

Communication between SIMATIC S5 and SIMATIC S7 over PROFIBUS

FMS protocol

FAQ • February 2011



Service & Support

Answers for industry.

SIEMENS

This entry is from the Service&Support portal of Siemens AG, Sector Industry, Industry Automation and Drive Technologies. The general terms of use (http://www.siemens.com/terms_of_use) apply.

Clicking the link below directly displays the download page of this document.

<http://support.automation.siemens.com/WW/view/en/48066532>

Question

How do you exchange data between SIMATIC S5 and SIMATIC S7 over PROFIBUS using the FMS protocol?

Answer

The instructions and notes listed in this document provide a detailed answer to this question.

Table of Contents

1	Introduction.....	5
2	Overview of the Sample Program	6
	Plant configuration.....	6
	Checking the function of the sample program.....	6
	Overview of functions	7
	Data communication.....	7
	FMS Server and FMS Client	7
	Program overview	7
	User program of the S7-400 (FMS client)	8
	User program of the S5 station (FMS server)	8
	DB5 "Index_DB" in the FMS client	8
	Initialization of the FB6 "WRITE"	8
	Error handling.....	8
	Operating and monitoring.....	8
3	Plant Configuration.....	10
3.1	Overview	10
3.2	Hardware and Software Components Used.....	10
	Hardware components	10
	Required cables and other hardware	11
	Software components.....	11
4	Function Mechanisms of the Sample Program	12
4.1	Working Method of the Sample Program	12
4.1.1	Data Communication.....	12
4.1.2	FMS Client (S7-400 Station) and FMS Server (S5 Station).....	12
4.1.3	Sequence of the User Program in the FMS Client (S7-400 Station) .	12
	Write data	12
	Read data	12
	Increase data.....	12
4.1.4	Sequence of the User Program in the FMS Server (S5 Station)	12
	FB120 "SEND_ALL"	12
	FB121 "RECEIVE_ALL"	12
4.2	Functions and Function Blocks	13
4.2.1	Functions and Function Blocks in the User Program of the S7-400 Station	13
	Task of the function block FB6 "WRITE"	13
	Task of the function block FB3 "READ"	13
	Read out the variable descriptions of the communication partner	13
4.3	Details of the User Program of the FMS Client (S7-400 Station).....	15
4.3.1	Program Sequence	15
4.3.2	Symbol Table	15
	Example	16
	Overview	16
4.3.3	OB100	16
	Overview	17
	Description	17
4.3.4	OB1	17
4.3.5	FC36 "MY_WRITE_CALL"	18
	Overview	18
	Description	18
	Processing of the write job is running	20
4.3.6	FC30 "SAVE_WRITE_ERROR".....	20
4.3.7	FC33 "MY_READ_CALL".....	22
	Overview	22

	Processing of the read job is running.....	24
4.3.8	FC35 "SAVE_READ_ERROR"	24
4.3.9	DB5 "Index_DB"	24
4.4	Details of the User Program of the FMS Server (S5 Station)	26
4.4.1	Program Sequence	26
4.4.2	Configure Communication Variable in the FMS Server	26
4.4.3	OB20	26
4.4.4	OB1	27
4.4.5	FB231	27
5	Installation and Commissioning	29
5.1	Installation of the Sample Program of the S7-400 Station	29
5.2	Installation of the Sample Program of the S5 Station	29
5.3	Installation of the Sample Program for the PC with CP5412 A2.....	29
5.4	Commissioning.....	30
6	How to Use the Sample Program.....	31
	Opening the variable table in the user program of the S7-400	31
	Variable table	32
	Data is successfully sent and received	33
	Errors in sending and receiving the data.....	33
	Send data	33
	Closing the variable table in the user program of the S7-400.....	34
7	Other Notes, Tips and Tricks, etc.....	35
7.1	Notes, Tips and Tricks for the Commissioning and Working of the Sample Program in the FMS Client.....	35
7.1.1	Not Possible to Download Hardware Configuration, Connection or Blocks to the S7-400 CPU	35
7.1.2	Values of the Send and Receive Data Do Not Change in the Variable Table	35
7.1.3	Control Values are Not Applied for the Variables in the Variable Table	36
7.2	Notes, Tips and Tricks for the Commissioning and Working of the Sample Program in the FMS Server	36
7.2.1	"PLC Timeout" Message Appears.....	36
7.2.2	Message "PLC interface on PG not ready"	37
7.2.3	Message "CP interface not ready" in the Configuration Software SINEC COM 5431	37

1 Introduction

This document provides information about examples of PROFIBUS communication between S7 stations and SIMATIC S5 stations using the services WRITE and READ based on the PROFIBUS FMS protocol.

Chapter 2 gives an overview of the sample program.

Chapter 4 gives information about the function mechanisms of the sample program.

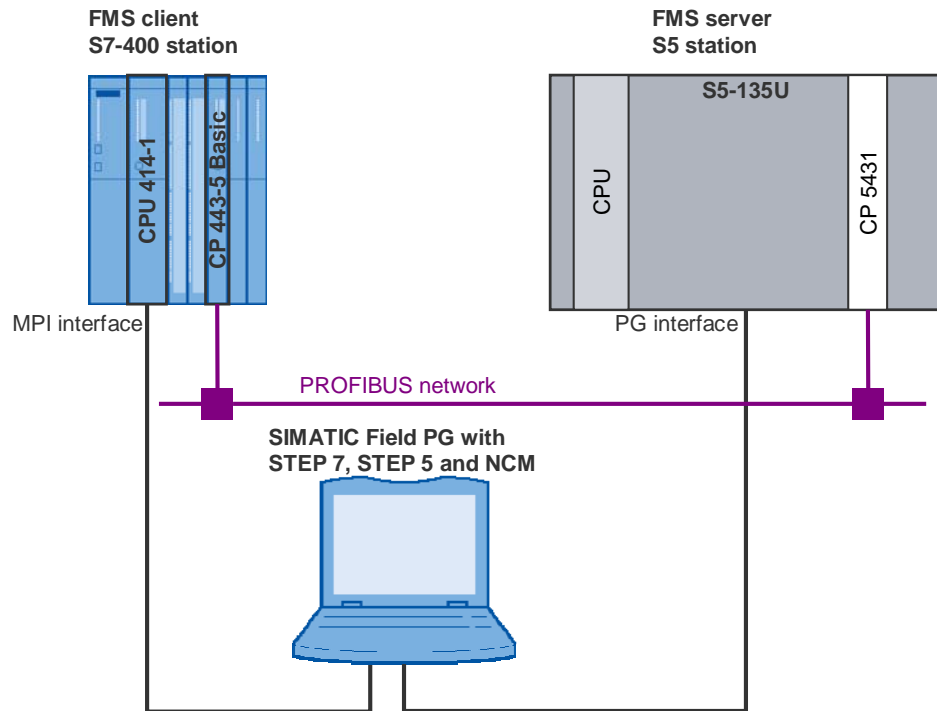
This document contains

- An overview of the plant configuration.
- An introduction to the blocks required and their communication structure.
- Details of the user programs of the S7-400 station and the S5 station.
- Information on how to use the sample programs.
- Other notes, tips and tricks, etc.

