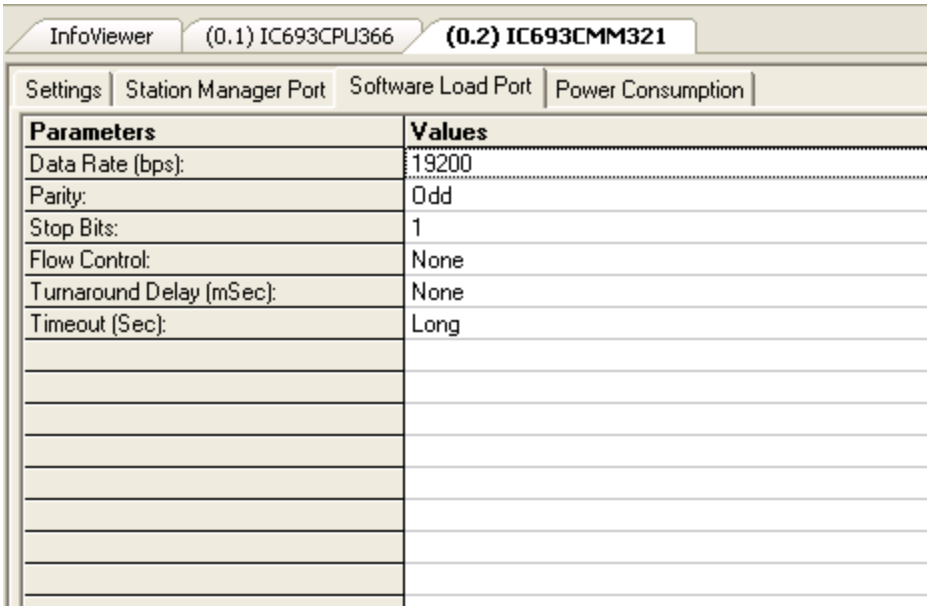


Logic Developer PLC Target Properties in Inspector window

4. Right click on Slot 2 (IC693CMM321) in the Navigator window and select Configure. Setup IP Address and Subnet parameters under Settings tab to work with your network configuration. Make sure that IP Address does not conflict with any other devices on your network. Also, configure Software Load tab parameter to match your Physical Port settings.

