

S7 communication with S7-300

FB14 "GET" and FB15 "PUT"

FAQ • April 2011



Service & Support

Answers for industry.

SIEMENS

This entry originates from the Service & Support Portal of Siemens AG, Sector Industry, Industry Automation and Drive Technologies. The conditions of use specified there apply (www.siemens.com/nutzungsbedingungen).

Go to the following link to download this document.

<http://support.automation.siemens.com/WW/view/de/49450152>

Question

How do you program the function blocks FB14 "GET" and FB15 "PUT" in the S7 program of the S7-300 in order to transfer more than 160 bytes of data?

Answer

Follow the instructions and notes listed in this document for a detailed answer to the above question.

Contents

1	Introduction	4
2	Description of the S7 Program	5
2.1	OB100.....	5
2.2	OB1.....	5
2.3	FB111	6

1 Introduction

You must call the function blocks below in the S7 program in order to transfer data with the S7-300 via an S7 connection configured in NetPro:

- FB14 "GET"
- FB15 "PUT"

The function block FB14 "GET" is for reading data from the partner CPU and the function block FB15 "PUT" is for writing data to the partner CPU.

The S7 connection does not have to be configured on both sides, because S7 communication via FB14 "GET" and FB15 "PUT" is based on the server-client principle.

FB14/FB15 for S7 communication via Communications Processor (CP)

If you are using an Industrial Ethernet CP or PROFIBUS CP for data communication via S7 connections in the S7-300 station, then utilize the function blocks FB14 "GET" and FB15 "PUT" from the "SIMATIC_NET_CP" library under "CP 300 > Blocks".

FB14/FB15 for S7 communication via Integrated PROFINET Interface of the CPU

If you are using the integrated PROFINET interface of the CPU for data communication via S7 connections, then utilize function blocks FB14 "GET" and FB15 "PUT" from "Standard Library -> Communication Blocks -> Blocks" with the family "CPU_300".

2 Description of the S7 Program

Only one job at a time can be triggered by the function blocks FB14 "GET" and FB15 "PUT" via an S7 connection configured in NetPro. It is not possible to trigger multiple jobs at the same time with the function blocks FB14 "GET" and FB15 "PUT" via an S7 connection.

The function blocks FB14 "GET" and FB15 "PUT" can only be called sequentially, that is one after the other. A job must first be completed and only then is the next job triggered.

In this sample program the function blocks FB14 "GET" and FB15 "PUT" are called sequentially so that the S7-300 can receive 400 bytes and send 240 bytes of data via an S7 connection.

First create the hardware configuration for your S7-300 station. Configure marker byte 10 as clock marker. The send request is triggered by this clock marker. Save and compile the hardware configuration of your S7-300 station and load it into the CPU.

The S7 program consists of the components below.

Table 2-1

Block	Description
OB100	Startup OB for restart (warm restart)
OB1	The operating system of the CPU processes OB1 cyclically. The cyclic processing of OB1 begins when the startup finishes.
FB111	FB111 calls the function blocks FB14 "GET" and FB15 "PUT" sequentially to receive 400 bytes of data and send 240 bytes of data via an S7 connection.
DB111	Instance data block of the FB111
DB200	The received data is stored in the DB200 data block.
DB201	The sent data is stored in the DB201 data block.