2 Overview of the Sample Program

Plant configuration

Figure 2-1



Checking the function of the sample program

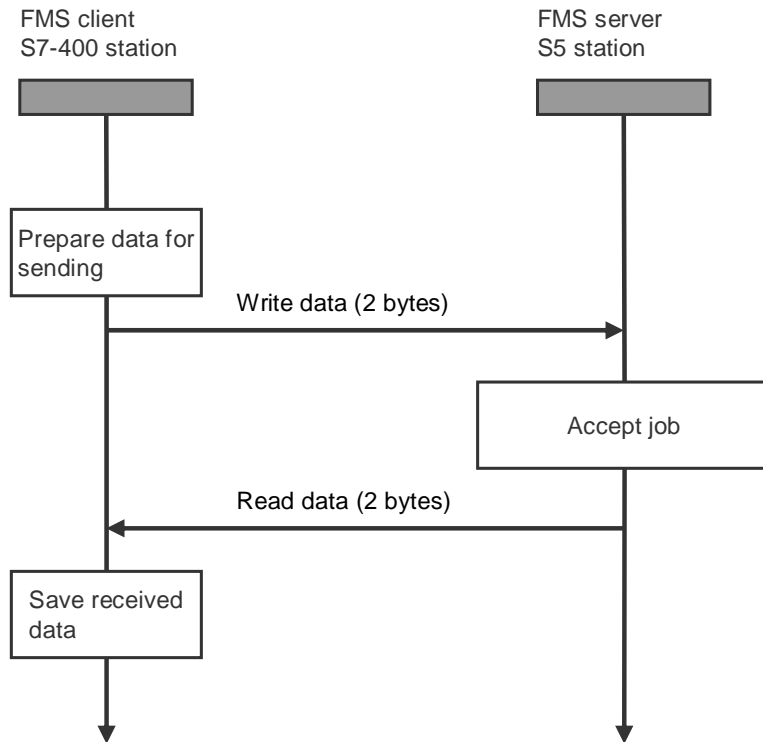
You can see whether the sample program is functioning correctly from variable table of the S7-400 station (FMS client). The variable table is included in the block folder of the STEP 7 project.

The data sent and received changes its value constantly (see chapter 6).

Overview of functions

The overview of functions shows the principal working method of the sample program.

Figure 2-2



Data communication

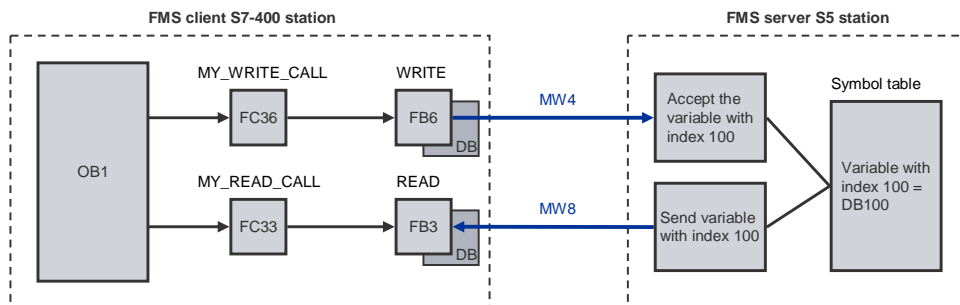
2 bytes of data are exchanged cyclically between the two stations involved in the communication.

FMS Server and FMS Client

The FMS client initiates reading and writing of the data. The FMS server reacts to the services of the FMS client; for example, it provides the data for the read job of the FMS client.

Program overview

Figure 2-3



User program of the S7-400 (FMS client)

The function block "B6 "WRITE" is called in the user program of the S7-400. The write job writes 2 bytes of data from the marker area of the FMS client (MW 4) to the marker area of the FMS server (MW 4).

Upon successful execution and termination of the write job the function block FB3 "READ" is called. The read job reads 2 bytes of data from the marker area of the FMS client (MW 4). The FMS client stores the data read in the marker area (MW 8).

When the read job has been completed successfully, then the data to be written is incremented by 1 and the send/read cycle starts anew.

User program of the S5 station (FMS server)

Only the function block for the communications processor is called in the user program of the S5 station. The data used for communication is defined in a symbol table. It is declared as a communications variable and assigned to an index.

DB5 "Index_DB" in the FMS client

It is not possible to specify the index or the name of the communications variable directly on FB3 "READ" and FB6 "WRITE". The communications variable is addressed over an ANY pointer at the input parameter VAR_1 of FB3 "READ" and FB6 "WRITE". For this reason, the data block DB5, which has the symbolic name of "Index_DB", is used as an auxiliary data block in this example.

The DB5 "Index_DB" is structured as follows:

Figure 2-4

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	Index100	STRING[10]	'<100>'	
=12.0		END_STRUCT		

Initialization of the FB6 "WRITE"

The function block FB6 "WRITE" must be initialized before it can be used for communication. To initialize the function block FB6 "WRITE" you call the block once with the input parameter REQ=0.

In this example initialization is done by calling the function FC36 "MY_WRITE_CALL" in the OB100 startup OB.

Error handling

If the function block FB3 "READ" or FB6 "WRITE" is not executed and terminated successfully, i.e. the output parameter ERROR=1, then you must first reset the function block FB3 "READ" or FB6 "WRITE" before the next write or read job can be initiated with REQ=1.

Proceed as follows to reset the function blocks FB3 "READ" and FB6 "WRITE":

- Call the function block with the input parameter REQ=0.
- Call the function block again with the input REQ=0 without changing the output parameter "STATUS".

Operating and monitoring

The operating and monitoring of the sample program is possible using the variable table provided.

The following requirements must be fulfilled for the "Monitor and modify variables" program.

- An online connection to the CPU must be established.
- The "Monitor variables" function must be activated.
- The control values must be marked as valid.

If the above-mentioned conditions are fulfilled, then you can tell that the sample program is working properly by the fact that the sent and received data changes constantly in the active partner.

3 Plant Configuration

This chapter gives you an overview of the configuration and the hardware and software components used to create the sample program.

3.1 Overview

Table 3-1 gives an overview of the configuration used to create the sample program.

Table 3-1

Bus system	PROFIBUS
Communication protocol	FMS
FMS Client	SIMATIC S7
FMS Server	SIMATIC S5
Communications processors	CP443-5 Basic and CP5431
Services	READ and WRITE

In this sample program the PROFIBUS FMS protocol is used with the services READ and WRITE for data communication between a SIMATIC S7 and a SIMATIC S5.

In the FMS client, the S7-400 station, a CP443-5 Basic communications processor is used for data communication using the FMS protocol and the READ / WRITE services.

In the FMS server, the S5 station, a CP5431 communications processor is used.

3.2 Hardware and Software Components Used

Hardware components

The following modules in the FMS client, the S7-400 station, were used to create the sample program.

Table 3-2

Slot	Module
1	PS 407 10A
3	CPU 414-1
5	CP443-5 Basic

The following modules in the FMS server, the S5 station, were used to create the sample program.

Table 3-3

Slot	Module
1	AS 135/155U (CPU 928)
2	CP5431 FMS

Note

The sample project has been created with a specific hardware configuration. This must be maintained to ensure proper functioning.

If you use a different configuration, with a different CPU or CP, for example, then you must change the sample program accordingly.

Required cables and other hardware

- MPI cable
- PG cable
- PROFIBUS bus cable with at least two male bus connectors
- SIMATIC Field PG or PC with MPI and AS interface

Software components

- STEP 7 V4.2 or higher
- NCM S7 PROFIBUS V4.2 or higher
- STEP 5 V6.65
- SINEC COM 5431 V4.09

4 Function Mechanisms of the Sample Program

This chapter describes how the sample program functions and which blocks are required for communication and their communication structure.