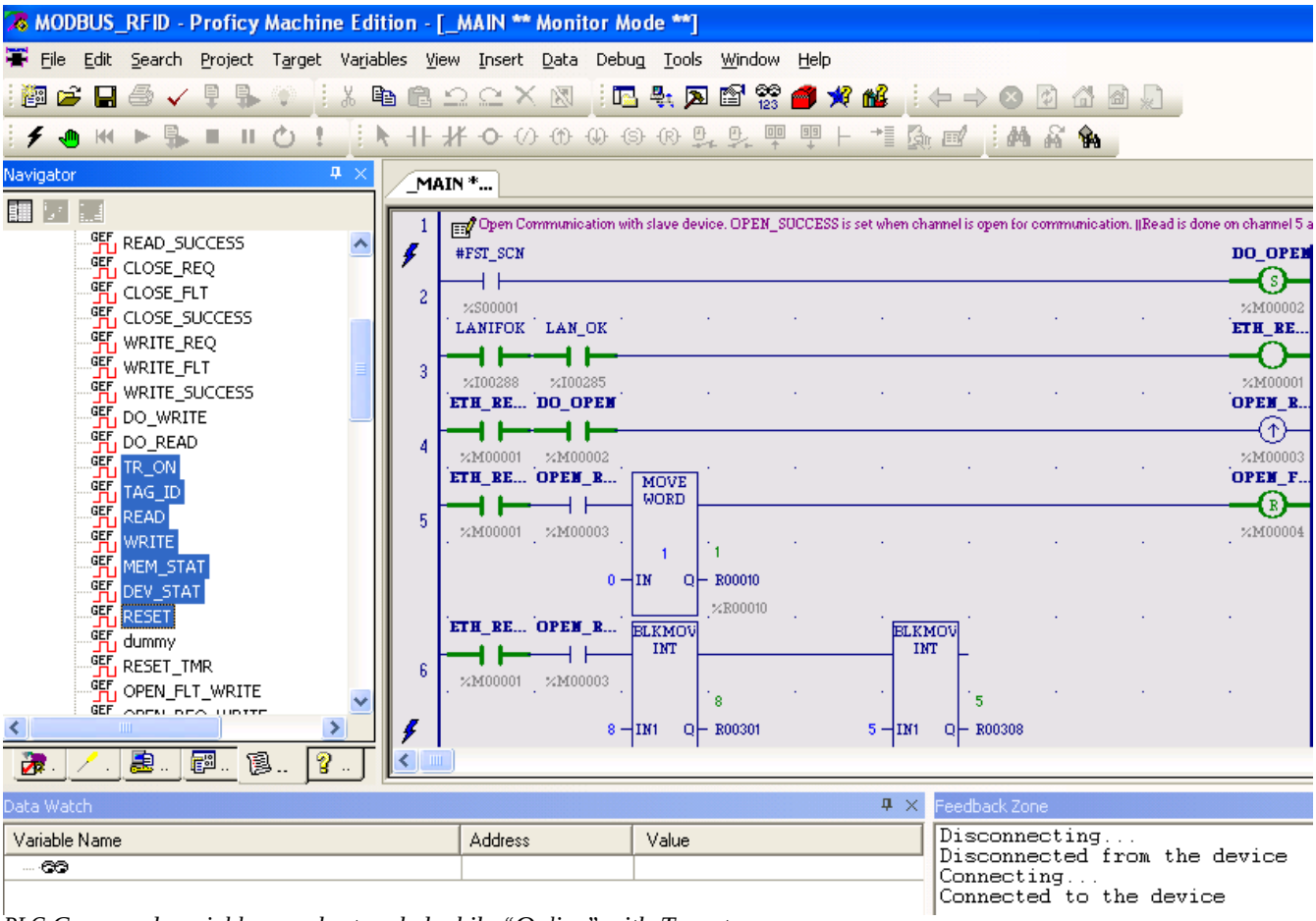


Configuration of Ethernet Interface, Settings tab



Configuration of Ethernet Interface, Software Load Port tab

5. Go to Variables tab of the Navigator window and declare all variables required for the project. See Appendix A for the detailed variable list.
6. Go to Logic->Program Blocks in Navigator window and double click on _MAIN to open the mail ladder logic program for the controller. See Appendix B for the detailed program description. Notice, that BL67 IP Address (192.168.1.1) is set within the program logic using COM_REQ function block (run 6 and 12).
7. Validate the Target, download it to the controller and start Run the program.
8. In order to be able to set variables in the PLC Set Programmer Mode while "Online" with the Target.
9. To issue commands to the BLident system, toggle PLC command variables TR_ON, TAG_ID, READ, WRITE, MEM_STAT, DEV_STAT, or RESET.



PLC Command variables can be toggled while "Online" with Target

10. To turn transceiver on, set TR_ON variable to 1. Observer status of the transceiver head.
11. To write data to a tag and read it back:
    a. Set RFID_INPUTS[1] to a value
    b. Set WRITE variable to 1.

    Present tag to the transceiver head for write operation.
    Reset WRITE variable to 0.
    Set READ variable to 1.
    Present tag to the transceiver head for read operation.
    Reset WRITE variable to 0.
    Verify that value of RFID_INPUTS[1] is now equal to RFID_OUTPUTS[1]
12. To see Operational Status of the RFID system
13. See value of RFID_OUTPUTS[0] variable.

6 Appendix A GE Series 90™-30 PLC Variable List for program _MAIN

Name	DataType	IOAddress
CLOSE_FLT	BOOL	%M00010
CLOSE_FLT_WRITE	BOOL	%M00254
CLOSE_REQ	BOOL	%M00009
CLOSE_REQ_WRITE	BOOL	%M00255
CLOSE_SUCCESS	BOOL	%M00011
CLOSE_SUCCESS_WRITE	BOOL	%M00256
DEV_STAT	BOOL	%M00142
DO_CLOSE	BOOL	%M00257
DO_OPEN	BOOL	%M00002
DO_READ	BOOL	%M00130
DO_WRITE	BOOL	%M00129
ETH_READY	BOOL	%M00001
LAN_OK	BOOL	%I00285
LANIFOK	BOOL	%I00288
MEM_STAT	BOOL	%M00141
OPEN_FLT	BOOL	%M00004
OPEN_FLT_WRITE	BOOL	%M00251
OPEN_REQ	BOOL	%M00003
OPEN_REQ_WRITE	BOOL	%M00252
OPEN_SUCCESS	BOOL	%M00005
OPEN_SUCCESS_WRITE	BOOL	%M00253
R00010	WORD	%R00010
R00011	WORD	%R00011
R00012	WORD	%R00012
R00013	INT	%R00013
R00014	WORD	%R00014
R00015	WORD	%R00015
R00301	INT	%R00301
R00308	INT	%R00308
R00351	INT	%R00351
R00358	INT	%R00358
R00401	INT	%R00401
R00408	INT	%R00408
R00451	INT	%R00451
R00458	INT	%R00458
R00481	INT	%R00481
R00488	INT	%R00488
R00581	INT	%R00581
R00588	INT	%R00588
READ	BOOL	%M00139
READ_FLT	BOOL	%M00007
READ_REQ	BOOL	%M00006
READ_SUCCESS	BOOL	%M00008
READ_TMR	DWORD	%R00712
READ_TMR[0]	DWORD	%R00712

