SIMATIC NET

Functions (FC) and function blocks (FB) for SIMATIC NET S7 CPs

Programming Manual

03/2008

C79000-G8976-C229-01

Safety Guidelines

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

⚠DANGER
indicates that death or severe personal injury **will** result if proper precautions are not taken.

⚠WARNING
indicates that death or severe personal injury **may** result if proper precautions are not taken.

⚠CAUTION
with a safety alert symbol, indicates that minor personal injury can result if proper precautions are not taken.

CAUTION
without a safety alert symbol, indicates that property damage can result if proper precautions are not taken.

NOTICE
indicates that an unintended result or situation can occur if the corresponding information is not taken into account.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

Qualified Personnel

The device/system may only be set up and used in conjunction with this documentation. Commissioning and operation of a device/system may only be performed by **qualified personnel**. Within the context of the safety notes in this documentation qualified persons are defined as persons who are authorized to commission, ground and label devices, systems and circuits in accordance with established safety practices and standards.

Prescribed Usage

Note the following:

⚠WARNING
This device may only be used for the applications described in the catalog or the technical description and only in connection with devices or components from other manufacturers which have been approved or recommended by Siemens. Correct, reliable operation of the product requires proper transport, storage, positioning and assembly as well as careful operation and maintenance.

Trademarks

All names identified by ® are registered trademarks of the Siemens AG. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

Siemens AG
Automation and Drives
Postfach 48 48
90327 NÜRNBERG
GERMANY

C79000-G8976-C229-01
℗ 05/2008

Preface

Preface

Target group and motivation

The interface to the communication services is implemented by readymade program blocks (FCs and FBs). This manual provides you with a full description of the FCs and FBs for SIMATIC NET S7 CPs. It extends the descriptions in the online help of the configuration tools for STEP 7.

The manual is intended for programmers of STEP 7 programs and service personnel.

Structure of the manual

The manual is structured according to network types and communications services.

Each function and each function block is described in the following sections:

- Meaning
- Call interface
- Operating principle
- Explanation of the formal parameters
- Condition codes

These sections may included further specific information.

Scope of this manual

This version of the manual is valid as of version V5.4 SP4 of the STEP 7 / NCM S7 configuration software.

New in this release

This manual groups together the block descriptions that were previously in the manuals for S7 CPs structured according to network types. These manuals will no longer include the block descriptions.

Compared with the previously valid versions of the manuals for S7 CPs, this version 1 includes the following innovations in the block descriptions:

- In the section on Industrial Ethernet

  A new function block FB 40 is available for the FTP client mode of advanced CPs. Using this FB, complete FTP job sequences can be created efficiently in the user program.

- In the section on PROFINET IO

Parameter modification in the blocks for PROFINET IO

– FC11 PNIO_SEND (block version 2.0)

– FC12 PNIO_RECV (block version 2.0)

These two functions must be used for CPs that use PROFINET IO controller and device mode at the same time.

CP documentation in the Manual Collection (order no. A5E00069051)

The SIMATIC NET Manual Collection DVD ships with each S7 CP. This DVD is regularly updated and contains the device manuals and descriptions valid at the time it is created.

Information on the current block versions (FCs/FBs)

Always use the latest block versions for new user programs. You will find information on the current block versions and the current blocks to download from the Internet at Download .

When replacing a CP, follow the instructions in the device-specific part of the device manual for your S7 CP.

Version history for the blocks (FCs/FBs) and for the SIMATIC NET S7 CPs

The "Version History/Current Downloads for SIMATIC NET S7 CPs" provides information on all CPs available up to now for SIMATIC S7 (Industrial Ethernet, PROFIBUS, IE/PB Link) and the blocks (FCs/FBs).

An up-to-date version of this document can be found at:

http://support.automation.siemens.com/WW/view/de/9836605 ()

SIMATIC NET Quick Start CD: Examples relating to communication

The Quick Start CD that can be ordered separately is a treasure-trove of sample programs and configurations.

You can order this directly over the Internet at:

http://support.automation.siemens.com/WW/view/de/21827955 ()

Additional information on SIMATIC S7 and STEP 7

The additional documentation on the basic software STEP 7 of the SIMATIC automation system is included in electronic format in your STEP 7 installation.

You will also find information on SIMATIC automation systems on the Quick Start CD and from the Customer Support Online services at:

http://www.automation.siemens.com/net/index_00.htm ()

(General information on SIMATIC NET)

or

http://support.automation.siemens.com/WW/view/de ()

(Product information and downloads)

See also

http://support.automation.siemens.com/WW/view/de/8797900 ()

Table of contents

Overview and general information on working with FCs and FBs 1

1.1 FCs / FBs for Ind. Ethernet / PROFINET

How supplied - block library

The SIMATIC NET FCs (functions) and FBs (function blocks) are supplied with the STEP 7 basic package unless indicated otherwise.

The following list shows the block numbers as they are supplied. You can change these block numbers.

Under SIMATIC_NET_CP, you can also see which folders contain blocks. Please note that you must use different FCs/FBs for the S7300 and S7400 (separate libraries).

Communication service / functional area	**Block type**		**SIMATIC Manager Library**	
			SIMATIC_NET_CP	
			CP 300	CP 400
SEND / RECEIVE (S5compatible communication)	FC5	AG_SEND	x	x
	FC50	AG_LSEND	x [2]	x
	FC53	AG_SSEND		x[3]
	FC6	AG_RECV	x	x
	FC60	AG_LRECV	x [2]	x
	FC63	AG_SRECV		x[3]
	FC7	AG_LOCK	x	x
	FC8	AG_UNLOCK	x	x
	FC10	AG_CNTRL	x [3]	x [3]
Programmed communication connections	FB55	IP_CONFIG	x	x
S7 communication [4]	FB12	BSEND	x	
	FB13	BRCV	x	
	FB15	PUT	x	
	FB14	GET	x	
	FB8	USEND	x	
	FB9	URCV	x	
	FC62	C_CNTRL	x	
FTP (advanced CPs)	FB40	FTP_CMD	x	x
	FC40	FTP_CONNECT	x	x

			SIMATIC Manager Library	
	FC41	FTP_STORE	x	x
	FC42	FTP_RETRIEVE	x	x
	FC43	FTP_DELETE	x	x
	FC44	FTP_QUIT	x	x
PROFINET CBA	FB88 1)	PN_InOut 1)	x 1)	x 1)
	FB90 1)	PN_InOut_Fast 1)		x 1)
PROFINET IO	FC11	PNIO_SEND	x	
	FC12	PNIO_RECV	x	
	FB52	PNIO_RW_REC	x	
	FB54	PNIO_ALARM	x	

Legend:

1) FB88 / FB90 is supplied along with the engineering tool SIMATIC iMap and is entered in the PROFINET system library when you install the STEP 7 addon.
2) Not to be used with the current CPs and not part of the current SIMATIC_NET_CP library.
3) depending on the CP type
4) Described in the STEP 7 documentation

Which block version should I use?

The following descriptions also include information on differences in behavior between the various block versions. Please check and note the version identifiers of the blocks you are using.

The SIMATIC Manager block libraries installed with STEP 7 / NCM S7 contain the block versions that were current at the time of the STEP 7 release.

Note

We recommend that you always use the latest block versions for all module types.

You will find information on the current block versions and the current blocks to download from the Internet in Customer Support:

http://support.automation.siemens.com/WW/view/en/8797900 ()

This recommendation assumes that you are using the latest firmware for the particular module type.

FCs when modules are replaced

Module replacement means the replacement of a module with another module that may be a more recent version.

NOTICE

Please remember that if you replace a module, you must only use the blocks permitted for the configured CP type in the user program.

This means:

- If you replace the module without adapting the configuration data to the possibly newer module type, you do not need to make any changes to the blocks used.
- If you replace the module and you do adapt the configuration data to the newer module type, make sure you use the block versions approved for this module type.

We recommend that you always use the latest block versions for all module types.

This recommendation assumes that you are using the latest firmware for the particular module type.

The specific device manuals contain information on the compatibility of the S7-CPs and the corresponding blocks (FCs / FBs).

1.2 FCs / FBs for PROFIBUS

How supplied - block library

The SIMATIC NET FCs (functions) and FBs (function blocks) are supplied with the STEP 7 basic package unless indicated otherwise.

The following list shows the block numbers as they are supplied. You can change these block numbers.

Under SIMATIC_NET_CP, you can also see which folders contain blocks. Please note that you must use different FCs/FBs for the S7300 and S7400 (separate libraries).

Communication service / functional area	Block type		SIMATIC Manager Library	
			SIMATIC_NET_CP	
			CP 300	CP 400
PROFIBUS DP	FC1	DP_SEND	x	
	FC2	DP_RECV	x	
	FC3	DP_DIAG	x	
	FC4	DP_CTRL	x	
SEND / RECEIVE (S5compatible communication)	FC5	AG_SEND	x	x
	FC50	AG_LSEND		x [2]
	FC6	AG_RECV	x	x
	FC60	AG_LRECV		x [2]
S7 communication [1]	FB12	BSEND	x	[1]
	FB13	BRCV	x	[1]
	FB15	PUT	x	[1]
	FB14	GET	x	[1]

			SIMATIC Manager Library	
	FB8	USEND	x	1)
	FB9	URCV	x	1)
	FC62	C_CNTRL	x	1)
PROFIBUS FMS	FB2	IDENTIFY	x	x
	FB3	READ	x	x
	FB4	REPORT	x	x
	FB5	STATUS	x	x
	FB6	WRITE	x	x

1) as described in the STEP 7 documentation. As with S7-400 any SFBs that can be used are available in the System Function Library.