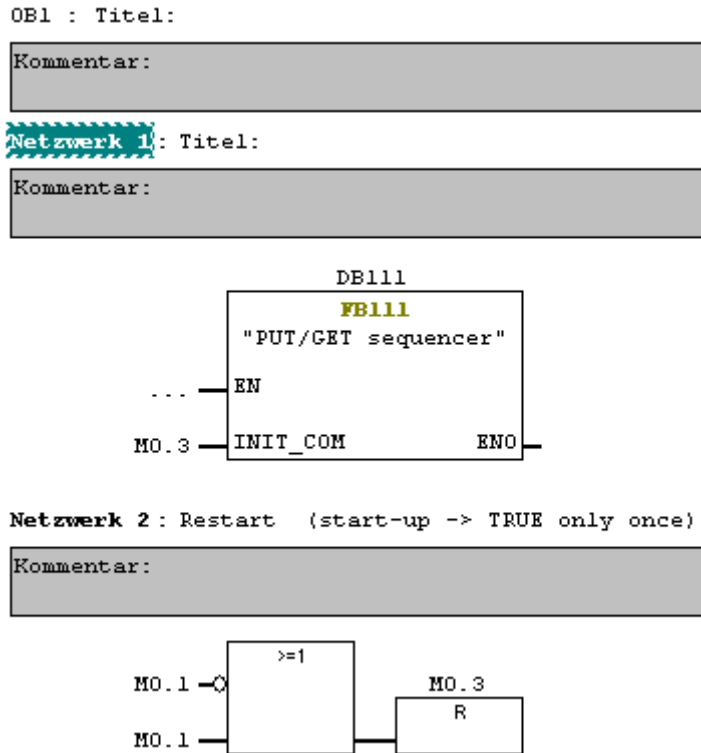
2.1 OB100

The OB100 is a startup OB and is run when the CPU is restarted (warm start). In this OB the first communication trigger is enabled with marker 0.3.

2.2 OB1

OB1 is called cyclically. The FB111 is called in OB1 with the instance data block DB111 and marker M0.3 as INIT_COM parameter. The marker M0.3 is reset in OB1 after the FB111 has been called.

Figure 2-1



2.3 FB111

FB111 is called cyclically in OB1.

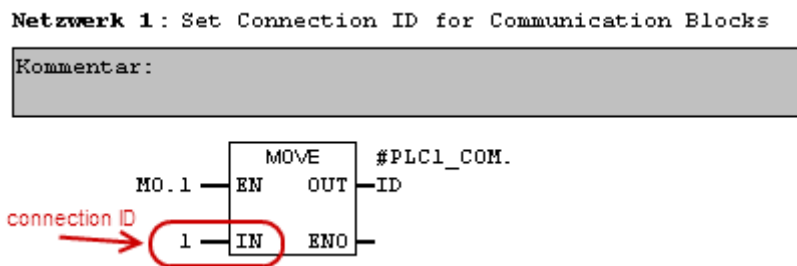
A sequencer is used to call FB14 "GET" three times and FB15 "PUT" twice sequentially, that is one after the other, in FB111.

Defining connection numbers

Via the input parameter "ID" you assign the same connection number to all function blocks. This means that all jobs are triggered one after the other via the same S7 connection.