4.1 Working Method of the Sample Program

4.1.1 Data Communication

2 bytes of data are exchanged cyclically between the two stations involved in the communication.

4.1.2 FMS Client (S7-400 Station) and FMS Server (S5 Station)

The FMS client initiates reading and writing of the data.

The FMS server reacts to the services of the FMS client; for example, it provides the data for the read job of the FMS client.

4.1.3 Sequence of the User Program in the FMS Client (S7-400 Station)

Write data

When the function block FB6 "WRITE" is called, the FMS client writes the data in the marker area of the FMS server, in this example MW 4.

Read data

When the function block FB3 "READ" is called, the data of the FMS server is read. The FMS client stores the data read in the marker area MW 8.

Increase data

The value of the data to be written is increased by one and the write and read cycle starts over.

4.1.4 Sequence of the User Program in the FMS Server (S5 Station)

The function blocks FB120 "SEND_ALL" and FB121 "RECEIVE_ALL" are called in the user program of the FMS server to execute the received write and read jobs of the FMS client.

FB120 "SEND_ALL"

The FB120 "SEND_ALL" is executed when the read job is executed by the FB3 "READ" in the FMS client.

The variable with the index 100 of the FMS server is read. The variable refers to the address below in the S5 station: Word 0 in data block DB100.

FB121 "RECEIVE_ALL"

The FB121 "RECEIVE_ALL" is executed when the write job is executed by the FB6 "WRITE" in the FMS client.

The variable with the index 100 of the FMS server is written. The variable refers to the address below in the S5 station: Word 0 in data block DB100.

Note The FMS client writes and reads the Word 0 in data block DB100 of the FMS server.

4.2 Functions and Function Blocks

4.2.1 Functions and Function Blocks in the User Program of the S7-400 Station

The function blocks FB3 "READ" and FB6 "WRITE" for data communication are called in the user program of the S7-400 station.

Task of the function block FB6 "WRITE"

The function block FB6 "WRITE" sends the data from the defined local data area of the FMS client to the data area of the partner station, the FMS server.

Note Section 4.3.5 gives a description of the input and output parameters of the function block FB6 "WRITE".

Task of the function block FB3 "READ"

Depending on the task parameters the function block FB3 "READ" reads the data from the data area in the communication partner via which the name or index is specified. The data read is stored in a data block, in the process picture of the inputs and outputs or in the marker area.

Note Section 4.3.7 gives a description of the input and output parameters of the function block FB3 "READ".

Read out the variable descriptions of the communication partner

The structure of the variables is determined by the partner station, the FMS server. The variable descriptions are read from the communication partner when the FMS connection is established. The structure of the variables is then available in the PROFIBUS CP to convert the data from the S7 format into the FMS format.

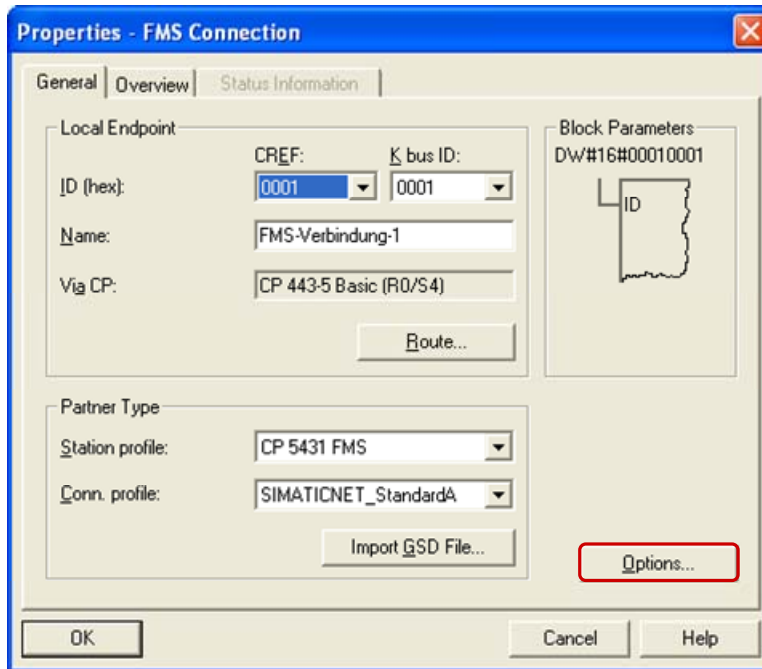
The variable descriptions are only read out if the connection is established using the "GET OD" service.

In NetPro you must activate one of the options below for the function "Read out the variable descriptions of the communication partner" when configuring the FMS connection.

- All
- Specified variables from the list below

In the Properties dialog of the FMS connection, click on the "Options..." button.

Figure 4-1



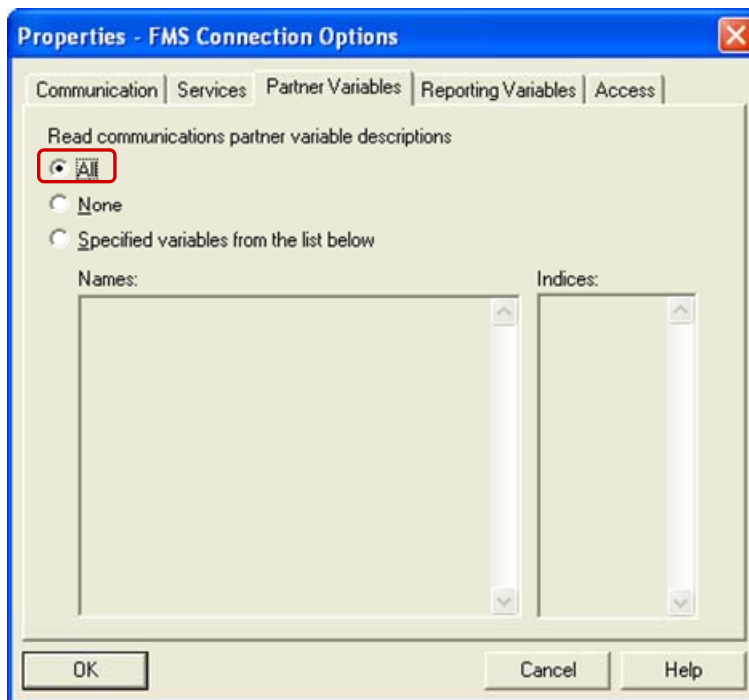
In the dialog "Properties – FMS Connection Options" you switch to the "Partner Variables" tab.

Activate one of the options below for the function "Read out the variable descriptions of the communication partner".

- All
- Specified variables from the list below

Apply the settings with "OK".