2) Can be used but has no special function with PROFIBUS.

Which block version should I use?

The following descriptions also include information on differences in behavior between the various block versions. Please check and note the version identifiers of the blocks you are using.

The SIMATIC Manager block libraries installed with STEP 7 / NCM S7 contain the block versions that were current at the time of the STEP 7 release.

Note

We recommend that you always use the latest block versions for all module types.

You will find information on the current block versions and the current blocks to download from the Internet in Customer Support:

http://support.automation.siemens.com/WW/view/en/8797900 ()

This recommendation assumes that you are using the latest firmware for the particular module type.

FCs / FBs and module replacement

Module replacement means the replacement of a module with another module that may be a more recent version.

NOTICE
Please remember that if you replace a module, you must only use the blocks permitted for the configured CP type in the user program. This means: • If you replace the module without adapting the configuration data to the possibly newer module type, you do not need to make any changes to the blocks used. • If you replace the module and you do adapt the configuration data to the newer module type, make sure you use the block versions approved for this module type. We recommend that you always use the latest block versions for all module types. This recommendation assumes that you are using the latest firmware for the particular module type.

The manuals contain information on the compatibility of the S7-CPs and the corresponding blocks (FCs / FBs).

1.3 Setting parameters for block / function calls

Before describing the blocks / FCs in detail, a few general comments on calling and setting parameters for FCs will be useful at this point.

The general information below applies to the following parameter groups that exist for all FCs / FBs:

- Parameters for CP and connection assignment (input parameters)
- Parameters for specifying a CPU data area (input parameters)
- Status information (output parameters)

Calling communication blocks for an S7300

CAUTION
The communication blocks for S7-300 (SIMATIC NET block libraries for S7300 in STEP 7) must not be called in more than one priority class! If, for example, you call a communication block in OB1 and in OB35, block execution could be interrupted by the higherpriority OB. If you call blocks in more than one OB, you must write your program so that a communication block that is currently executing cannot be interrupted by another communication block (for example by disabling/enabling SFC interrupts).

1.4 Parameters for CP and connection assignment (input parameters)

When you call an FC, you transfer the module start address of the S7 CP in the CPLADDR or LADDR parameter. The module start address of the S7 CP can be found in the properties dialog of the CP in the "Address/Input" tab (can be selected in the SIMATIC Manager or in HW Config).

With connectionoriented jobs, you must also reference the connection to be used by its connection ID. You will find this in the properties dialog of the connection under "Block parameters" (refer to the information in NetPro).

Setting block parameters automatically

To ensure correct parameter settings for the block calls, The LAD/STL/FBD editor in STEP 7 provides you with the option of accepting all the relevant parameters automatically from the hardware configuration (HW Config) and from the connection configuration (NetPro).

When assigning the parameters for the block in the user program, follow the steps outlined below:

1. Select the block call and its block parameters;
2. Rightclick and select the menu command "**Connections...**".
3. Depending on the block type, you can now select the connection and/or module intended for the block from a list.
4. Confirm your selection; as far as possible, the available parameter values are entered in the block call.

Response to incorrect addresses

If the S7CPU cannot communicate with the PROFIBUS CP using the specified module start address or cannot identify it as a CP, the errors described below result.

Cause	Reaction / code
No module can be addressed or identified at the specified CP address.	The CPU remains in STOP with system error state; in this case, evaluate the diagnostic buffer of the CPU.
The CP address points to a different module type.	Possible error code in the STATUS parameter of the communication block: 8184_H System error $80B0_H$ The module does not recognize the data record. $80C0_H$ The data record cannot be read. $80C3_H$ Resources (memory) occupied. $80D2_H$ Logical base address is wrong.

NOTICE

If you inadvertently address not a CP but another module type, errors occur that cannot be indicated by the error messages of the FCs/FBs themselves.

1.5 Parameters for specifying a CPU data area (input parameters)

Specifying the data area on the CPU

When you call an FC, you transfer the address and length of the data area on the CPU in which the user data is available or will be stored or which can contain further parameter information.

The ANY pointer data type is used to address this area. For more detailed information on this data type, refer to the STEP 7 online help under the appendix topic "Format of the parameter type ANY".

1.6 Status information (output parameters)

Evaluating status codes

For status evaluation, the following parameters must be evaluated in the user program:

- DONE or NDR
  These parameters (DONE with send jobs and NDR with receive jobs) signal (successful) completion of the job.
- ERROR
  This indicates that the job could not be executed errorfree.
- STATUS
  This parameter supplies detailed information about the execution of the job. Status codes can be returned during execution of the job (DONE=0 and ERROR=0).

Note

Remember that the status codes DONE, NDR, ERROR, STATUS are updated at each block call.

Status codes during CP startup

With a complete restart or restart of the Ethernet CP (for example after activating a switch on the module), the output parameters of the FC are reset as follows:

- DONE = 0
- NDR = 0
- ERROR = 0
- STATUS =
  - 8180$_H$ for AG_RECV / AG_LRECV
  - 8181 $_H$ for AG_SRECV
  - 8181$_H$ for AG_SEND /AG_LSEND / AG_SSEND

FCs / FBs for Industrial Ethernet 2

2.1 FCs for S5-compatible communication (SEND/RECEIVE interface)

2.1.1 Overview of FCs and their use

Overview

The following FCs are available for transferring data on the SEND/RECEIVE interface:

FC	Can be used with [1]		Meaning
	S7-300	S7-400	
AG_SEND (FC5)	x	x	for sending data
AG_RECV (FC6)	x	x	for receiving data
AG_LSEND (FC50)		x	for sending data
AG_LRECV (FC60)		x	for receiving data
AG_SSEND (FC 53)		x	for sending data
AG_SRECV (FC 63)		x	for receiving data

1) Notes on the FCs for an S7300 and S7400

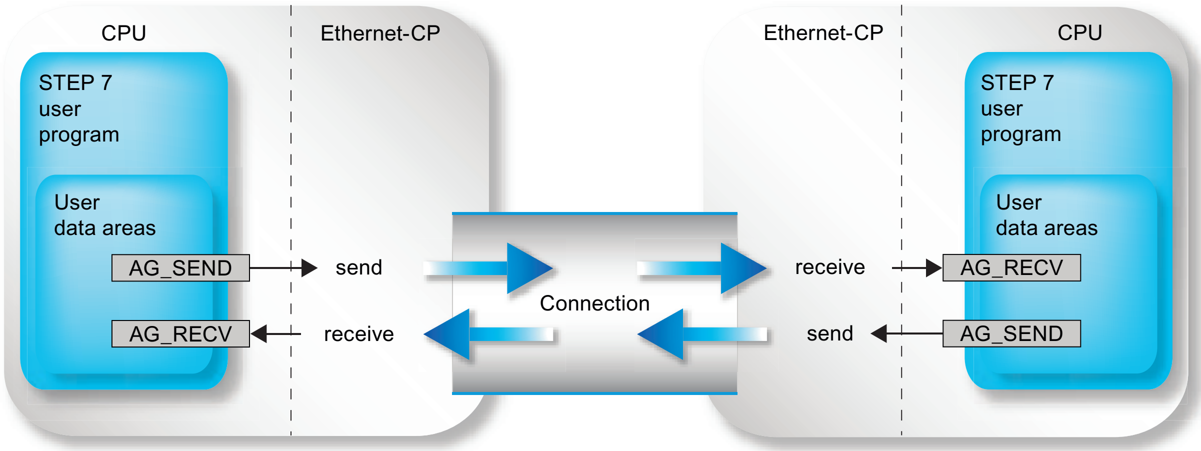
- S7300:
  - With the latest versions of the Ethernet CPs, only FCs AG_SEND and AG_RECV are used; data with a length of up to 8192 bytes can be transferred.
  - With S7–300 CPs (up to 6GK7 343–1EX10–0XE0 with firmware version V2.2), use FC60 on TCP connections instead of FC6. For the CP 343-1 (EX10), you can use FC5/FC6 up to block version V3.0.
- S7400:
  - With FCs AG_SEND / AG_RECV, the data length per job is restricted to <=240 bytes. Longer data records (up to 8192 bytes) can be transferred with FCs AG_LSEND or AG_LRECV.
  - The FCs AG_SSEND and AG_SRECV are for accelerated transfer of data by using optimized block communication between CPU and CP in the S7 station. The fast communication has no effect on LAN communication. These two blocks are supported as of STEP 7 V5.4 SP3.
  - On an S7–400, FC6 cannot be used on TCP connections but only FC60 or FC63.

Further information

Please check the supported data area for the S7CP you are using in the manual for the specific device. You will find an overview of the versions of the FCs/FBs in the SIMATIC NET block history.

Application

The following diagram illustrates the use of the FCs described here for bidirectional data transfer on one configured connection.



Note

Unless specifically stated otherwise, the information on this and the following pages refers to the AG_SEND / AG_LSEND / AG_SSEND or AG_RECV / AG_LRECV / AG_SRECV blocks.

Sample programs

Please note that the following sample programs are also available on the Internet.

- Sample program for the Send-Receive interface with the blocks FC5 (AG_SEND) and FC6 (AG_RECV) for S7-300:

  http://support.automation.siemens.com/WW/view/en/17853532 ()

- Sample program for the Send-Receive interface with the functions FC50 (AG_LSEND) and FC60 (AG_LRECV) for S7-400:

  http://support.automation.siemens.com/WW/view/en/18513371 ()

Specifying the data area on the CPU

When you call an FC, you transfer the address and length of the data area in the CPU. Remember, that the maximum length of the data area depends on the block type and block version being used.