READ_TMR[1]	DWORD	%R00714
READ_TMR[2]	DWORD	%R00716
RESET	BOOL	%M00143
RESET_TMR	BOOL	%M00250
RFID_INPUTS	WORD	%R00717
RFID_INPUTS[0]	WORD	%R00717
RFID_INPUTS[1]	WORD	%R00718
RFID_OUTPUTS	WORD	%R00700
RFID_OUTPUTS[00]	WORD	%R00701
RFID_OUTPUTS[01]	WORD	%R00702
RFID_OUTPUTS[02]	WORD	%R00703
RFID_OUTPUTS[03]	WORD	%R00704
RFID_OUTPUTS[04]	WORD	%R00705
RFID_OUTPUTS[05]	WORD	%R00706
RFID_OUTPUTS[06]	WORD	%R00707
RFID_OUTPUTS[07]	WORD	%R00708
RFID_OUTPUTS[08]	WORD	%R00709
RFID_OUTPUTS[09]	WORD	%R00710
RFID_OUTPUTS[10]	WORD	%R00711
RFID_OUTPUTS[11]	WORD	%R00712
RFID_OUTPUTS[12]	WORD	%R00713
TAG_ID	BOOL	%M00138
TR_ON	BOOL	%M00137
WRITE	BOOL	%M00140
WRITE_FLT	BOOL	%M00013
WRITE_REQ	BOOL	%M00012
WRITE_SUCCESS	BOOL	%M00014
WRITE_TMR	WORD	%R00726
WRITE_TMR[0]	WORD	%R00726
WRITE_TMR[1]	WORD	%R00727
WRITE_TMR[2]	WORD	%R00728

7 Appendix B GE Series 90™-30 PLC ladder logic program _MAIN

The program _MAIN consist of four (4) sections:

1. Initialization of communication channels
2. Read operation of modbus registers from BL67
3. Write operation of modbus registers
4. Close operation of communication channels

The read and write operations are performed every scan cycle of the GE controller with 2 second delay.

7.1 Initialization of Communication Channels

The first 14 lines of the _MAIN program perform initialization of 2 communication channels: 1 for read operations and 1 for write operations. The channels are initialized using COMM_REQ command block that sends "Channel Open" command request to GE Ethernet Interface module.

7.2 Read Operation

The next 15 through 22 lines of the _MAIN program are responsible for reading 3 Modbus registers from BL67, including RFID status bits and 2 words of the RFID receive buffer, into locally declared variable RFID_OUTPUTS. The Modbus registers read from BL67 are addressed in the table 7.2. Notice, Modbus addresses are listed with offset of 1 according to COMM_REQ command block specification.

Modbus Address in GE	**Corresponding Variable in BL67 Program**
16385.1	Done
16385.2	Busy
16385.3	Error
16385.4	Tag_Present
16385.5	Transceiver_On
16385.6	Tag_Fully_Read
16385.7	Transceiver_Connected
16386 - 16387	RX[0-3]

Table 7.2 Modbus Mapping of BL67 Variables to GE Input Registers

7.3 Write Operation

The next 23 through 39 lines of the _MAIN program are responsible for writing 3 Modbus registers from locally declared variable RFID_INPUTS to BL67, including RFID command bits and 2 words of RFID transfer buffer. All of the RFID_INPUTS registers can be manipulated by the GE program logic. The Modbus registers written to BL67 are addressed in the table 7.3. Notice, Modbus addresses are listed with offset of 1 according to COMM_REQ command block specification.

Modbus Address in GE	**Corresponding Variable in BL67 Program**
17409.1	Transceiver
17409.2	Tag_ID
17409.3	Read
17409.4	Write
17409.5	Mem_Status
17409.6	Dev_Status
17409.7	Reset
17410-17411	TX[0-3]

Table 7.3 Modbus Mapping of BL67 Variables to GE Output Registers

7.4 Close Operation

The last 40 through 50 lines of the _MAIN program perform channel close operation based on error condition or explicit close request, CLOSE_REQ variable. Both read and write communication channels are closed and all required registers are cleared.

7.5 Ladder Logic _MAIN Program

1 Open Communication with slave device. OPEN_SUCCESS is set when channel is open for communication.
Read is done on channel 5 and write on channel 6

2 #FST_SCN — S — DO_OPEN %M00002

3 %S00001 LANIFOK — LAN_OK — %M00002 ETH_READY

4 %I00288 ETH_READY — %I00285 DO_OPEN — %M00001 OPEN_REQ

5 %M00001 ETH_READY — %M00002 OPEN_REQ — MOVE WORD — %M00001 — %M00003 — R — %M00003 OPEN_FLT — %M00004

MOVE WORD: 1 → IN; Q → R00010 (%R00010)

6 %M00001 ETH_READY — %M00003 OPEN_REQ — BLKMOV INT — BLKMOV INT

BLKMOV INT (%R00301):

8	IN1
0	IN2
8	IN3
9	IN4
0	IN5
0	IN6
3000	IN7

Q → R00301 (%R00301)

BLKMOV INT (%R00308):