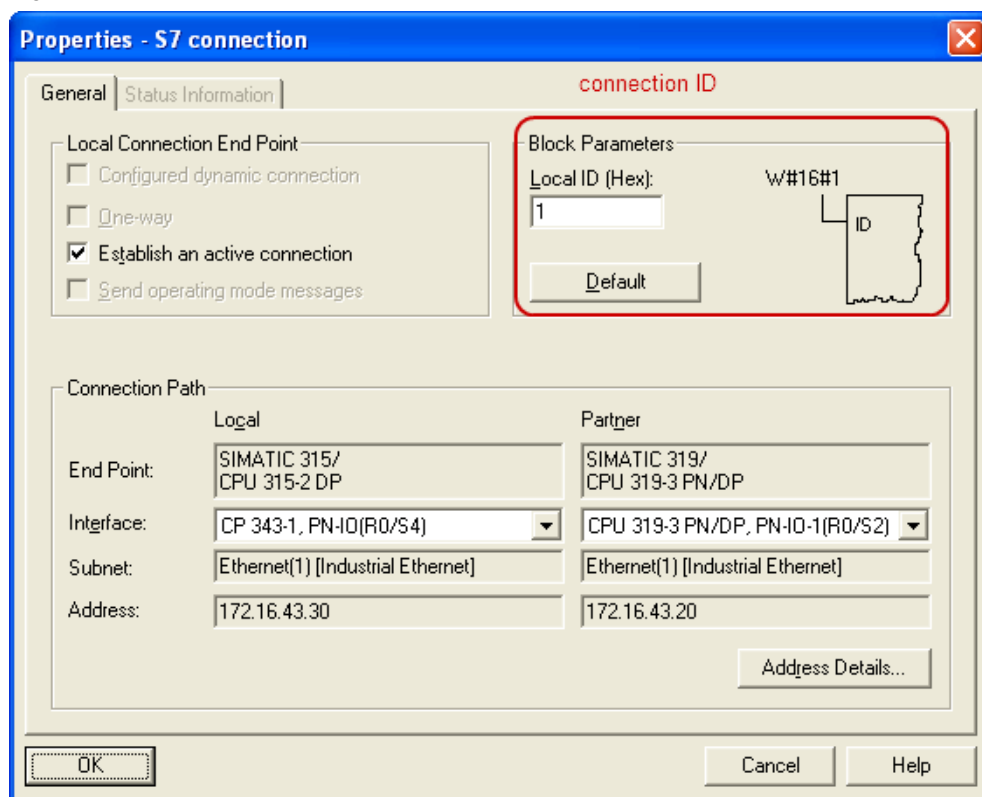
You enter the connection number in Network 1 of FB111. This is stored in a static tag and so in the instance data block DB111.

Figure 2-2



You take the connection number from the Properties dialog of the S7 connection configured in NetPro.

Figure 2-3

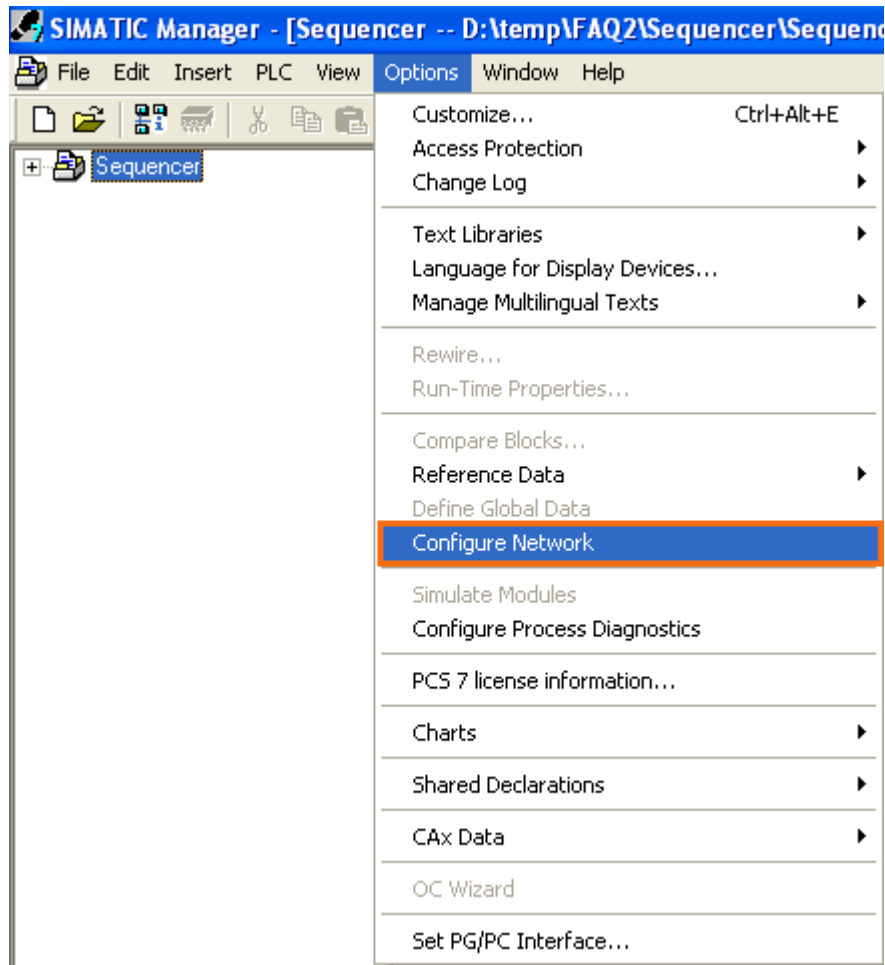


Open the Properties dialog of the S7 connection configured in NetPro

Use the following instruction to open the Properties dialog of the S7 connection configured in NetPro.