- AG_SEND and AG_RECV
  Up to version V3.0 of these blocks, a maximum of 240 bytes can be sent or received. The current block versions allow a data area of up to 8192 bytes for an S7300. With an S7-400, the FCs AG_LSEND / AG_LRECV must still be used for larger data areas.
- AG_LSEND / AG_LRECV
  Using the CPs of the S7400 and with earlier versions of the S7300, larger data areas can only be transferred with the FCs AG_LSEND or AG_LRECV. Please check the length of the data area in the product information of the CP.
- AG_SSEND / AG_SRECV
  With CPs of the S7−400 that support PROFINET communication in conjunction with CPUs as of version 5.1, data can be transferred at higher transmission speeds with the FCs AG_SSEND or AG_SRECV (does not apply to the CP 443−1 Advanced 6GK7 443–1EX41–0XE0).
  You can check which CP types are supported by CPUs as of version 5.1 in the manual of your CP (Section "Requirements for use").

The following table shows the limit values of the various connection types.

FC	ISO transport	ISO-on-TCP	TCP	UDP
AG_LSEND (S7-400) AG_SEND (S7-300)	8192 bytes	8192 bytes	8192 bytes	2048 bytes
AG_SEND (S7-400)	240 bytes	240 bytes	240 bytes	240 bytes
AG_LRECV (S7-400) AG_RECV (S7-300)	8192 bytes	8192 bytes	8192 bytes	2048 bytes
AG_RECV (S7-400)	240 bytes	240 bytes	240 bytes	240 bytes
AG_SSEND (S7-400) AG_SRECV (S7-400)	1452 bytes	1452 bytes	1452 bytes	1452 bytes

Note

For information on the length of the data area you can transfer with older versions of the Ethernet CPs, refer to the product information bulletin / manual of the Ethernet CP you are using.

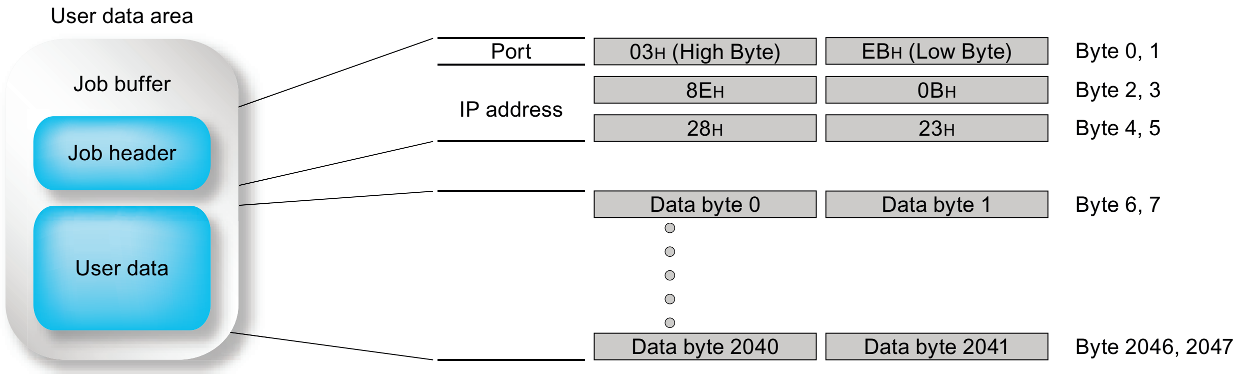
Use without job header

On specified connections, the address and job parameters are specified by the connection configuration. The user program only provides the user data in the UDP data area when sending with AG_SEND / AG_LSEND / AG_SSEND or receives the data with AG_RECV / AG_LRECV / AG_SRECV.

Use with header

Free UDP connections require a job header in the user data area.

The following schematic illustrates the structure of the job buffer and the meaning and location (high byte / low byte) of the parameters in the job header.



Port	03_H (High Byte)	EB_H (Low Byte)	Byte 0, 1
IP address	$8E_H$	$0B_H$	Byte 2, 3
	28_H	23_H	Byte 4, 5
	Data byte 0	Data byte 1	Byte 6, 7
	Data byte 2040	Data byte 2041	Byte 2046, 2047

Figure 2-1 Sending and receiving on a free UDP connection with programmed addresses

- In the diagram (entries in hexadecimal) the following IP address is assumed as an example: 142.11.40.35;
- For the port address 1003, the following would be entered: For high byte: 03_H; For low byte: EB_H.
- The user data area can be up to 2048 bytes. Up to 2042 bytes of user data can be transferred. 6 bytes are reserved for the job header.
  Please note that the data length specified in the block call (LEN parameter) must include the header and the user data!

Change call parameters only after job confirmation

NOTICE
Once the job has been triggered, you can only change the call parameters of the FC call interface of the FCs AG_SEND or AG_RECV after the FC has confirmed completion of the job with DONE=1 or with ERROR=1. If you do not keep to this rule, it is possible that the job will be aborted with an error.

Status display on the FC call interface; Special case with FC versions (only for S7-300) *)

With the FCs AG_SEND (FC5) and AG_RECV (FC6), you will receive the codes shown below in the following situations:

- CP is in STOP;
- Connection is not configured;
- Connection is not established

- Connection is aborted;

Codes:

- AG_SEND:
  DONE=0; ERROR=1; Status=8183$_H$
- AG_RECV:
  DONE=0; ERROR=0; Status=8180$_H$
  or
  DONE=0; ERROR=1; Status=8183$_H$

*) applies to FCs as of version 4.0

2.1.2 FC5 AG_SEND / FC50 AG_LSEND / FC53 AG_SSEND

2.1.2.1 Meaning and call - AG_SEND / AG_LSEND / AG_SSEND

Meaning of the block

The FCs AG_SEND / AG_LSEND / AG_SSEND pass data to the Ethernet CP for transfer over a configured connection.

The selected data area can be a memory bit area or a data block area.

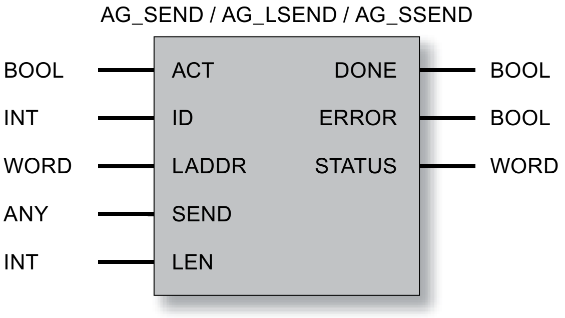
Errorfree execution of the function is indicated when the entire user data area could be sent over Ethernet.

The way in which the FC functions depends on the CP type you are using. Please note the differences in the following section.

Note:
Unless otherwise stated, all the following information applies equally to the FCs AG_SEND, AG_LSEND and AG_SSEND.

Call interface

Call interface in FBD representation



Example in STL representation

STL	**Explanation**
`call fc 5(`	`//Block call`
`ACT:=M 10.0,`	`//Job triggered by memory bit`
`ID:=MW 12,`	`//Connection ID acc. to configuration`
`LADDR:=W#16#0100,`	`//=LADDR 256 dec. in HW Config`
`SEND := P#db99.dbx10.0 byte 240,`	`//Buffer with send data`
`LEN:=MW 14,`	`//Length for send data`
`DONE:=M 10.1,`	`//Execution code`
`ERROR:=M 10.2,`	`//Error code`
`STATUS:=MW 16 );`	`//Status code`

Note

If you want to use FC53 AG_SSEND, you will need to select the "SPEED SEND/RECV" mode in the connection properties during configuration of the connection.

2.1.2.2 How AG_SEND / AG_LSEND / AG_SSEND work

Operating principle

The following diagrams illustrate the normal sequence of data transmission triggered in the user program using AG_SEND.

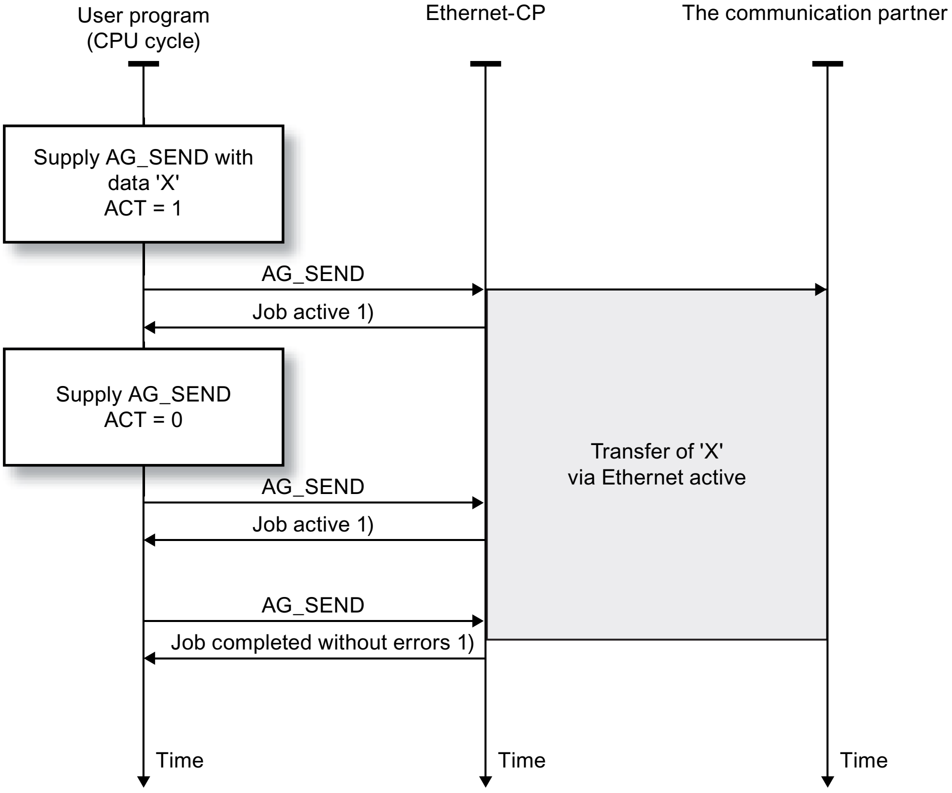
The way in which the FC functions depends on the CP type you are using.