5	IN1
1	IN2
4	IN3
192	IN4
168	IN5
1	IN6
1	IN7

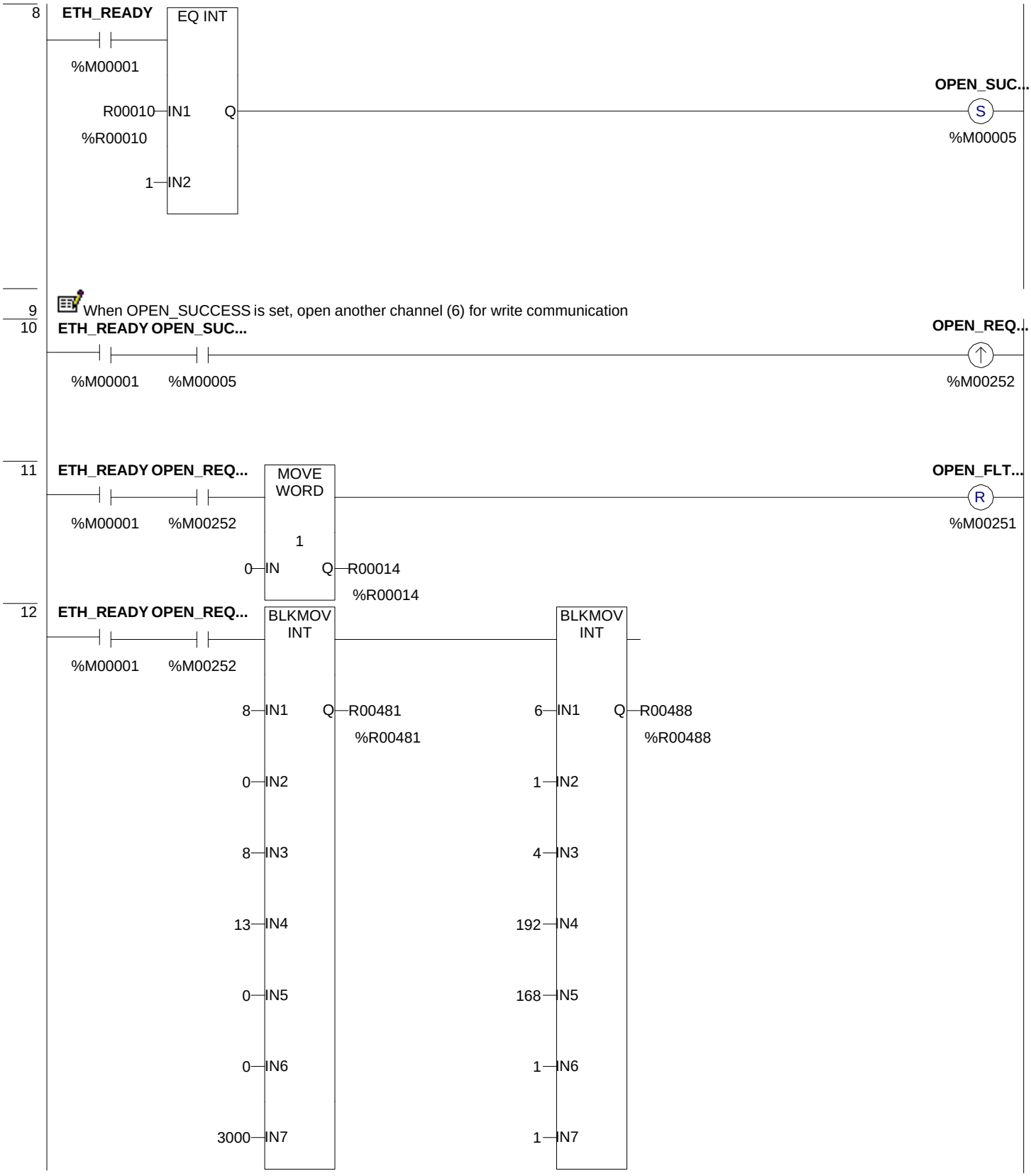
Q → R00308 (%R00308)

7 %M00001 ETH_READY — %M00003 OPEN_REQ — COMM REQ

COMM REQ (%R00301):