Figure 4-2

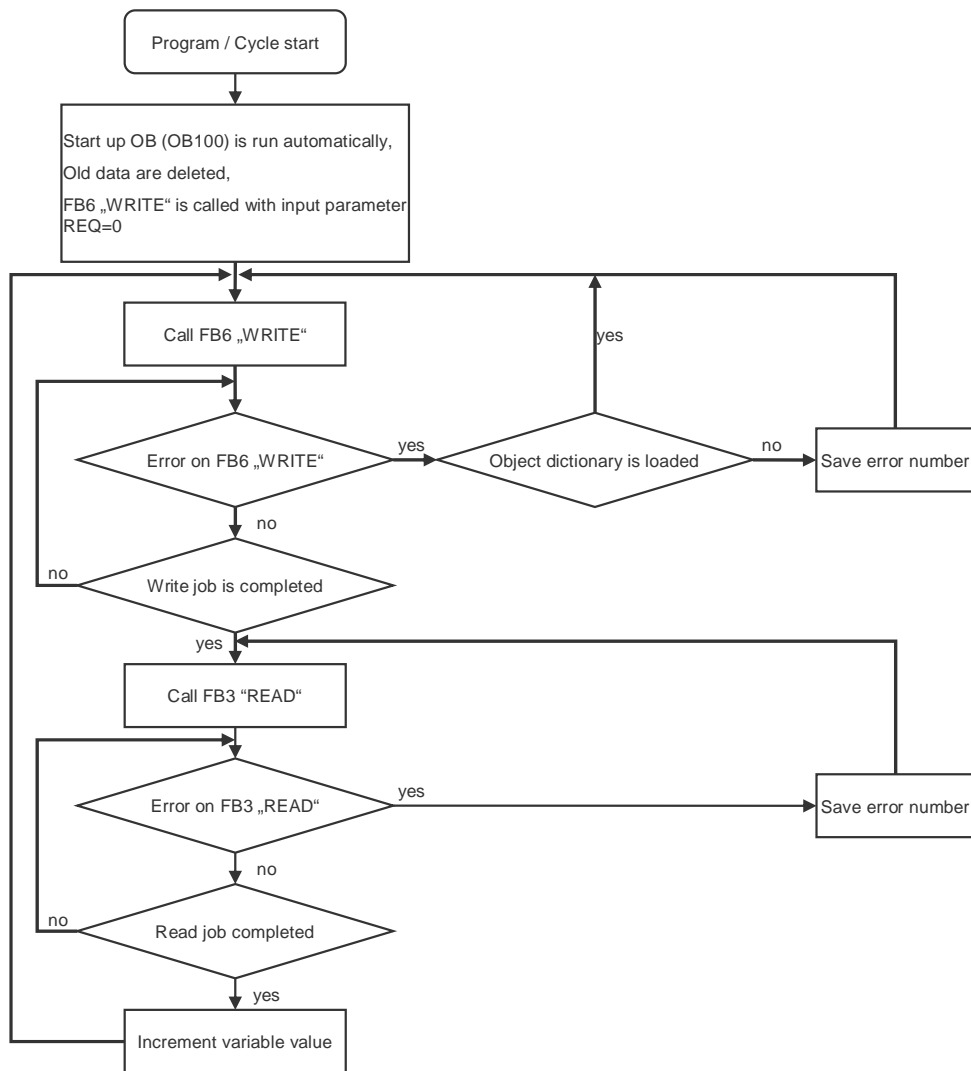


Note If you define access rights for the "READ" and "WRITE" services, then data transfer is only possible if the access rights defined are assigned to the FMS client.

4.3 Details of the User Program of the FMS Client (S7-400 Station)

4.3.1 Program Sequence

Figure 4-3



4.3.2 Symbol Table

In the symbol table you assign symbolic names to the addresses used in the user program of the S7-400 station.

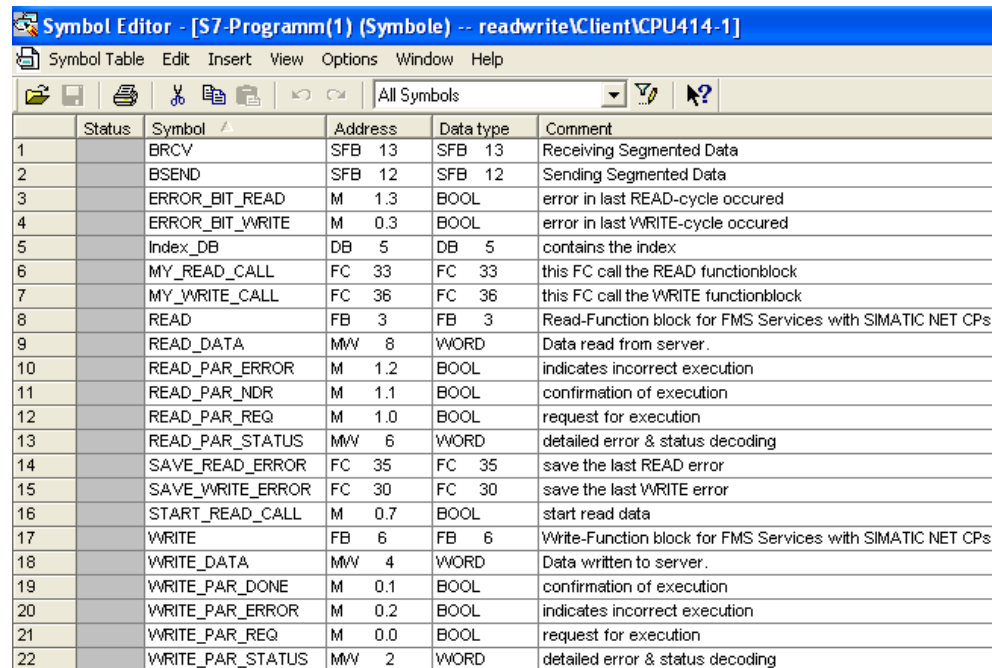
The use of symbols increases the readability of the program code. The symbolic name is displayed in the program code instead of the address.

Example

The symbolic name "READ_DATA" is in the program code instead of the address MW8.

Overview

Figure 4-4 shows the symbol table used in the user program of the S7-400 station.
Figure 4-4



	Status	Symbol	Address	Data type	Comment
1		BRCV	SFB 13	SFB 13	Receiving Segmented Data
2		BSEND	SFB 12	SFB 12	Sending Segmented Data
3		ERROR_BIT_READ	M 1.3	BOOL	error in last READ-cycle occurred
4		ERROR_BIT_WRITE	M 0.3	BOOL	error in last WRITE-cycle occurred
5		Index_DB	DB 5	DB 5	contains the index
6		MY_READ_CALL	FC 33	FC 33	this FC call the READ functionblock
7		MY_WRITE_CALL	FC 36	FC 36	this FC call the WRITE functionblock
8		READ	FB 3	FB 3	Read-Function block for FMS Services with SIMATIC NET CPs
9		READ_DATA	MW 8	WORD	Data read from server.
10		READ_PAR_ERROR	M 1.2	BOOL	indicates incorrect execution
11		READ_PAR_NDR	M 1.1	BOOL	confirmation of execution
12		READ_PAR_REQ	M 1.0	BOOL	request for execution
13		READ_PAR_STATUS	MW 6	WORD	detailed error & status decoding
14		SAVE_READ_ERROR	FC 35	FC 35	save the last READ error
15		SAVE_WRITE_ERROR	FC 30	FC 30	save the last WRITE error
16		START_READ_CALL	M 0.7	BOOL	start read data
17		WRITE	FB 6	FB 6	Write-Function block for FMS Services with SIMATIC NET CPs
18		WRITE_DATA	MW 4	WORD	Data written to server.
19		WRITE_PAR_DONE	M 0.1	BOOL	confirmation of execution
20		WRITE_PAR_ERROR	M 0.2	BOOL	indicates incorrect execution
21		WRITE_PAR_REQ	M 0.0	BOOL	request for execution
22		WRITE_PAR_STATUS	MW 2	WORD	detailed error & status decoding

4.3.3 OB100

The OB100 is a startup OB.