- Case 1: Sequence with FC5, FC50, FC53 in S7-400 CPs
  In the S7–400, the transfer of the entire data area regardless of its length is handled by the CP after the first block call.
- Case 2: Sequence with FC5 in S7–300 CPs
  In the S7-300, the transfer takes place several data segments (each with 240 bytes of user data) and requires several FC calls to transfer the entire data.

Case 1: Sequence with FC5, FC50, FC53 in S7–400 CPs

The send job is executed as soon as the parameter ACT = 1 is passed.
Following this, the parameter ACT = 0 must be passed in at least one further call.

The status code in the output parameters DONE, ERROR and STATUS is updated in each block call and can be evaluated. To update the status code without starting a new send job, start a new block call with the parameter ACT = 0.

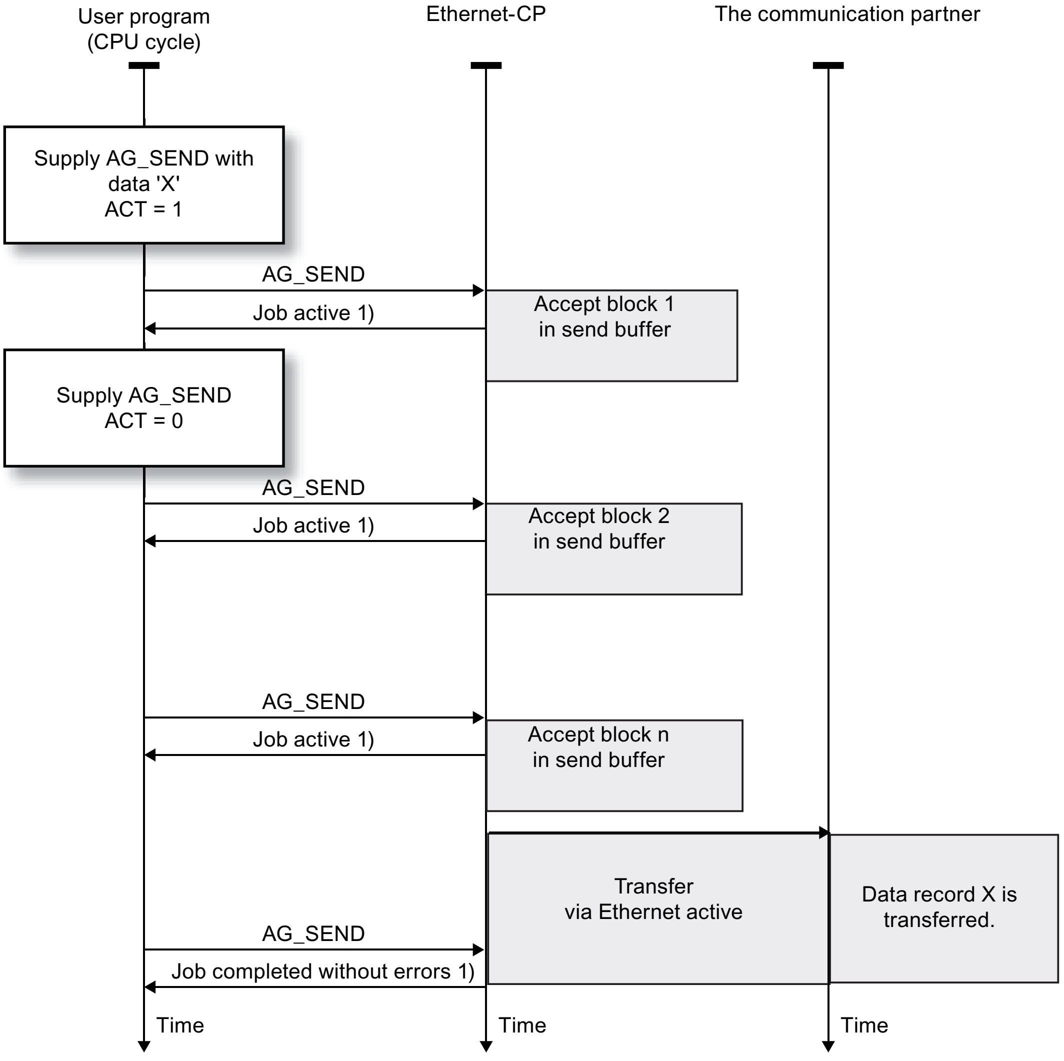


1) Parameter transfer DONE, ERROR, STATUS

Case 2: Sequence with FC5 in S7–300 CPs

The send job is started as soon as the parameter ACT = 1 is passed.
In contrast to case 1, the protocol used here to transfer the data segments (each 240 bytes of user data) requires the FC to be called again for each segment.
Depending on the length of the user data, you must therefore continue to call the FC with ACT=0 until transfer of the entire data record is indicated; at least one further call is necessary. The data is transferred to the communication partner in segments of 240 bytes.

The status code in the output parameters DONE, ERROR and STATUS is updated in each block call and can be evaluated.


User program
(CPU cycle)
Ethernet-CP
The communication partner
Supply AG_SEND with data 'X' ACT = 1
AG_SEND
Job active 1)
Accept block 1 in send buffer
Supply AG_SEND ACT = 0
AG_SEND
Job active 1)
Accept block 2 in send buffer
AG_SEND
Job active 1)
Accept block n in send buffer
Transfer via Ethernet active
Data record X is transferred.
AG_SEND
Job completed without errors 1)
Time
Time
Time

1) Parameter transfer DONE, ERROR, STATUS

Note

In principle, it is possible to call the FC more than once within the CPU cycle to speed up the handling of the job. Do not forget, however, that this increases the load during the CPU cycle (the load differs depending on the CPU type)!

2.1.2.3 Explanation of the formal parameters - AG_SEND / AG_LSEND / AG_SSEND

Explanation of the formal parameters

The following table explains all the formal parameters for the AG_SEND / AG_LSEND / AG_SSEND functions:

Parameter	Declaration	Data type	Possible values	Description
ACT	INPUT	BOOL	0,1	If an FC is called with ACT=1, LEN bytes are sent from the ISO transport data area specified with the SEND parameter. If an FC is called with ACT = 0, the status codes DONE, ERROR and STATUS are updated.
ID	INPUT	INT	1,2...64 (S7-400) 1,2...16 (S7-300)	The connection number of the connection is specified in the parameter ID.
LADDR	INPUT	WORD		Module start address When you configure the CP with STEP 7 HW Config, the module start address is displayed in the configuration table. Specify this address here.
SEND	INPUT	ANY		Specifies the address and length The address of the data area points to one of the alternatives: • Memory bit area • Data block area
LEN	INPUT	INT	On ISO transport and ISOonTCP / TCP: 1,2,...8192 (or up to "length specified for SEND parameter") On UDP: 1,2,...2048 (or up to "length specified for SEND parameter")	Number of bytes to be sent from the data area with this job. The possible values range from 1 to length specified for the SEND parameter. • Note the block type: – For S7-300 The current versions of FC AG_SEND allow up to 8192 bytes (2048 bytes for UDP) to be transferred. – For S7-400 With FC AG_SEND, the data area is restricted to a maximum of 240 bytes. Note the following with an S7-400: • Improved performance with shorter data records: Transfer of data records up to 240 bytes results in better performance! This applies regardless of the block type used (AG_SEND/AG_LSEND). • With AG_SSEND, the data area is restricted to a maximum of 1452 bytes.
DONE	OUTPUT	BOOL	0: Job active 1: Job done	The status parameter indicates whether or not the job was completed without errors. As long as DONE = 0, no further job can be triggered. DONE is set to 0 by the CP when it accepts a new job. For the meaning in conjunction with the ERROR and STATUS parameters, refer to the following table.

Parameter	Declaration	Data type	Possible values	Description
ERROR	OUTPUT	BOOL	0: - 1: Error	Error code For the meaning in conjunction with the DONE and STATUS parameters, refer to the follow table.
STATUS	OUTPUT	WORD	See following table	Status code The following table shows the condition codes formed by the DONE and ERROR parameters.

2.1.2.4 Condition codes of AG_SEND, AG_LSEND and AG_SSEND

Condition codes

The following table shows the condition codes formed based on DONE, ERROR and STATUS that must be evaluated by the user program.

Note

For entries coded with 8FxxH in STATUS, refer to the information in the STEP 7 Standard and System Functions reference manual. The chapter describing error evaluation with the RET_VAL output parameter contains detailed information.

To find out which SFCs are used and are relevant for error evaluation, display the properties dialog of the FC described here in the "Calls" tab.

DONE	ERROR	STATUS	Meaning
1	0	0000_H	Job completed without error.
0	0	0000_H	No job being executed.
0	0	8181_H	Job active.
0	1	7000_H	The condition code is possible only with S7-400: The FC was called with ACT=0; the job has not yet been processed.
0	1	8183_H	No configuration or the ISO/TCP service has not yet started on the Ethernet CP.
0	1	8184_H	• Illegal data type specified for the SEND parameter. • System error (the source data area is incorrect).
0	1	8185_H	LEN parameter longer than SEND source area.
0	1	8186_H	ID parameter invalid. • ID != 1,2....16 (S7-300). • ID != 1,2....64.(S7-400)
0	1	8302_H	No receive resources on the destination station; the receiving station cannot process received data quickly enough or has not prepared any receive resources.
0	1	8304_H	The connection is not established. The send job should only be attempted again after waiting for at least 100 ms.
0	1	8311_H	The destination station cannot be obtained under the specified Ethernet address.
0	1	8312_H	Ethernet error on the CP.