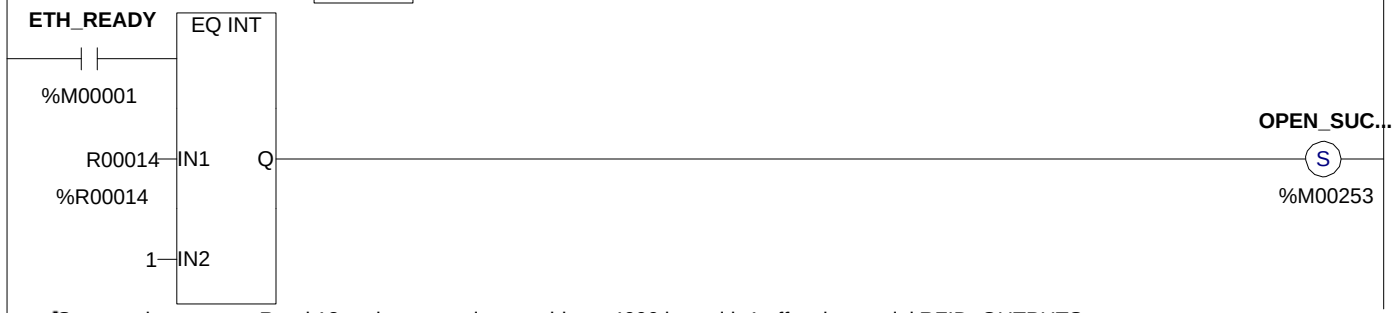
R00301	IN
2	SYSI
0	TASK

FT — S — %M00004 OPEN_FLT


8
ETH_READY
%M00001
EQ INT
R00010
IN1
Q
%R00010
1
IN2
OPEN_SUC...
S
%M00005
9
When OPEN_SUCCESS is set, open another channel (6) for write communication
10
ETH_READY OPEN_SUC...
%M00001
%M00005
OPEN_REQ...
%M00252
11
ETH_READY OPEN_REQ...
%M00001
%M00252
MOVE
WORD
1
0
IN
Q
R00014
%R00014
OPEN_FLT...
R
%M00251
12
ETH_READY OPEN_REQ...
%M00001
%M00252
BLKMOV
INT
8
IN1
Q
R00481
%R00481
0
IN2
8
IN3
13
IN4
0
IN5
0
IN6
3000
IN7
BLKMOV
INT
6
IN1
Q
R00488
%R00488
1
IN2
4
IN3
192
IN4
168
IN5
1
IN6
1
IN7

13

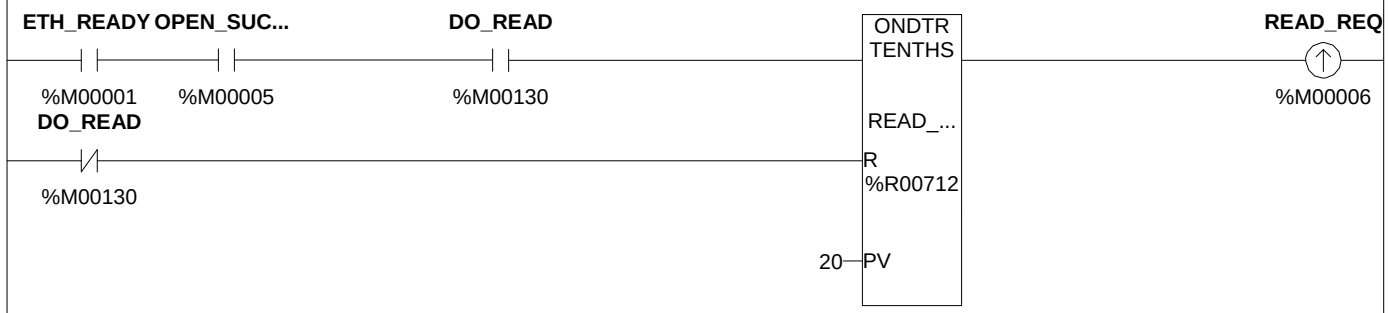
14



15 Start read sequence. Read 13 registers starting at address 4000 hex with 1 offset into varial RFID_OUTPUTS

16

17



18

19

20 **ETH_READY** **READ_REQ**

%M00001 %M00006

BLKMOV INT

8—IN1 Q—R00351

%R00351

0—IN2

8—IN3

10—IN4

0—IN5

0—IN6

3003—IN7

BLKMOV INT

5—IN1 Q—R00358

%R00358

4—IN2

8—IN3

700—IN4

16385—IN5

3—IN6

1—IN7

21 **ETH_READY** **READ_REQ**

%M00001 %M00006

COMM REQ

R00351—IN FT

%R00351

2—SYSI

0—TASK

READ_FLT

S

%M00007

22 **ETH_READY**

%M00001

EQ INT

R00011—IN1 Q

%R00011

1—IN2

READ_SUC...

S

%M00008

23 Start write sequence. Write 3 registers starting with address 4400 with offset of 1. RFID_INPUTS

24 **ETH_READY** **WRITE_REQ**

%M00001 %M00012

DO_WRITE

S

%M00129

25 **ETH_READY** **OPEN_SUC...** **DO_WRITE** ONDTR TENTHS **WRITE_REQ**

%M00001 %M00253 %M00129 WRITE_... %M00012

DO_WRITE R

%M00129 %R00726

20 PV

26 **ETH_READY** **WRITE_REQ** **DO_WRITE**

R

%M00001 %M00012 %M00129

27 Write RFID commands into RFID_INPUT[0].
To set command in PLC, set appropreate resiters:TR_ON, TAG_ID, READ, WRITE, MEM_STAT, DEV_STAT, RESET

28 **WRITE_REQ** MOVE WORD

%M00012

1

0 IN Q RFID_INPU...

%R00717

29 **WRITE_REQ** TR_ON OR WORD

%M00012 %M00137

16#0100 IN1 Q RFID_INPU...

%R00717

RFID_INPU... IN2

%R00717

30 **WRITE_REQ** TAG_ID OR WORD

%M00012 %M00138

16#0200 IN1 Q RFID_INPU...

%R00717

RFID_INPU... IN2

%R00717

31 **WRITE_REQ** READ OR WORD

%M00012 %M00139

16#0400 IN1 Q RFID_INPU...