In the SIMATIC Manager you select the "Options → Configure Network" menu to open NetPro.

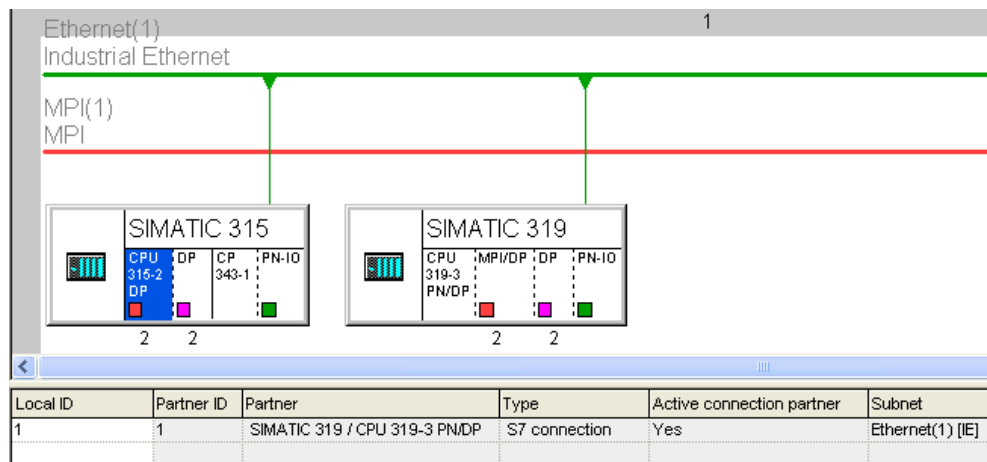
Figure 2-4



In NetPro you select the CPU in the already projected S7-300 station. The connection table shows the already projected connections.

In this sample a S7 connection is configured between two S7-300 stations.

Figure 2-5



In the connection table right-click the already configured S7 connection. Select the "Object Properties" context menu.

Figure 2-6

Local ID	Partner ID	Partner	Type	Active connection partner	Subnet
1			connection	Yes	Ethernet(1) [E]

First job trigger by FB14 "GET"

When the clock marker 10.5 has a positive edge and no other job is already running, then:

- The first FB14 "GET" is activated via the input parameter "REQ".
- The sequencer is initialized.

The input parameter "ADD_R" refers to a data area in the partner CPU from which the data is read. The first 160 bytes of data are read from DB201 starting with address 0.

The input parameter "RD_1" refers to a data area in the local CPU in which the data read is stored. The data read is stored in the local CPU starting with address 0 in DB200.

The output parameters "NDR", "ERROR" and "STATUS" are required for the job evaluation and are only valid in the same cycle.

When the first FB14 "GET" terminates successfully or with an error, then the input parameter "REQ" of the first FB14 "GET" is reset to the value "0" so that the next job can be triggered.