DONE	ERROR	STATUS	Meaning
0	1	$8F22_H$	Source area invalid, e.g.: Area does not exist in the DB LEN parameter < 0
0	1	$8F24_H$	Area error reading a parameter.
0	1	$8F28_H$	Alignment error reading a parameter.
0	1	$8F32_H$	Parameter contains a DB number that is too high.
0	1	$8F33_H$	DB number error.
0	1	$8F3A_H$	Area not loaded (DB).
0	1	$8F42_H$	Timeout reading a parameter from the I/O area.
0	1	$8F44_H$	Access to a parameter to be read during block execution is prevented.
0	1	$8F7F_H$	Internal error, e.g. illegal ANY reference e.g. parameter LEN=0
0	1	8090_H	• Module with this module start address does not exist; • The FC being used does not match the system family being used (remember to use different FCs for S7300 and S7400).
0	1	8091_H	Module start address not at a doubleword boundary.
0	1	8092_H	In the ANY reference, a type other than BYTE is specified. (S7-400 only)
0	1	$80A4_H$	The communication bus connection between the CPU and CP is not established. (With newer CPU versions)
0	1	$80B0_H$	The module does not recognize the data record.
0	1	$80B1_H$	The specified length (in the LEN parameter) is incorrect.
0	1	$80B2_H$	The communication bus connection between the CPU and CP is not established.
0	1	$80C0_H$	The data record cannot be read.
0	1	$80C1_H$	The specified data record is currently being processed.
0	1	$80C2_H$	There are too many jobs pending.
0	1	$80C3_H$	CPU resources (memory) occupied.
0	1	$80C4_H$	Communication error (occurs temporarily and a repetition in the user program will often remedy the problem).
0	1	$80D2_H$	Module start address incorrect.

2.1.3 FC6 AG_RECV / FC60 AG_LRECV / FC63 AG_SRECV

2.1.3.1 Meaning and call - AG_RECV / AG_LRECV / AG_SRECV

Meaning of the block

The AG_RECV / AG_LRECV / AG_SRECV function receives the data transferred on a configured connection from the Ethernet CP.

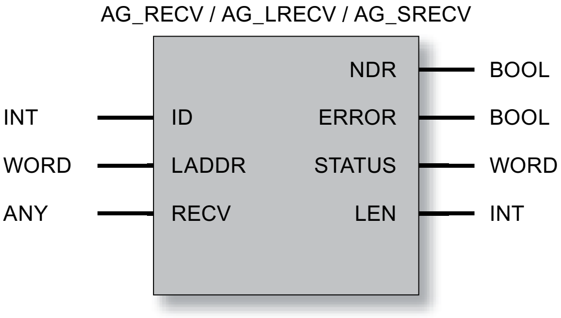
The data area specified for the receive data can be a memory bit area or a data block area.

Errorfree execution is indicated when the data could be received from the Ethernet CP.

Note:
Unless otherwise stated, all the following information applies to both the FCs AG_RECV and AG_LRECV / AG_SRECV.

Call

Call interface in FBD representation



Example in STL representation

STL	**Explanation**
`call fc 6(`	`//Block call`
`ID:=MW 40,`	`//Connection ID acc. to configuration`
`LADDR:=W#16#0100,`	`//=LADDR 256 dec. in HW Config`
`RECV:=P#M 0.0 BYTE 100,`	`//Buffer for received data`
`NDR:=DB 110.DBX 0.6,`	`//Receive code`
`ERROR:=DB 110.DBX 0.7,`	`//Error code`
`STATUS:=DB 110.DBW 2,`	`//Status code`
`LEN:=DB 110.DBW 4 );`	`//Received data length`

Note

If you want to use FC63 AG_SRECV, you will need to select the "SPEED SEND/RECV" mode in the connection properties during configuration of the connection.

2.1.3.2 How AG_RECV / AG_LRECV / AG_SRECV work

Operating principle

The following diagram illustrates the normal sequence of data acceptance triggered by an AG_RECV in the user program.

Each AG_RECV job in the user program is acknowledged by the Ethernet CP with an entry in the output parameters NDR, ERROR and STATUS.

The way in which the FC functions depends on the CP type you are using and the connection types.

- Case 1: Sequence with FC6 in S7-300 CPs

  With the newer CP types, optimized data transfer on the SEND/RECEIVE interface is available. In particular with longer data records, this allows a much higher data throughput on the interface between the CPU and CP.

- Case 2: Sequence with FC6 and FC60 in S7-400 CPs

  With FC6 / FC60 AG_RECV, the response on the S7-400 depends on the protocol used.

  – Case 2a: Sequence with ISO transport, ISO-on-TCP, UDP connections

    Wit these connection types, the transfer is handled by the CP with one or more FC6 /FC60 calls depending on the length of the data area.

  – Case 2b: Sequence with TCP connections

    On a TCP connection, the length specified in the ANY pointer of the RECV parameter is the deciding factor. An FC6 /FC60 job is completed with the condition code NDR=1, as soon as an amount of data corresponding to the specified length has been written to the receive buffer.

- Case 3: Sequence with FC63 in S7-400 CPs

  With FC63 AG_SRECV, the response on the S7-400 depends on the protocol used.

  – Case 3a: Sequence with ISO transport, ISO-on-TCP, UDP connections

    With these connection types, the transfer of the entire data area regardless of its length is started by the CP after the first block call.

  – Case 3b: Sequence with TCP connections

    On a TCP connection, the data on the CP is accepted up to the maximum specified job length with every call.

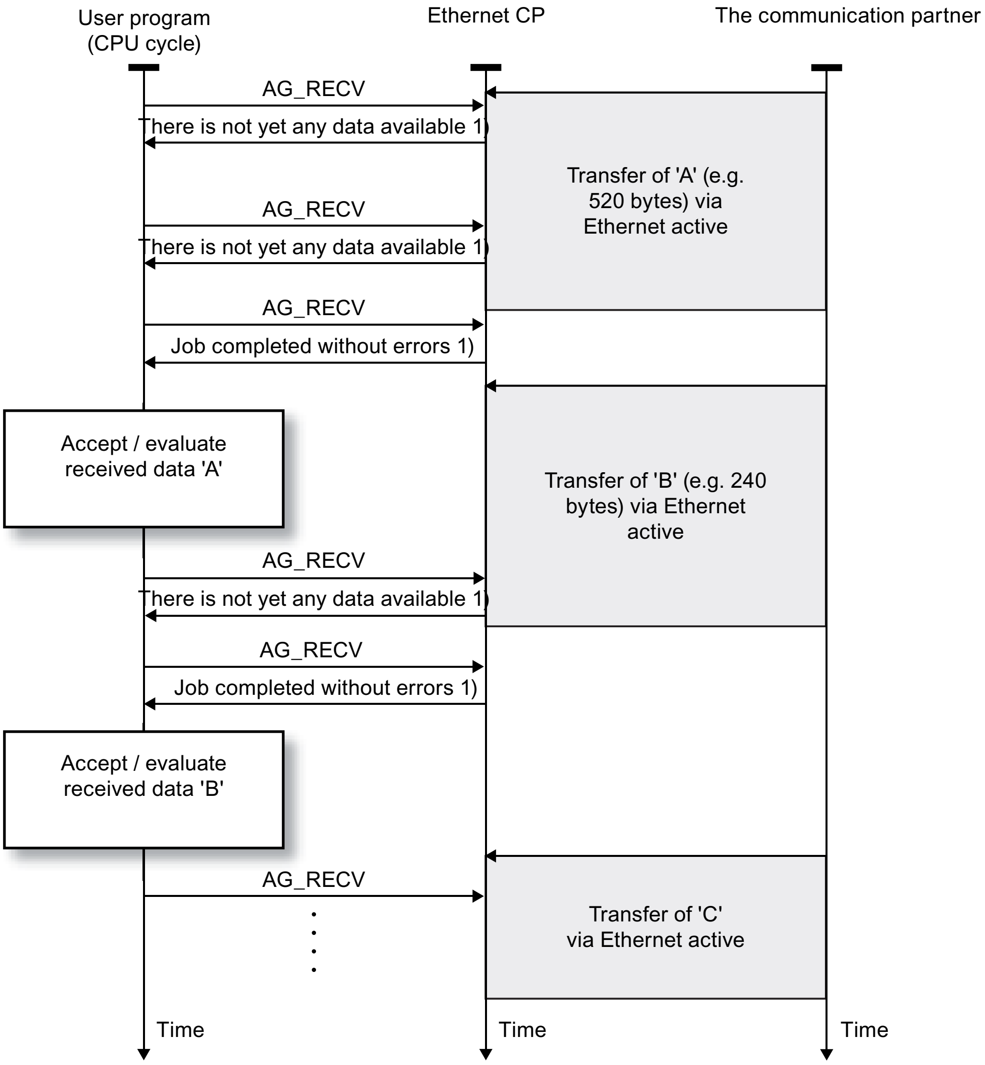
    The call must be repeated until a data record has been entered completely and consistently in the receive buffer. The reception of the completed data record is indicated in one of the later FC calls with the parameter NDR=1.

Case 1: Sequence with FC6 in S7-300 CPs

When FC6 is called, the user program prepares the buffer to receive data and instructs the CP to enter the received data there.