%R00717

RFID_INPU... IN2

%R00717

32 **WRITE_REQ** WRITE OR WORD

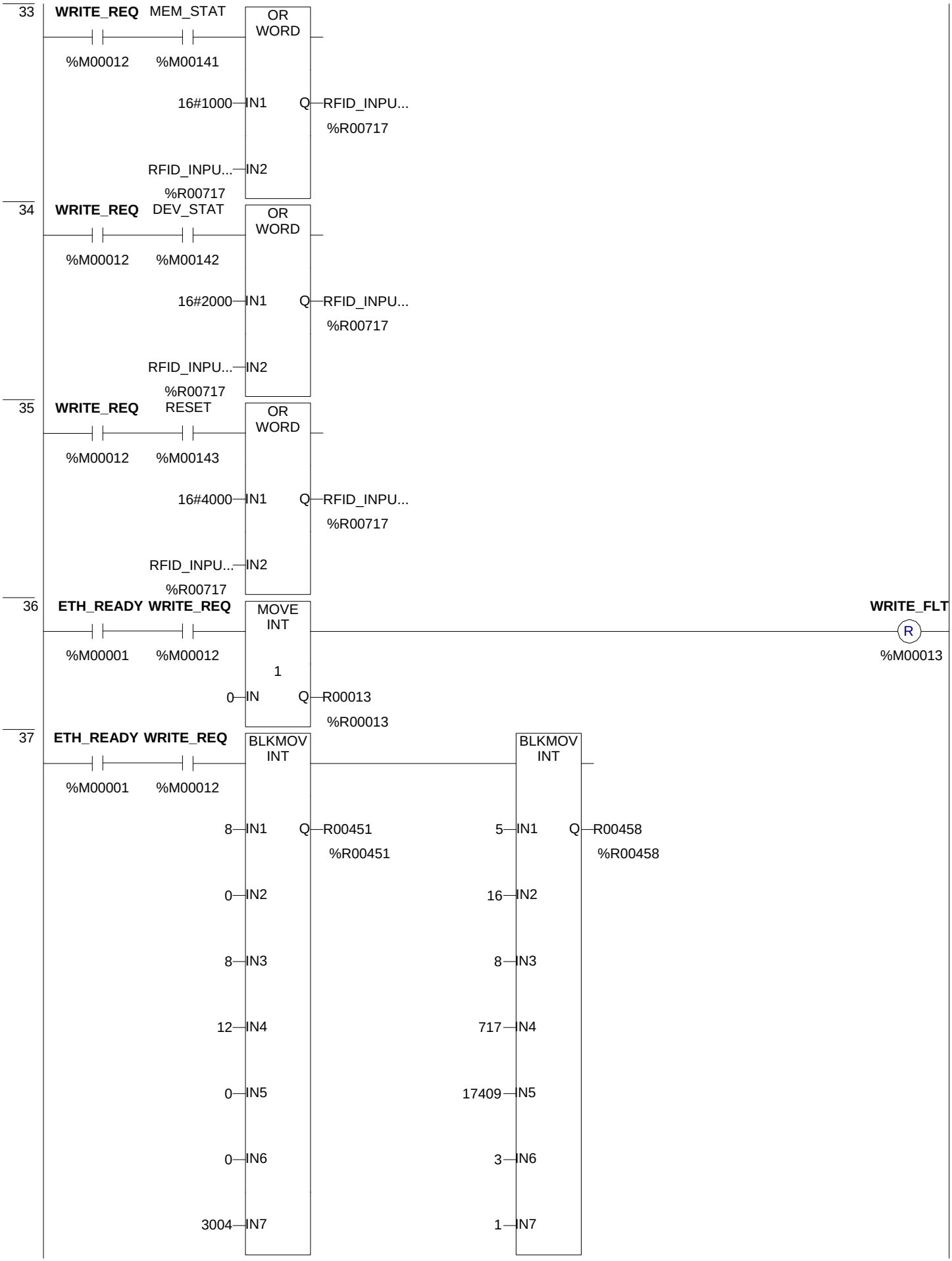
%M00012 %M00140

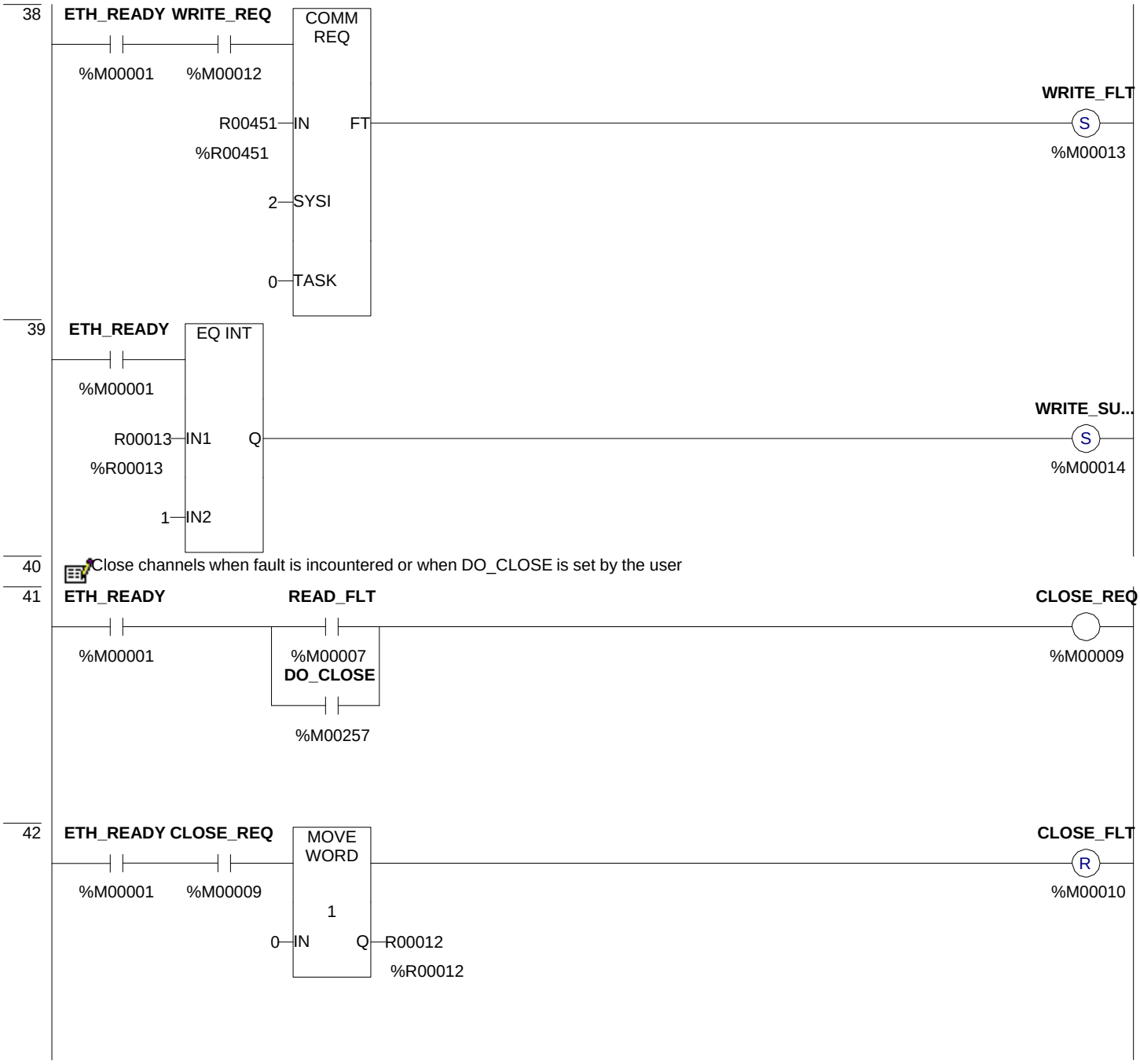
16#0800 IN1 Q RFID_INPU...

%R00717

RFID_INPU... IN2

%R00717


33
WRITE_REQ
MEM_STAT
%M00012
%M00141
OR WORD
16#1000
IN1
Q
RFID_INPU...
%R00717
RFID_INPU...
IN2
%R00717
34
WRITE_REQ
DEV_STAT
%M00012
%M00142
OR WORD
16#2000
IN1
Q
RFID_INPU...
%R00717
RFID_INPU...
IN2
%R00717
35
WRITE_REQ
RESET
%M00012
%M00143
OR WORD
16#4000
IN1
Q
RFID_INPU...
%R00717
RFID_INPU...
IN2
%R00717
36
ETH_READY
WRITE_REQ
%M00001
%M00012
MOVE INT
1
0
IN
Q
R00013
%R00013
WRITE_FLT
R
%M00013
37
ETH_READY
WRITE_REQ
%M00001
%M00012
BLKMOV INT
8
IN1
Q
R00451
%R00451
0
IN2
8
IN3
12
IN4
0
IN5
0
IN6
3004
IN7
BLKMOV INT
5
IN1
Q
R00458
%R00458
16
IN2
8
IN3
717
IN4
17409
IN5
3
IN6
1
IN7