Overview

Figure 4-5

```

OB100 : Startup
CP Startup
Network 1: Title:
Signal edge change to execute WRITE
Bit memory and data block area delete

L    0
T    DB2.DBW    0          //DB2, error data block area delete
T    DB2.DBW    2

T    MW        0          //Bit memory delete
T    "WRITE_PAR_STATUS"
T    "WRITE_DATA"
T    "READ_PAR_STATUS"
//executing WRITE once with WRITE_PAR_REQ = 0 so that our CP is ready for communication:
CALL "MY_WRITE_CALL"
SET  "MY_WRITE_CALL"
S    "WRITE_PAR_REQ"          //signal edge change, REQ: 0 -> 1 WRITE

```

Description

The data area used is preset with "0" with load and transfer commands.

The function FC36 "MY_WRITE_CALL" is called in OB100. At this point in time the input parameter REQ of FB6 "WRITE" has the value "0".

The write job is then triggered. The value at the input parameter REQ of FB6 "WRITE" is set to "1".

The write job is executed the next time the function FC36 "MY_WRITE_CALL" is called.

4.3.4 OB1

OB1 is the block responsible for cyclic processing of the user program. The function FC36 "MY_WRITE_CALL" is called cyclically in OB1. The function "MY_READ_CALL" is only called if the "START_READ_CALL" variable has the value "1", otherwise OB1 is terminated immediately.

The "START_READ_CALL" is set to "1" only when previously started write job is completed successfully.

Figure 4-6

```

OB1 : cycle

WRITE/READ

Network 1: Title:
write/read variable INT (IN16) with index 100

CALL "MY_WRITE_CALL"           //write INT16
AN  "START_READ_CALL"
BEC
CALL "MY_READ_CALL"           //read INT16
    
```

4.3.5 FC36 "MY_WRITE_CALL"

Overview

Figure 4-7

```

FC36 : write
The WRITE FB transfers data from from a specified local data (SD_1, MW4) to a
data area (MW4) on the communications partner

Network 1: Title:
The data area of the communications partner is specified using a variable index
100

zr: CALL "WRITE" , DB6           //write block call with instance DB
REQ  := "WRITE_PAR_REQ"         //Signal edge change to execute the FB
ID   := DW#16#10001             //Connection 10001: compared with configuration of FMS connection
VAR_1 := "Index_DB".Index100    //addresses variable to be read, "Index_DB"=DB5
SD_1 := P#M 4.0 WORD 1         //addresses source data area (MW4)
DONE  := "WRITE_PAR_DONE"       //confirmation of execution
ERROR := "WRITE_PAR_ERROR"      //indicates incorrect execution
STATUS := "WRITE_PAR_STATUS"    //detailed error decoding

//question about error
A "ERROR_BIT_WRITE"
JC erro
AN "WRITE_PAR_ERROR"
JC nerr

S "ERROR_BIT_WRITE"           //"error bit"
R "WRITE_PAR_REQ"
JU zr

// error occurred
erro: SET
R "ERROR_BIT_WRITE"
CALL "SAVE_WRITE_ERROR"
BEU

//no error
nerr: AN "WRITE_PAR_DONE"       //reception of data completed?
JC ndon

//reception of data completed
S "READ_PAR_REQ"             //edge signal executing READ
S "START_READ_CALL"
BEU

//reception of data not completed
ndon: A "WRITE_PAR_REQ"
R "WRITE_PAR_REQ"
    
```

Description

The function FC36 "MY_WRITE_CALL" is called cyclically in OB1. It internally calls the function block FC6 "WRITE". The write job writes 2 bytes of data to the FMS server.

The write job is called only when the previous write job and associated read job have been completed successfully, i.e. the user program checks whether the newly started write job has been executed and completed successfully. The read job is executed only when the newly started write job has been executed and completed successfully. Otherwise the user program waits until the write job has been

completed successfully or calls the function FC30 "SAVE_WRITE_ERROR" for error handling.

FB6 "WRITE" has the following input parameters.

Table 4-1

Input parameters	Data type	Description
REQ	BOOL	Edge signal for executing the block
ID	DWORD	This ID identifies the FMS connection. Take the ID from the connection dialog.
VAR_1	ANY	The parameter addresses the remote communication variable that is to be read or written. In this example the communication variable is address over the index 100.
SD_1	ANY	Address of a local data area from which variables are to be transferred.

FB6 "WRITE" has the following output parameters.

Table 4-2

Output parameters	Data type	Description
DONE	BOOL	Displays the processing of the job.
ERROR	BOOL	Indicates whether an error has occurred.
STATUS	WORD	Status value Provides detailed information about warnings or errors after processing of the job.

Error evaluation

The value of the output parameter ERROR is evaluated to determine whether the write job has been executed successfully.

If the write job has not been executed successfully, the output parameter ERROR of function block FB6 "WRITE" has the value "1". In this case the marker bit M0.3 "ERROR_BIT_WRITE" is set to the value "1" and the input parameter REQ of function block FB6 "WRITE" is reset to the value "0".

The next time the function FC36 "MY_WRITE_CALL" is called no write job is executed because the function block FB6 "WRITE" is called with the input parameter REQ=0.

Marker bit M0.3 "ERROR_BIT_WRITE" is reset again because the function FC30 "SAVE_WRITE_ERROR" is called and the function FC36 "MY_WRITE_CALL" is quit by way of an absolute block end.

Processing of the write job completed

If the write job has been completed successfully, the value of the output parameter DONE is evaluated to determine whether the write job has been completed or is still running.

The write job is completed once the DONE output parameter has the value "1". In this case the marker bit M0.7 "START_READ_CALL" and the input parameter REQ of function block FB3 "READ" are set to the value "1". This calls the function FC33 "MY_READ_CALL" and a read job is executed.

Processing of the write job is running

The write job is running if the DONE output parameter has the value "0". In this case the input parameter REQ of function block FB6 "WRITE" is reset to the value "0" so that no new write job can be executed.

Since the function "MY_WRITE_CALL" is called cyclically in OB1 a check is made in each block cycle as to whether the write job has been completed or is still running.

4.3.6 FC30 "SAVE_WRITE_ERROR"

If an error occurs when calling the function block FB6 "WRITE", i.e. the write job has not been executed and completed successfully, then

- The write job is repeated.
- The status value of the function block FB6 "WRITE" is stored.

Note

As long as the CP in the FMS client reads the object directory the status value of function block FB6 "WRITE" is not stored and the write job is repeated.

Overview

Figure 4-8

```

FC30 : write error store

When the FB is called while the PROFIBUS CP is starting up (for example due to
turning the power off and on again or due to a power outage) the following
parameter is possible: STATUS=0001 (K bus connections is not yet established)
or 0607 (GetOD still active)

Network 1: Title:
error occurred

L    "WRITE_PAR_STATUS"
L    W#16#601
<>I                               //GetOD still active?
JC   god                           //GetOD no active

//GetOD still active, variable non existent
SET
S    "WRITE_PAR_REQ"               //write,Signal edge change "0 -> 1"
BEU

//GetOD completed, variable exist
god: L    "WRITE_PAR_STATUS"
T    DB2.DEW    0

S    "WRITE_PAR_REQ"
BE
    
```

Description

The function FC30 "SAVE_WRITE_ERROR" is called in FC36 "MY_WRITE_CALL" if the previous write job has not been completed successfully.

The function FC30 "SAVE_WRITE_ERROR" checks the value returned at the output parameter of the function block FB6 "WRITE".

The value 0601(hex) at the output parameter STATUS of FB6 "WRITE" means that the CP in the FMS client is still loading the object directory from the FMS server

and therefore no write job can be executed. In this case the input parameter REQ of FB6 "WRITE" is set to the value "1" so that write job is repeated.