The protocol used here to transfer the data to the receive buffer requires the FC to be called again for each segment (240 bytes of user data).
Depending on the length of the user data, the FC must be called repeatedly until the complete transfer is indicated by parameter NDR=1.

The status code in the output parameters NDR, ERROR and STATUS is updated in each block call and can be evaluated.


User program
(CPU cycle)
Ethernet CP
The communication partner
AG_RECV
There is not yet any data available 1)
Transfer of 'A' (e.g. 520 bytes) via Ethernet active
AG_RECV
There is not yet any data available 1)
AG_RECV
Job completed without errors 1)
Accept / evaluate received data 'A'
Transfer of 'B' (e.g. 240 bytes) via Ethernet active
AG_RECV
There is not yet any data available 1)
AG_RECV
Job completed without errors 1)
Accept / evaluate received data 'B'
AG_RECV
Transfer of 'C' via Ethernet active
Time
Time
Time

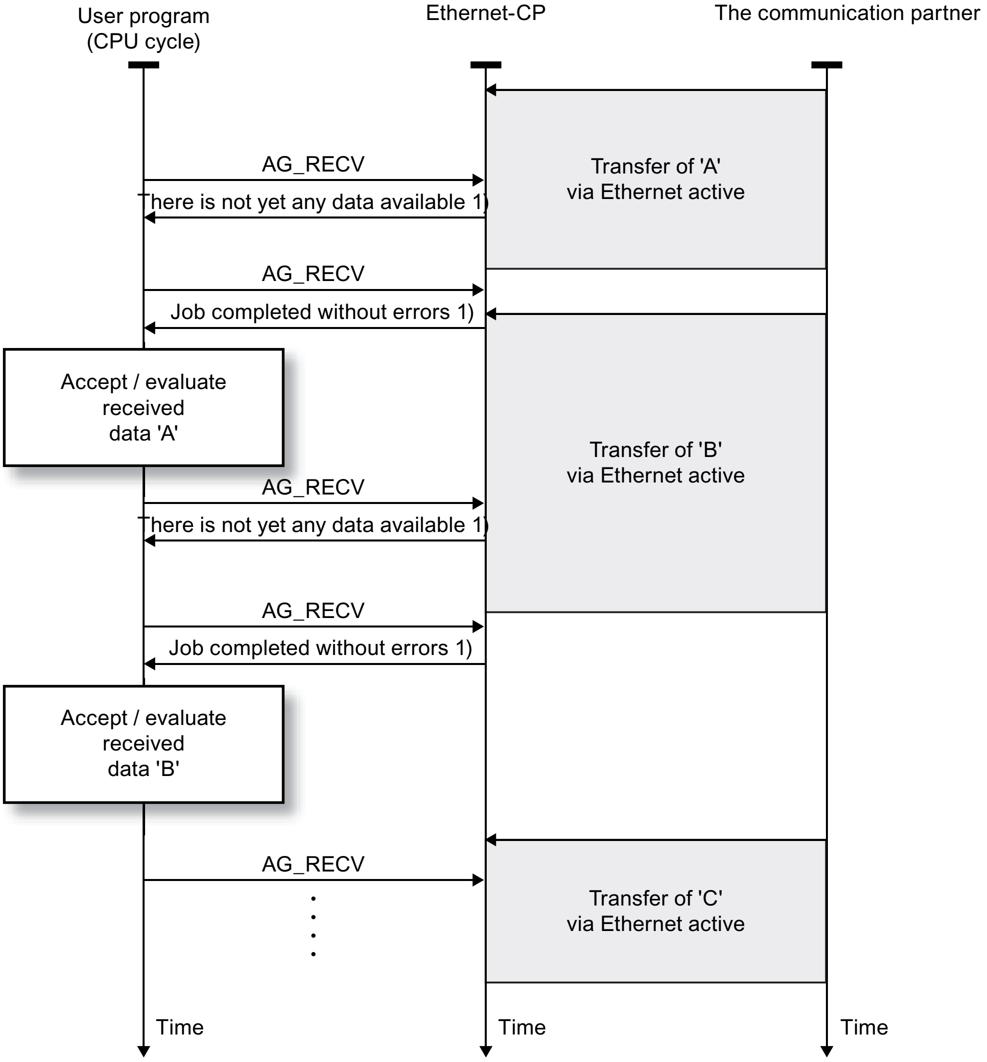
1) Parameter transfer NDR, ERROR, STATUS

Case 2a: Sequence with FC6 and FC60 in S7-400 CPs (with ISO transport, ISO-on-TCP, UDP connections)

When the FC is called, the user program prepares the buffer to receive data and instructs the CP to enter all available data there.

As soon as a data record has been entered fully and consistently in the receive buffer, this is indicated by the parameter NDR=1 in one of the next FC calls.

The status code in the output parameters NDR, ERROR and STATUS is updated in each block call and can be evaluated.

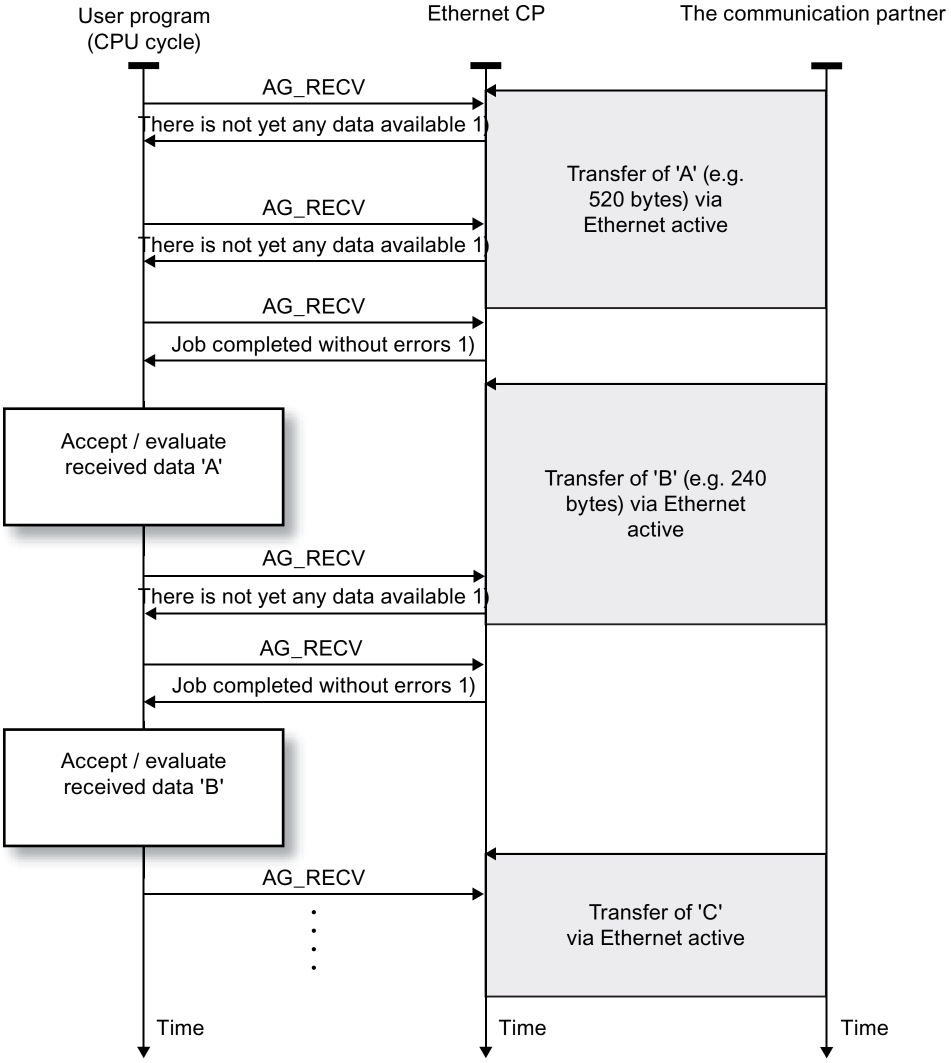


1) Parameter transfer DONE, ERROR, STATUS

Case 2b: Sequence with FC6 / FC60 in S7-400 CPs (only for TCP connections)

On a TCP connection, the length specified in the ANY pointer of the RECV parameter is the deciding factor. An FC6 /FC60 job is completed with the condition code NDR=1, as soon as an amount of data corresponding to the specified length has been written to the receive buffer.

The sequence example shows a situation in which the length in the ANY pointer was set to 400 bytes for an FC60.