38
ETH_READY
WRITE_REQ
%M00001
%M00012
COMM REQ
R00451
IN
FT
%R00451
2
SYSI
0
TASK
WRITE_FLT
S
%M00013
39
ETH_READY
%M00001
EQ INT
R00013
IN1
Q
%R00013
1
IN2
WRITE_SU...
S
%M00014
40
Close channels when fault is incountered or when DO_CLOSE is set by the user
41
ETH_READY
%M00001
READ_FLT
%M00007
DO_CLOSE
%M00257
CLOSE_REQ
%M00009
42
ETH_READY
CLOSE_REQ
%M00001
%M00009
MOVE WORD
1
0
IN
Q
R00012
%R00012
CLOSE_FLT
R
%M00010

43 **ETH_READY CLOSE_REQ**

%M00001 %M00009

BLKMOV INT

2—IN1 Q—R00401

%R00401

0—IN2

8—IN3

11—IN4

0—IN5

0—IN6

3001—IN7

BLKMOV INT

5—IN1 Q—R00408

%R00408

0—IN2

0—IN3

0—IN4

0—IN5

0—IN6

0—IN7

44 **ETH_READY CLOSE_REQ**

%M00001 %M00009

COMM REQ

R00401—IN FT

%R00401

2—SYSI

0—TASK

CLOSE_FLT

S

%M00010

45 **ETH_READY**

%M00001

EQ INT

R00012—IN1 Q

%R00012

1—IN2

CLOSE_SU...