If the value at the output parameter STATUS of FB6 "WRITE" is not 0(hex) or 0601(hex), then the status value is stored in data word 0 of data block DB2. Finally, the input parameter REQ of FB6 "WRITE" is set to the value "1" so that write job is repeated.

4.3.7 FC33 "MY_READ_CALL"

Overview

Figure 4-9

FC33 : read

The READ FB reads data from area of the communication partner (MW4) specified by index 100

Network 1: error in last READ-cycle occurred

The data that are read are saved locally in a bit memory area (MW4)

```

rs:  CALL "READ" , DB3           //READ block call with instance DB
      REQ  := "READ_PAR_REQ"     //Signal edge change to execute the FB
      ID   := "DW#16#10001"     //Connection 10100: compared with configuration of FMS connection
      VAR_1 := "Index_DB".Index100 //addresses variable that will be read, "Index_DB"=DB5
      RD_1  := "PMM 8.0 WORD 1" //addresses data area as destination (MW4)
      NDR   := "READ_PAR_NDR"   //confirmation of execution
      ERROR := "READ_PAR_ERROR" //indicates incorrect execution
      STATUS := "READ_PAR_STATUS" //detailed error decoding

//question about error
A    "ERROR_BIT_READ"
JC   erro
AN   "READ_PAR_ERROR"
JC   nerr
S    "ERROR_BIT_READ"         //set "error bit"
R    "READ_PAR_REQ"
JU   rs

//error occurred
erro: SET
R    "ERROR_BIT_READ"
CALL "SAVE_READ_ERROR"
BEU

//no error
nerr: AN "READ_PAR_NDR"       //reception of data completed?
      JC ndr

//reception of data completed
SET
S    "WRITE_PAR_REQ"         //edge signal executing WRITE
R    "START_READ_CALL"
L    "WRITE_DATA"           //data + 1
+    1
T    "WRITE_DATA"

//reception of data not completed
ndr: A "READ_PAR_REQ"
      R "READ_PAR_REQ"
    
```

Description

The function FC333 "MY_READ_CALL" is called cyclically in OB1 if the previous write job has been completed successfully. It internally calls the function block FB3 "READ". The read job reads 2 bytes of data from the FMS server.

The read job is called only when the previous read job and associated write job have been completed successfully. When the read job is called, the user program checks whether the active read job has been completed successfully and that the data has been received in full. If the read job has been completed successfully and the data has been received in full, then the write job is executed. Otherwise the user program waits until the read job has been completed successfully or calls the function FC35 "SAVE_READ_ERROR" for error handling.

FB3 "READ" has the following input parameters.

Table 4-3

Input parameters	Data type	Description
REQ	BOOL	Edge signal for executing the block
ID	DWORD	This ID identifies the FMS connection. Take the ID from the connection dialog.
VAR_1	ANY	The parameter addresses the remote communication variable that is to be read or written. In this example the communication variable is address over the index 100.
RD_1	ANY	Address of a local data area to which variables are to be transferred.

FB3 "READ" has the following output parameters.

Table 4-4

Output parameters	Data type	Description
NDR	BOOL	Indicates receipt of data.
ERROR	BOOL	Indicates whether an error has occurred.
STATUS	WORD	Status value Provides detailed information about warnings or errors after processing of the job.

Error evaluation

The value of the output parameter ERROR is evaluated to determine whether the read job has been executed successfully.

If the read job has not been executed successfully, the output parameter ERROR of function block FB3 "READ" has the value "1". In this case the marker bit M1.3 "ERROR_BIT_READ" is set to the value "1" and the input parameter REQ of function block FB3 "READ" is reset to the value "0".

The next time the function FC33 "MY_READ_CALL" is called no read job is executed because the function block FB3 "READ" is called with the input parameter REQ=0.

Marker bit M1.3 "ERROR_BIT_READ" is reset again because the function FC35 "SAVE_READ_ERROR" is called and the function FC33 "MY_READ_CALL" is quit by way of an absolute block end.

Processing of the read job completed

If the read job has been completed successfully, the value of the output parameter NDR is evaluated to determine whether the read job has been completed or is still running.

The read job is completed and the data received in full once the NDR output parameter has the value "1". In this case the marker bit M0.7 "START_READ_CALL" and the input parameter REQ of function block FB6 "WRITE" are set to the value "1" so that a new write job can be executed. The value of the send data is increased by "1".

Processing of the read job is running

The read job is running if the NDR output parameter has the value "0". In this case the input parameter REQ of function block FB3 "READ" is reset to the value "0" so that no new read job can be executed.

Since the function "MY_READ_CALL" is called cyclically in OB1 a check is made in each block cycle as to whether the read job has been completed or is still running.

4.3.8 FC35 "SAVE_READ_ERROR"

If an error occurs when calling the function block FB3 "READ", i.e. the read job has not been executed and completed successfully, then

- The read job is repeated.
- The status value of the function block FB3 "READ" is stored.

Overview

Figure 4-10

```

FC35 : read error store

When the FB is called while the PROFIBUS CP is starting up (for example due to
turning the power off and on again or due to a power outage) the following
parameter is possible: STATUS=0001 (K bus connections is not yet established)
or 0607 (GetOD still active)

Netzwerk 1: Titel:
error occurred

L   "READ_PAR_STATUS"
T   DB2.DEW      2
S   "READ_PAR_REQ"
BE
    
```

Description

The function FC35 "SAVE_READ_ERROR" is called in FC33 "MY_READ_CALL" if the previous read job has not been completed successfully.

The function FC35 "SAVE_READ_ERROR" saves the value returned at the output parameter of the function block FB3 "READ" in data word 2 of data block DB2. Finally, the input parameter REQ of FB3 "READ" is set to the value "1" so that write job is repeated.

4.3.9 DB5 "Index_DB"

In data block DB5 "Index_DB" you define the index via which the communication variable to be read and written is addressed. Definition of the communication variable and assignment of the Index 100 are done in the FMS server.

Note

Each communication variable is assigned to an index. The index uniquely identifies the communication variable. The communication variable and assignment of the index are defined in the FMS server.

Overview

Figure 4-11

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	Index100	STRING[10]	'<100>'	
=12.0		END_STRUCT		

Description

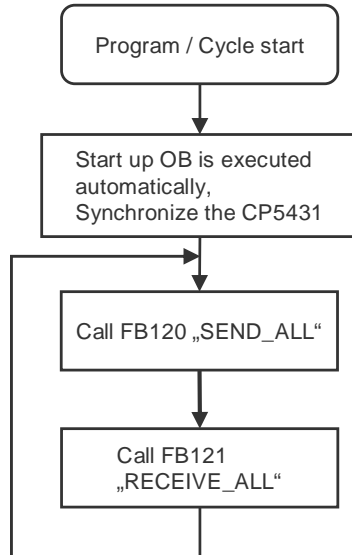
The variable "Index100" of data type "STRING" is created in DB5 "Index_DB". The variable has the value "<100>". This designates the FMS Index.

Function blocks FB6 "WRITE" and FB3 "READ" access the value of the variable "Index_100" in DB5 "Index_DB" by way of the input parameter "VAR_1". This means that the DB5 "Index_DB" is transferred as value of the input parameter "VAR_1" at FB6 "WRITE" and FB3 "READ". In this way the function blocks FB3 "READ" and FB6 "WRITE" can address the communication variable in the FMS server.