User program
(CPU cycle)
Ethernet CP
The communication partner
AG_RECV
There is not yet any data available 1)
Transfer of 'A' (e.g. 520 bytes) via Ethernet active
AG_RECV
There is not yet any data available 1)
AG_RECV
Job completed without errors 1)
Accept / evaluate received data 'A'
Transfer of 'B' (e.g. 240 bytes) via Ethernet active
AG_RECV
There is not yet any data available 1)
AG_RECV
Job completed without errors 1)
Accept / evaluate received data 'B'
AG_RECV
Transfer of 'C' via Ethernet active
Time
Time
Time

1) Parameter transfer NDR, ERROR, STATUS

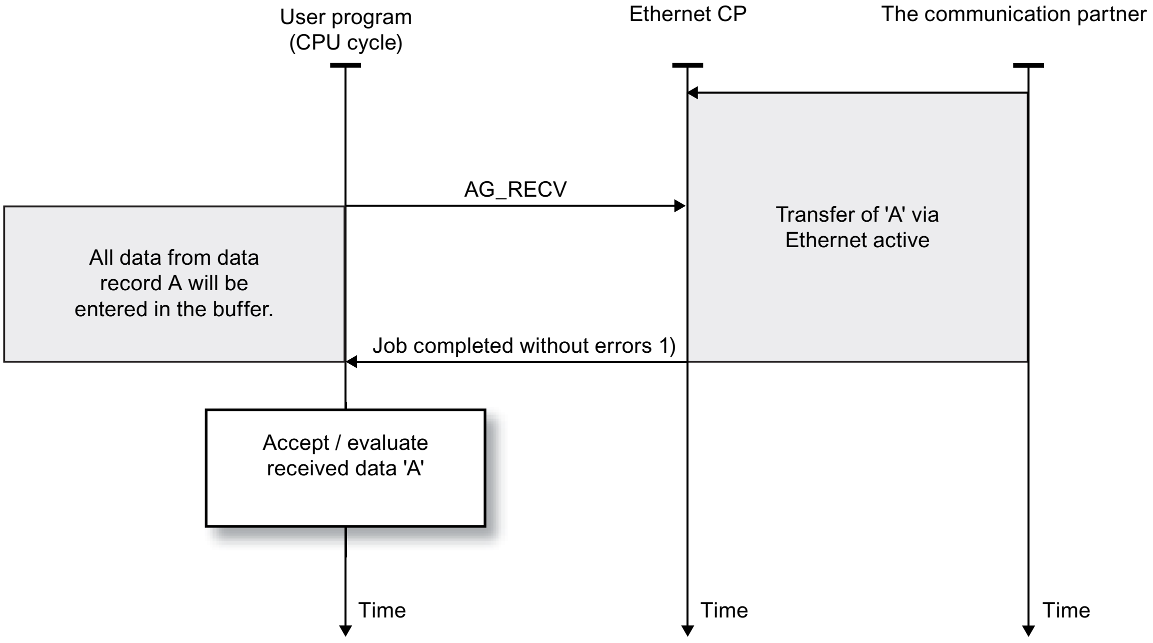
Case 3a: Sequence with FC63 in S7-400 CPs (with ISO transport, ISO-on-TCP, UDP connections)

When the FC is called, the user program prepares the buffer for the received data and instructs the CP to enter all data until the end of the transfer there; in other words, when the length specified in the ANY pointer is reached.

As soon as the data record has been entered fully and consistently in the receive buffer, this is indicated by the parameter NDR=1 in one of the next FC calls.

The maximum length for received data is 1452 bytes. The size of the receive buffer must always be set to this value.

The status code in the output parameters NDR, ERROR and STATUS is updated in each block call and can be evaluated.


User program
(CPU cycle)
Ethernet CP
The communication partner
AG_RECV
Transfer of 'A' via Ethernet active
All data from data record A will be entered in the buffer.
Job completed without errors 1)
Accept / evaluate received data 'A'
Time
Time
Time

1) Parameter transfer NDR, ERROR, STATUS

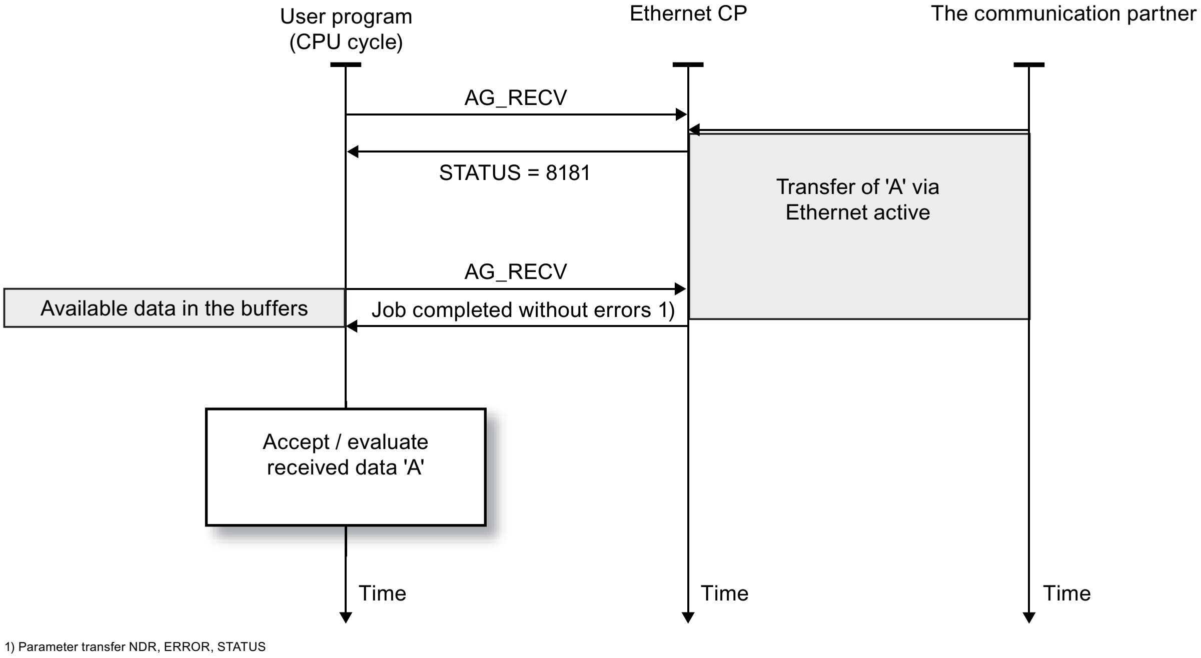
Case 3b: Sequence with FC63 in S7-400 CPs (only for TCP connections)

When the FC is called, the user program prepares the buffer to receive data and instructs the CP to enter the currently available data there. Until the data has been completely entered in the receive buffer, the message "Job active" (8181H) is sent.

With each new FC call, the currently available data is entered in the receive buffer. When the data record has been entered fully and consistently in the receive buffer, this is indicated by the parameter NDR=1 in one of the next FC calls.

The maximum length for received data is 1452 bytes. The size of the receive buffer must always be set to this value.

The status code in the output parameters NDR, ERROR and STATUS is updated in each block call and can be evaluated.



2.1.3.3 Explanation of the formal parameters - AG_RECV / AG_LRECV / AG_SRECV

Explanation of the formal parameters

The following table explains all the formal parameters for the AG_RECV / AG_LRECV / AG_SRECV function:

Parameter	Declaration	Data type	Possible values	Description
ID	INPUT	INT	1,2...64 (S7-400) 1,2...16 (S7-300)	The connection number of the ISO transport connection is specified in the ID parameter.
LADDR	INPUT	WORD		Module start address When you configure the CP with STEP 7 HW Config, the module start address is displayed in the configuration table. Specify this address here.

Parameter	Declaration	Data type	Possible values	Description
RECV	INPUT	ANY		Specifies the address and length The address of the data area points to one of the alternatives: • Memory bit area • Data block area Note on length: Performance is improved when transferring data records up to 212 bytes if you also restrict the length to 212 bytes at the RECV parameter. Note on FC63 AG_SRECV: With FC63 AG_SRECV, always set RECV to the maximum receive buffer length of 1452 bytes. Otherwise, the following error can occur in certain situations: NDR=0; ERROR=1; STATUS=8185$_H$
NDR	OUTPUT	BOOL	0: - 1: new data	The parameter indicates whether or not new data was accepted. For the meaning in conjunction with the ERROR and STATUS parameters, refer to the following table.
ERROR	OUTPUT	BOOL	0: - 1: Error	Error code For the meaning in conjunction with the NDR and STATUS parameters, refer to the following table.
STATUS	OUTPUT	WORD	See following table	Status code For the meaning in conjunction with the NDR and ERROR parameters, refer to the following table.
LEN	OUTPUT	INT	**On ISO Transport and ISOonTCP:** 1,2,...8192 **On UDP:** 1,2,...2048	Specifies the number of bytes accepted from the Ethernet CP and entered in the data area. Note the block type: • For S7-300 The current versions of FC AG_RECV allow up to 8192 bytes (2048 bytes for UDP) to be transferred. • For S7-400 With FC AG_RECV, the data area is restricted to a maximum of 240 bytes. With FC AG_SRECV, the data area is restricted to a maximum of 1452 bytes.

2.1.3.4 Condition codes of AG_RECV, AG_LRECV and AG_SRECV

Condition codes

The following table shows the codes formed by the NDR, ERROR and STATUS parameters that must be evaluated by the user program.

Note

For entries coded with 8FxxH in STATUS, refer to the information in the STEP 7 Standard and System Functions reference manual. The chapter describing error evaluation with the RET_VAL output parameter contains detailed information.

To find out which SFCs are used and are relevant for error evaluation, display the properties dialog of the FC described here in the "Calls" tab.