Figure 2-7

Netzwerk 4 : Invoke GET function (number #1)

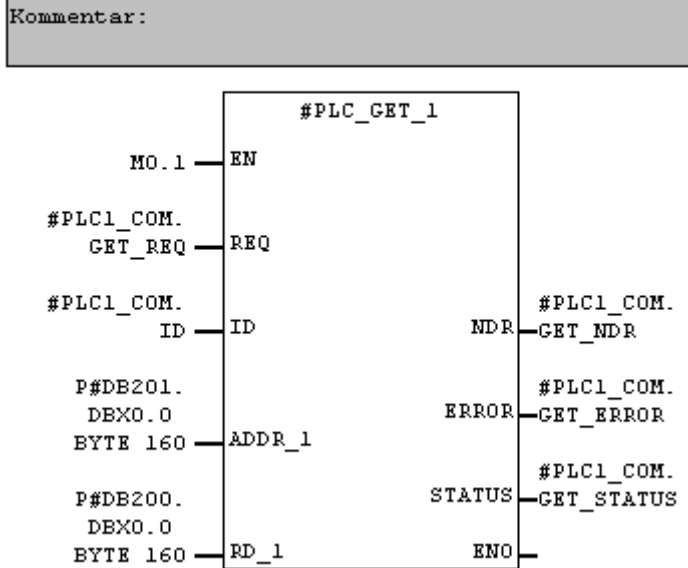
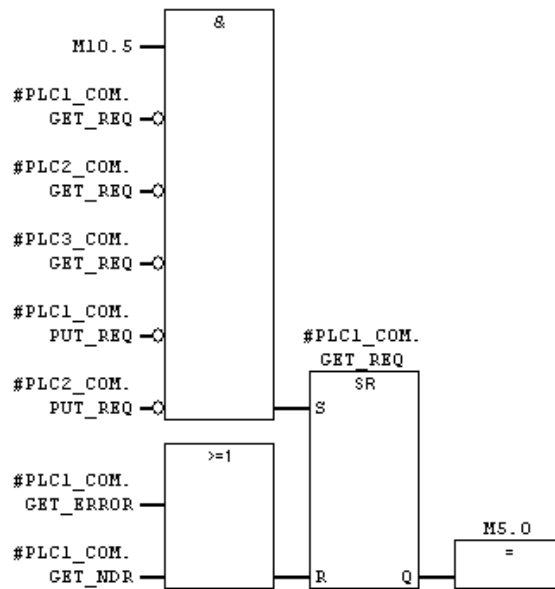


Figure 2-8

Netzwerk 5 : Sequencer initialized by clock marker

Start GET function with the rising edge of the clock marker if the GET function is not BUSY.
The REQ input parameter of the GET function is triggered with a rising edge and BUSY is set as long as the GET function is not completed!

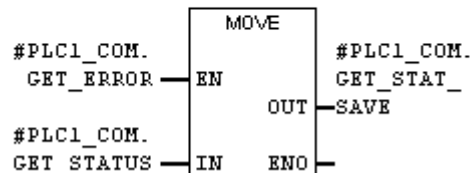


If the first FB14 "GET" is completed with an error, then the value of the output parameter "STATUS" is saved for error analysis.

Figure 2-9

Netzwerk 6: Evaluate STATUS information of the GET function

If ERROR occurred then save STATUS!



Second job trigger by FB14 "GET"

When the first FB14 "GET" is successful and no other job is running, then the second FB14 "GET" is activated via the input parameter "REQ".

The input parameter "ADD_R" refers to a data area in the partner CPU from which the data is read. After the first 160 bytes of data have been received successfully, another 160 bytes of data are read out of DB201 as from address 160.

The input parameter "RD_1" refers to a data area in the local CPU in which the data read is stored. The data read is stored in the local CPU starting with address 160 in DB200.

The output parameters "NDR", "ERROR" and "STATUS" are required for the job evaluation and are only valid in the same cycle.

When the second FB14 "GET" terminates successfully or with an error, then the input parameter "REQ" of the second FB14 "GET" is reset to the value "0" so that the next job can be triggered.