4.4 Details of the User Program of the FMS Server (S5 Station)

4.4.1 Program Sequence

Figure 4-12



4.4.2 Configure Communication Variable in the FMS Server

The configuration software SINEC COM 5431 is used to define the communication variable and assign the index for the FMS server.

In this example a communication variable of the data type Integer (IN 16) is defined with the Index 100. The communication variable corresponds to the S5 address Word 0 in data block DB100.

Figure 4-13

Index	Typ	ZGRF	Passw	S5-Adresse	ANZW	SSNR
100	IN 16			DB 100 0	W 0	0

4.4.3 OB20

The OB20 calls the function block FB125 with the relevant parameter values. The FB125 is responsible for synchronizing the CP5431.

Overview

Figure 4-14

```
:JU FB 125  
NAME:SYNCHRON  
SSNR:KY 0,0  
BLGR:KY 0,0  
PAFE:FY 255  
:BE
```

4.4.4 OB1

OB1 is responsible for cyclic processing of the user program and call function block FB231 cyclically.

Overview

Figure 4-15

```
:JU FB 125  
NAME:SYNCHRON  
SSNR:KY 0,0  
BLGR:KY 0,0  
PAFE:FY 255  
:BE
```

4.4.5 FB231

The FB231 calls the function blocks FB120 "SEND_ALL" and FB121 "RECEIVE_ALL" with the relevant parameter values.

Function block FB120 "SEND_ALL" sends the data requested by the FMS client with FB3 "READ".

Function block FB121 "RECEIVE_ALL" receives the data written by the FMS client with FB6 "WRITE".

FB120 "SEND_ALL"

Figure 4-16

```
      :O   M   0.0
      :ON  M   0.0
      :SPA FB 120
Name  :SEND
SSNR >   KY 0,0
A-NR :    KY 0,0
ANZW :    MW 80
QTYP :    KC NN
DBNR :    KY 0,0
QANF :    KF +0
QLAE :    KF +0
PAFE :    MB 86
```

FB121 "RECEIVE_ALL"

Figure 4-17

```
      :O   M   0.0
      :ON  M   0.0
      :SPA FB 120
Name  :SEND
SSNR >   KY 0,0
A-NR :    KY 0,0
ANZW :    MW 80
QTYP :    KC NN
DBNR :    KY 0,0
QANF :    KF +0
QLAE :    KF +0
PAFE :    MB 86
```

5 Installation and Commissioning

5.1 Installation of the Sample Program of the S7-400 Station

The various sample programs for the S7-400 station are available for downloading as a ZIP file. Extract the relevant file, for example "RW_S7S5_FMS_CP443-5_CP5431_V20.zip", in a separate directory.

The STEP 7 project that contains the Hardware Configuration and the user program of the S7-400 is then unpacked automatically with all the associated subdirectories. You can then use the SIMATIC Manager to open and process the extracted STEP 7 project.

5.2 Installation of the Sample Program of the S5 Station

The various sample programs for the S5 station are available for downloading as a ZIP file.

Extract the relevant file, for example "RW_S7S5_FMS_CP5431_CP443-5_V20.zip", in a separate directory.

The STEP 5 project that contains the configuration and the user program of the S5 station is then unpacked automatically with all the associated subdirectories. You can then open and process the extracted STEP 5 project.

5.3 Installation of the Sample Program for the PC with CP5412 A2

FMS client sample programs for the PC with CP5412 A2 are available for downloading as a ZIP file.

Extract the relevant file, for example "RW_PCS5_FMS_CP5412_CP5431_V20.zip", in a separate directory.

The sample program for the FMS client is unpacked automatically with the subdirectories below:

- code
- database
- exe

5.4 Commissioning

Keep to the order below for commissioning the sample program.

Table 5-1

No.	
1.	Start the FMS server.
2.	Switch to the "exe" subdirectory of the unpacked sample program for FMS client.
3.	Double-click the ".exe" file.
4.	Start the FMS client with the menu <code>Write/Read</code> → <code>Start</code> .

Note

You must start the FMS server before the FMS client, because the FMS server provides the services for the FMS client.

6 How to Use the Sample Program

You use the variable table supplied with the user program of the S7-400 to operate the sample program.

Opening the variable table in the user program of the S7-400

Table 6-1

No.	Action
5.	Use the MPI cable to connect the SIMATIC Field PG to the MPI/DP interface of the S7-400 CPU.
6.	Open the SIMATIC Manager and navigate in the project window to the block folder of the SIMATIC 400 station.
7.	In the block folder you double-click the variable table "VAT1".
8.	In the variable table you establish a connection to the S7-400 by means of the menu <code>Target system</code> → <code>Establish connection to</code> → <code>Configured CPU</code> .
9.	In the variable table you select the menu <code>Variable</code> → <code>Monitor</code> to control and monitor the defined variables.

Variable table

Figure 6-1

VAT1 -- readwriteClient\CPU414-1S7-Programm(1)					
	Address	Symbol	Display format	Status value	Modify value
1		//WRITE-Parameter:			
2		//REQ:			
3	M 0.0	"WRITE_PAR_REQ"	BIN		
4		//DONE:			
5	M 0.1	"WRITE_PAR_DONE"	BIN		
6		//ERROR:			
7	M 0.2	"WRITE_PAR_ERROR"	BIN		
8		//STATUS:			
9	MW 2	"WRITE_PAR_STATUS"	HEX		
10		//Last WRITE error:			
11	DB2.DBW 0		HEX		
12					
13					
14		//READ-Parameter:			
15		//REQ:			
16	M 1.0	"READ_PAR_REQ"	BIN		
17		//NDR:			
18	M 1.1	"READ_PAR_NDR"	BIN		
19		//ERROR:			
20	M 1.2	"READ_PAR_ERROR"	BIN		
21		//STATUS:			
22	MW 6	"READ_PAR_STATUS"	HEX		
23		//Last READ error:			
24	DB2.DBW 2		HEX		
25					
26					
27		//DATA:			
28		//Written data to server:			
29	MW 4	"WRITE_DATA"	DEC		
30		//Last read data from server:			
31	MW 8	"READ_DATA"	DEC		
32					

In the variable table you can monitor the values of the output parameters of the function blocks FB6 "WRITE" and FB3 "READ".

Data is successfully sent and received

If the data is successfully sent and received, the output parameters of the functions FB6 "WRITE" and FB3 "READ" have the values below.

Table 6-2

Output parameters	Addresses	Symbolic name	Value
DONE	M0.1	WRITE_PAR_DONE	The value of marker bit M0.1 changes constantly between 0 and 1.
ERROR	M0.2	WRITE_PAR_ERROR	M0.2 = 0
STATUS	MW2	WRITE_PAR_STATUS	MW2 = 0
NDR	M1.1	READ_PAR_NDR	The value of marker bit M1.1 changes constantly between 0 and 1.
ERROR	M1.2	READ_PAR_ERROR	M1.2 = 0
STATUS	MW6	READ_PAR_STATUS	MW6 = 0

The send data is stored in MW4 "WRITE_DATA".

The receive data is stored in MW8 "READ_DATA".

When the data has been sent and received successfully the send and receive data is increased constantly by the value "1".

Errors in sending and receiving the data

If the data is **not** sent successfully, the output parameters of the functions FB6 "WRITE" and FB3 "READ" have the values below.