S

%M00011

46 **ETH_READY**

%M00001

WRITE_FLT

%M00013

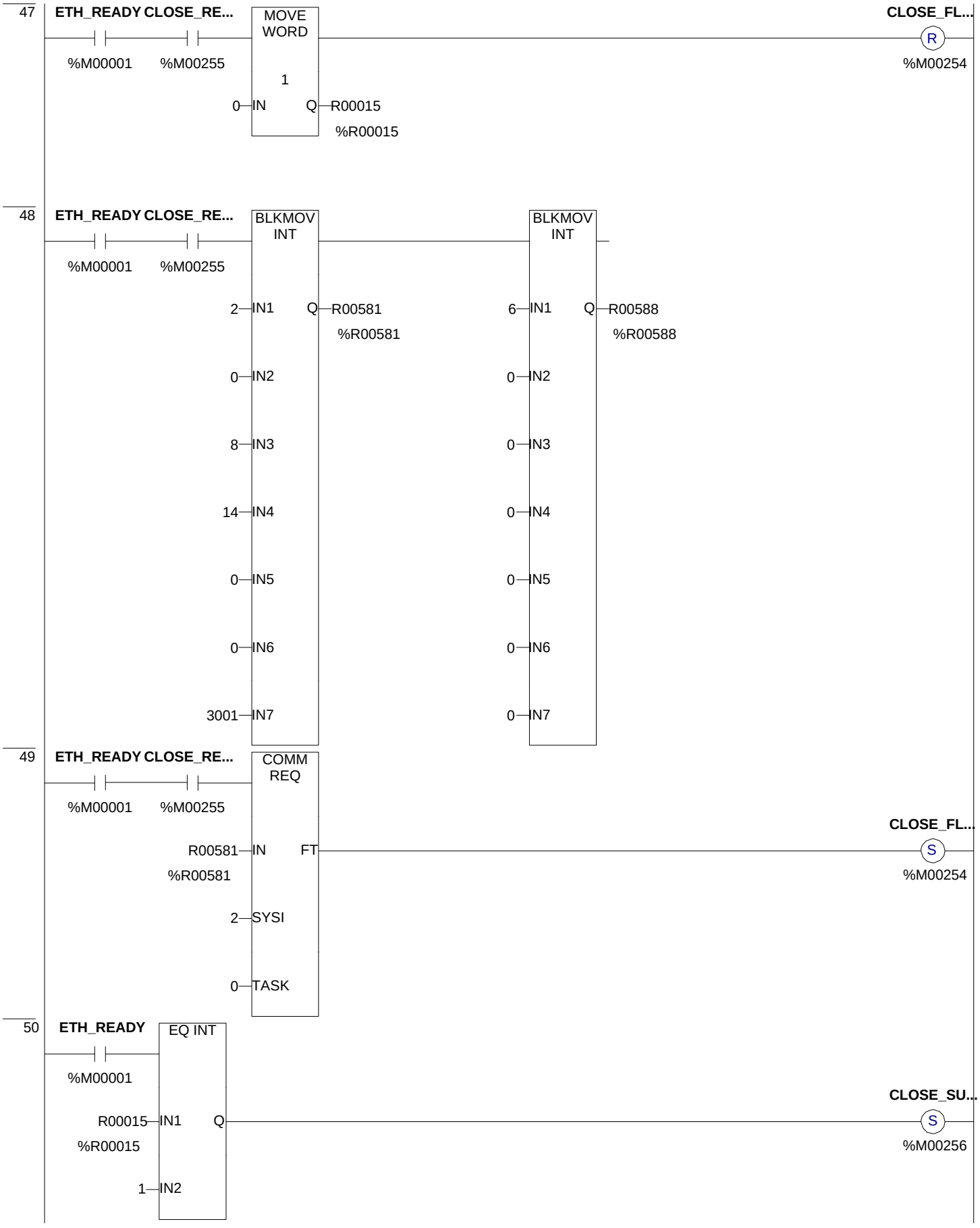
DO_CLOSE

%M00257

CLOSE_RE...

S

%M00255


47
ETH_READY CLOSE_RE...
%M00001
%M00255
MOVE
WORD
1
0
IN
Q
R00015
%R00015
CLOSE_FL...
R
%M00254
48
ETH_READY CLOSE_RE...
%M00001
%M00255
BLKMOV
INT
2
IN1
Q
R00581
%R00581
0
IN2
8
IN3
14
IN4
0
IN5
0
IN6
3001
IN7
BLKMOV
INT
6
IN1
Q
R00588
%R00588
0
IN2
0
IN3
0
IN4
0
IN5
0
IN6
0
IN7
49
ETH_READY CLOSE_RE...
%M00001
%M00255
COMM
REQ
R00581
IN
FT
%R00581
2
SYSI
0
TASK
CLOSE_FL...
S
%M00254
50
ETH_READY
%M00001
EQ INT
R00015
IN1
Q
%R00015
1
IN2
CLOSE_SU...
S
%M00256