Figure 2-10

Netzwerk 7: Invoke GET function (number #2)

Kommentar:

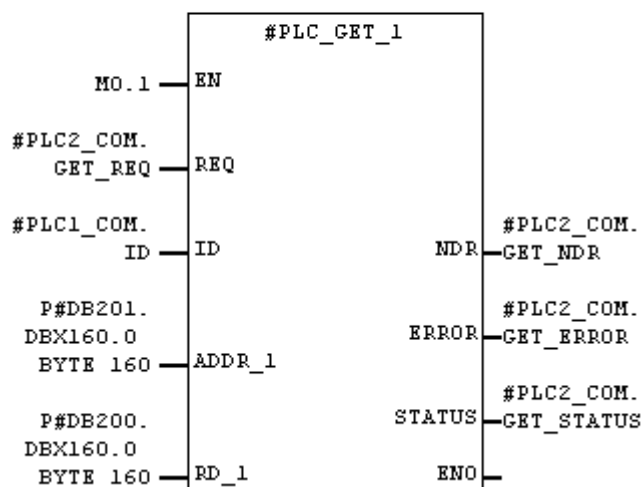
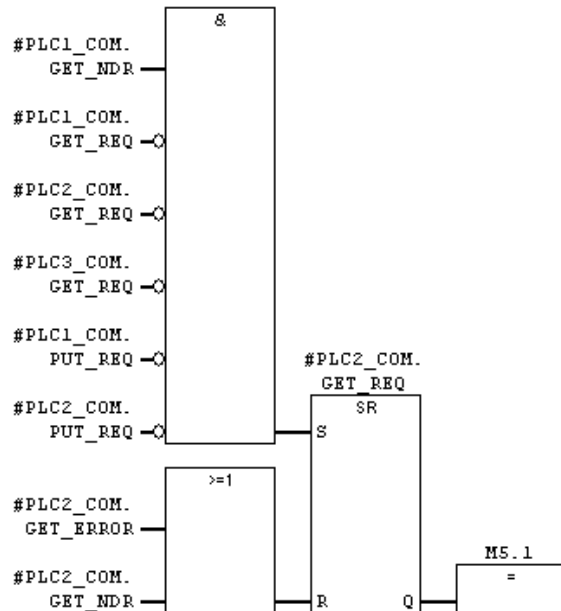


Figure 2-11

Netzwerk 8 : Sequencer initialized by GET#1

Start GET function with the rising edge of the clock marker if the GET function is not BUSY.
 The REQ input parameter of the GET function is triggered with a rising edge and BUSY is set as long as the GET function is not completed!

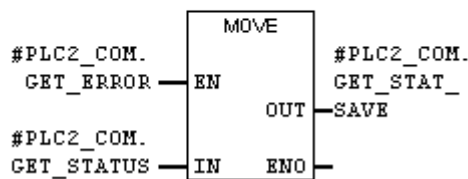


If the second FB14 "GET" is completed with an error, then the value of the output parameter "STATUS" is saved for error analysis.

Figure 2-12

Netzwerk 9 : Evaluate STATUS information of the GET function

If ERROR occurred then save STATUS!



Third job trigger by FB14 "GET"

When the second FB14 "GET" is successful and no other job is running, then the third FB14 "GET" is activated via the input parameter "REQ".

The input parameter "ADD_R" refers to a data area in the partner CPU from which the data is read. After a total of 320 bytes of data have been received successfully, another 80 bytes of data are read out of DB201 as from address 320.

The input parameter "RD_1" refers to a data area in the local CPU in which the data read is stored. The data read is stored in the local CPU starting with address 320 in DB200.

The output parameters "NDR", "ERROR" and "STATUS" are required for the job evaluation and are only valid in the same cycle.

When the third FB14 "GET" terminates successfully or with an error, then the input parameter "REQ" of the third FB14 "GET" is reset to the value "0" so that the next job can be triggered.