Table 6-3

Output parameters	Addresses	Symbolic name	Value
DONE	M0.1	WRITE_PAR_DONE	M0.1 = 0
ERROR	M0.2	WRITE_PAR_ERROR	M0.2 = 1
STATUS	MW2	WRITE_PAR_STATUS	MW2 <> 0
NDR	M1.1	READ_PAR_NDR	M1.1 = 0
ERROR	M1.2	READ_PAR_ERROR	M1.2 = 1
STATUS	MW6	READ_PAR_STATUS	MW6 <> 0

The status value of the function block FB6 "WRITE" is stored in data word 0 of data block DB2.

The status value of the function block FB3 "READ" is stored in data word 2 of data block DB2.

Send data

Proceed as follows for sending data to the FMS server.

Table 6-4

No.	Action
1.	In the variable table you enter a control value of between 0 and 32767 for the marker word MW4.
2.	The control value entered is applied for the marker word MW4 by means of the menu Variable → Activate control values.

If the data is transferred successfully, the FMS client reads back the value from the FMS server. The FMS client stores the value read in MW8. The value read and stored in MW8 differs from the written value by the value of "1".

Closing the variable table in the user program of the S7-400

Close the variable table by means of the menu `Table` → `Close`.

7 Other Notes, Tips and Tricks, etc.

This chapter gives you more notes, tips and tricks for the commissioning and working of the sample program.

7.1 Notes, Tips and Tricks for the Commissioning and Working of the Sample Program in the FMS Client

7.1.1 Not Possible to Download Hardware Configuration, Connection or Blocks to the S7-400 CPU

If it is not possible to download the hardware configuration, connection or blocks to the CPU, then make the following checks and take the following measures to remedy the situation.

Table 7-1

Check	Remedy
<p>Check that the MPI cable is connected on the SIMATIC Field PG and on the MPI or MPI/DP interface of the S7-400 CPU.</p> <p>If the MPI cable is not connected on the SIMATIC Field PG and on the MPI or MPI/DP interface of the S7-400 CPU, no connection can be established between the two stations.</p>	<p>Connect the MPI cable to the RS485 interface of the SIMATIC Field PG and to the MPI or MPI/DP interface of the S7-400 CPU.</p>
<p>Check the settings in "Set PG/PC interface...".</p> <p>The interface parameters used must be assigned to the access point of the application "S7ONLINE (STEP 7)".</p> <p>If the settings in "Set PG/PC interface..." are not correct, no connection can be established between the two stations.</p>	<p>Go to <i>Start</i> → <i>SIMATIC</i> → <i>STEP 7</i> and select the menu item "Set PG/PC interface".</p> <p>Under "Interface parameters used" you select the module to which the MPI cable on the SIMATIC Field PG is connected, CP5611 (MPI), for example.</p> <p>Apply the settings with "OK".</p>
<p>Open the hardware configuration. The hardware configuration in the STEP 7 project must match the configuration of the S7-400 station.</p>	<p>Change the hardware configuration in the STEP 7 project to match the configuration of the S7-400 station.</p>
<p>Check that the CPU has been completely reset.</p>	<p>If the CPU has not been completely reset, then do a complete reset of the S7-400 CPU.</p>

7.1.2 Values of the Send and Receive Data Do Not Change in the Variable Table

If the values of the send and receive data do not change in the variable table, then make the following checks and take the following measures to remedy the situation.

Table 7-2

Check	Remedy
Check whether the PROFIBUS bus cable is connected on the CP443-5 in the S7-400 station and on CP5431 in the S5 station. If the PROFIBUS cable is not connected on CP443-5 in the S7-400 station and on CP5431 in the S5 station, no connection can be established between the two stations. Data exchange is not possible.	Connect the PROFIBUS bus cable to the RS485 interface of the CP443-5 and the CP5431. Switch on the terminator of the PROFIBUS connector.
Check whether "Monitor variables" function is activated in the variable table.	Activate the "Monitor variable" function by means of the menu Variable → Monitor.

7.1.3 Control Values are Not Applied for the Variables in the Variable Table

If the control values for the variables in the variable table are not applied, then make the following checks and take the following measures to remedy the situation.

Table 7-3

Check	Remedy
Check whether the control value of the marker word MW4 is in the range of valid values.	Enter a control value for the marker word MW4 from the valid range of values of between 0 and 32767.

Note

If these notes, tips and tricks etc. for the commissioning and working of the sample program have not been of assistance, then repeat the commissioning of the sample program.

7.2 Notes, Tips and Tricks for the Commissioning and Working of the Sample Program in the FMS Server

7.2.1 "PLC Timeout" Message Appears

If the "PLC Timeout" message appears, make the following checks and take the following measures to remedy the situation.

Table 7-4

Check	Remedy
Check that the PG cable is connected to the PG interface of the CPU in the S5 station and to the PG or PC. If the PG cable is not connected to the PG interface of the CPU in the S5 station and to the PG or PC, no connection can be established between the two stations.	Connect the PG cable to the PG interface of the CPU in the S5 station and to the PG or PC. Acknowledge the "PLC Timeout" message with "OK". In the "Settings" dialog you use the arrow keys to select the "Mode" item. Press the <F3> key to switch to "Online" mode. Press the <F8> key to switch to the main menu of STEP 5.
Check whether the S5 station is switched	Switch on the S5 station.

Check	Remedy
<p>on.</p> <p>If the S5 station is not switched on, no connection can be established between the S5 station and the PG or PC.</p>	<p>Acknowledge the "PLC Timeout" message with "OK".</p> <p>In the "Settings" dialog you use the arrow keys to select the "Mode" item.</p> <p>Press the <F3> key to switch to "Online" mode.</p> <p>Press the <F8> key to switch to the main menu of STEP 5.</p>

7.2.2 Message "PLC interface on PG not ready"

If the "PLC interface on PG not ready" message appears, make the following checks and take the following measures to remedy the situation.

Table 7-5

Check	Remedy
<p>Check that the PG cable is connected to the PG interface of the CPU in the S5 station and to the PG or PC.</p> <p>If the PG cable is not connected to the PG interface of the CPU in the S5 station and to the PG or PC, no connection can be established between the two stations.</p> <p>The PG cable might be connected to the PG interface of the CP5431 instead of to the PG interface of the CPU.</p>	<p>Connect the PG cable to the PG interface of the CPU in the S5 station and to the PG or PC.</p> <p>Acknowledge the "PLC interface on PG not ready" message with "OK".</p> <p>Repeat the steps you made until the "PLC interface on PG not ready" message appeared.</p>

7.2.3 Message "CP interface not ready" in the Configuration Software SINEC COM 5431

If the "CP interface not ready" message appears in the configuration software SINEC COM 5431, make the following checks and take the following measures to remedy the situation.

Table 7-6

Check	Remedy
<p>Check that the PG cable is connected to the PG interface of the CP5431 in the S5 station and to the PG or PC.</p> <p>If the PG cable is not connected to the PG interface of the CP5431 and to the PG or PC, no connection can be established between the CP5431 and the PG or PC.</p>	<p>Connect the PG cable to the PG interface of the CP5431 in the S5 station and to the PG or PC.</p> <p>Acknowledge the "CP interface not ready" message with "OK".</p> <p>Repeat the steps you made until the "CP interface not ready" message appeared.</p>
<p>Check whether the S5 station is switched on.</p> <p>If the S5 station is not switched on, no connection can be established between the CP5431 and the PG or PC.</p>	<p>Switch on the S5 station.</p> <p>Acknowledge the "CP interface not ready" message with "OK".</p> <p>Repeat the steps you made until the "CP interface not ready" message appeared.</p>

Note

If these notes, tips and tricks etc. for the commissioning and working of the sample program have not been of assistance, then repeat the commissioning of the sample program.