Figure 2-13

Netzwerk 10 : Invoke GET function (number #3)

Kommentar:

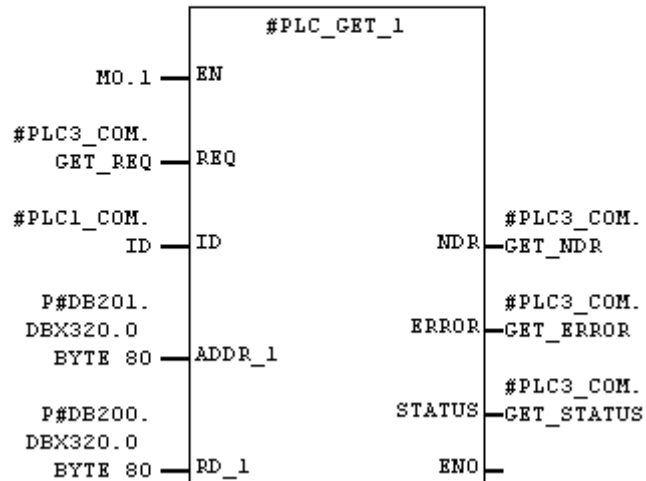
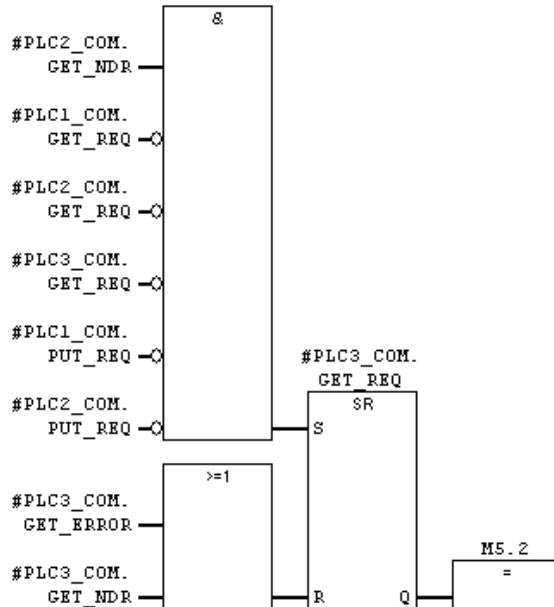


Figure 2-14

Netzwerk 11: Sequencer initialized by GET#2

Start GET function with the rising edge of the clock marker if the GET function is not BUSY.
The REQ input parameter of the GET function is triggered with a rising edge and BUSY is set as long as the GET function is not completed!

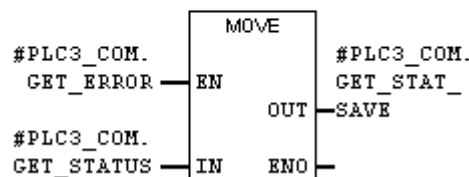


If the third FB14 "GET" is completed with an error, then the value of the output parameter "STATUS" is saved for error analysis.

Figure 2-15

Netzwerk 12: Evaluate STATUS information of the GET function

If ERROR occurred then save STATUS!



First job trigger by FB15 "PUT"

When the third FB14 "GET" has terminated successfully and no other job is running, then:

- A total of 400 bytes of data have been read and stored as received data in DB200 starting with address 0.
- Another job for sending data is triggered by function block FB15 "PUT". The first FB15 "PUT" is activated via the input parameter "REQ".

The input parameter "ADD_R" refers to a data area in the partner CPU to which the data is written. The data written is stored in the partner CPU starting with address 0 in DB200.

The input parameter "SD_1" refers to a data area in the local CPU which contains the data to be sent. 160 bytes of data are sent from DB201 starting with address 0.

The output parameters "DONE", "ERROR" and "STATUS" are required for the job evaluation and are only valid in the same cycle.

When the first FB15 "PUT" terminates successfully or with an error, then the input parameter "REQ" of the first FB15 "PUT" is reset to the value "0" so that the next job can be triggered.

Figure 2-16

Netzwerk 13: Invoke PUT function (number #4)

Kommentar:

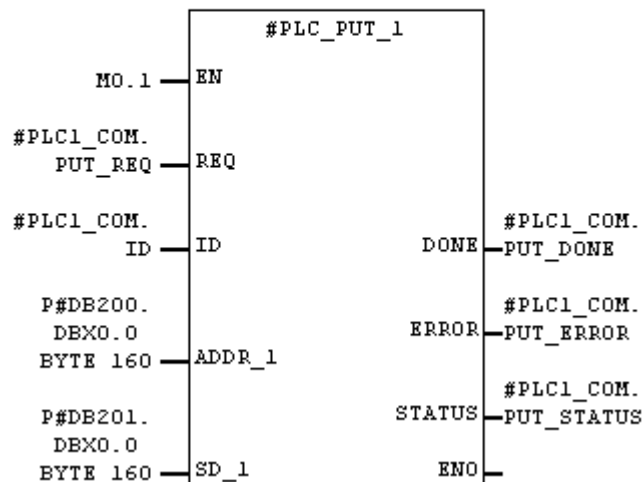
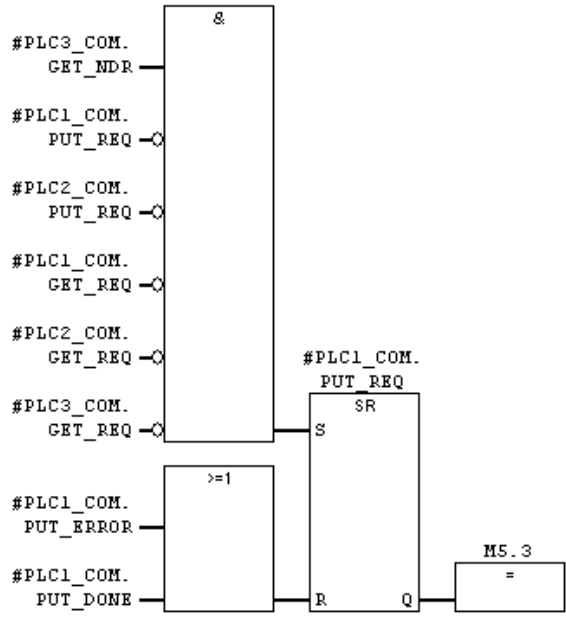


Figure 2-17

Netzwerk 14 : Sequencer initialized by GET#3

Start GET function with the rising edge of the clock marker if the GET function is not BUSY.
 The REQ input parameter of the GET function is triggered with a rising edge and BUSY is set as long as the GET function is not completed!

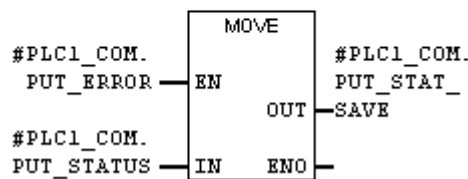


If the first FB15 "PUT" is completed with an error, then the value of the output parameter "STATUS" is saved for error analysis.

Figure 2-18

Netzwerk 15 : Evaluate STATUS information of the PUT function

If ERROR occurred then save STATUS,
 if DONE signals completion then reset BUSY bit for the next cycle!



Second job trigger by FB15 "PUT"

When the first FB15 "PUT" has terminated successfully and no other job is running, then another job for sending data by FB15 "PUT" is triggered. The second FB15 "PUT" is activated via the input parameter "REQ".

The input parameter "ADD_R" refers to a data area in the partner CPU to which the data is written. The data written is stored in the partner CPU starting with address 160 in DB200.

The input parameter "SD_1" refers to a data area in the local CPU which contains the data to be sent. After the first 160 bytes of data have been transferred successfully, another 80 bytes of data are sent from DB201 as from address 160.

The output parameters "DONE", "ERROR" and "STATUS" are required for the job evaluation and are only valid in the same cycle.

When the second FB15 "PUT" terminates successfully or with an error, then the input parameter "REQ" of the second FB15 "PUT" is reset to the value "0" so that the first job can once again be triggered by FB14 "GET" in the next cycle of FB111. The function blocks FB14 "GET" and FB15 "PUT" are once again called sequentially for receiving and sending new data.