Table 2-1 AG_RECV / AG_LRECV / AG_SRECV condition codes

NDR	ERROR	STATUS	Meaning
1	0	0000$_H$	New data accepted.
0	0	8180$_H$	There is no data available yet (not with AG_SRECV).
0	0	8181$_H$	Job active.
0	1	8183$_H$	• The configuration is missing; • The ISO transport service has not yet started on the Ethernet CP; • The connection is not established.
0	1	8184$_H$	• Illegal type specified for the RECV parameter; • System error.
0	1	8185$_H$	Destination buffer (RECV) is too short.
0	1	8186$_H$	ID parameter invalid. ID != 1,2....16 (S7-300). ID != 1,2....64.(S7-400)
0	1	8304$_H$	The connection is not established. The send job should only be attempted again after waiting for at least 100 ms.
0	1	8F23$_H$	Source area invalid, e.g.: Area does note exist in the DB.
0	1	8F25$_H$	Area error writing a parameter.
0	1	8F29$_H$	Alignment error writing a parameter
0	1	8F30$_H$	Parameter is in the writeprotected first current data block.
0	1	8F31$_H$	Parameter is in the writeprotected second current data block.
0	1	8F32$_H$	Parameter contains a DB number that is too high.
0	1	8F33$_H$	DB number error.
0	1	8F3A$_H$	Destination area not loaded (DB).
0	1	8F43$_H$	Timeout writing a parameter to the I/O area.
0	1	8F45$_H$	Address of the parameter to be read is disabled in the access track.
0	1	8F7F$_H$	Internal error, e.g. illegal ANY reference.
0	1	8090$_H$	• No module with this module start address exists or the CPU is in STOP mode; • The FC being used does not match the system family being used (remember to use different FCs for S7300 and S7400).
0	1	8091$_H$	Module start address not at a doubleword boundary.
0	1	8092$_H$	In the ANY reference, a type other than BYTE is specified. (S7-400 only)

NDR	ERROR	STATUS	Meaning
0	1	$80A0_H$	Negative acknowledgment reading from the module.
0	1	$80A4_H$	The communication bus connection between the CPU and CP is not established.
0	1	$80B0_H$	The module does not recognize the data record.
0	1	$80B1_H$	Destination area invalid.
0	1	$80B2_H$	The communication bus connection between the CPU and CP is not established.
0	1	$80C0_H$	The data record cannot be read.
0	1	$80C1_H$	The specified data record is currently being processed.
0	1	$80C2_H$	There are too many jobs pending.
0	1	$80C3_H$	CPU resources (memory) occupied.
0	1	$80C4_H$	Communication error (occurs temporarily and a repetition in the user program will often remedy the problem).
0	1	$80D2_H$	Module start address incorrect.

2.2 FCs for access coordination with FETCH/WRITE

2.2.1 Overview of FCs and their use

Overview

The following FCs are available for FETCH/WRITE function to coordinate access:

FC	can be used with:		Meaning
	S7-300	S7-400	
AG_LOCK (FC7)	x	x	Locks external data access with FETCH/WRITE.
AG_UNLOCK (FC8)	x	x	Releases external data access with FETCH/WRITE.

Caution when Configuring

If you use FCs AG_LOCK and AG_UNLOCK, you must specify the following information for CPs in S7400 stations in the configuration:

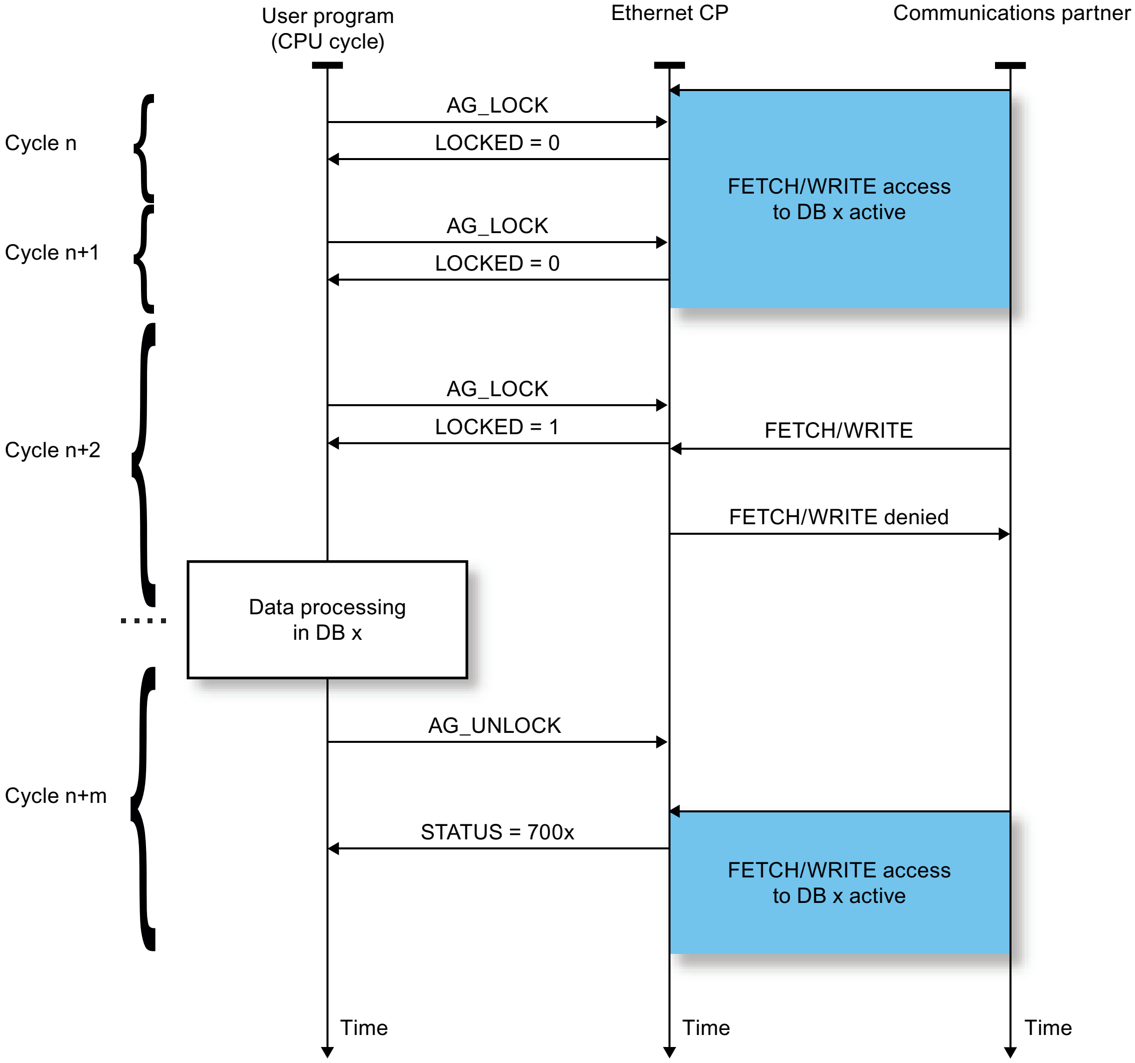
- Under "Properties > Addresses"
  The "Address setting for LOCK/UNLOCK" option must be selected if the selection is available.

How It works

With these FCs, you can coordinate access to system memory areas so that no inconsistent data is created and transferred. The control is from the user program in the S7 CPU that can, if necessary, disable an external FETCH/WRITE access using an AG_LOCK call. After a certain time or after the local write/read access is completed, an AG_UNLOCK job can be used to enable external access again.

Another advantage is that this access lock only applies to the FETCH/WRITE connection specified in the call. If more than one FETCH/WRITE connection is configured, these can, for example, be used for certain specific system areas and a selective access coordination can be implemented.

The following diagram illustrates the usual chronological sequence of memory access coordination controlled in the user program with AG_LOCK and AG_UNLOCK.



The lock job must first be monitored in the user program using the code in the return parameter LOCKED. As long as LOCKED=0 is indicated, it must be assumed that there is still an external FETCH/WRITE access active.

If LOCKED=1 is indicated, this shows that the lock is active; data can now be modified by the user program.

The status code is updated at each block call.

2.2.2 FC7 AG_LOCK

2.2.2.1 Meaning and call - AG_LOCK

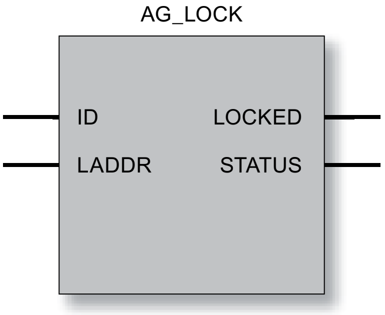
Meaning of the block

Using the AG_LOCK block the data exchange using FETCH or WRITE on the connection selected with the parameter ID is disabled. The LOCKED output indicates whether or not the lock was successful. If the lock was not successful, the job must be triggered again in a later CPU cycle.

The STATUS output indicates the status of the CP for this connection.

Call

Call interface in FBD representation



Example in STL representation

STL	Explanation
`call fc 7(`	`//Block call`
`ID:=DB 100.DBW 2,`	`//Connection ID acc. to configuration`
`LADDR:=W#16#0100,`	`//=LADDR 256 dec. in HW Config`
`LOCKED:=DB 100.DBX 0.6,`	`//Status code of access lock`
`STATUS:=DB 100.DBW 4 );`	`//Status code`

2.2.2.2 Explanation of the formal parameters - AG_LOCK

Explanation of the formal parameters

The following table explains all the formal parameters for the AG_LOCK function:

Parameter	Declaration	Data type	Possible values	Description
ID	INPUT	INT	1,2...16 for S7300 1,2...64 for S7400	The connection number of the connection is specified in the parameter ID. (See Configuration)
LADDR	INPUT	WORD		Module start address When you configure the CP with STEP 7 HW Config, the module start address is displayed in the configuration table. Specify this address here.
LOCKED	OUTPUT	BOOL	0: not (yet) locked 1: locked	Shows the status of the access lock requested on the specified FETCH/WRITE connection.
STATUS	OUTPUT	WORD	See following table	Status code For meaning, refer to the table below.

2.2.2.3 Condition codes of AG_LOCK

Condition codes

The following table shows the STATUS code that must be evaluated by the user program.

Table 2-2 AG_LOCK condition codes

STATUS	Meaning
7000$_H$	CP is not processing a job
7001$_H$	FETCH active
7002$_H$	WRITE active
8183$_H$	FETCH/WRITE not configured for this connection (S7-400 only)
8186$_H$	ID number not in permitted range (e.g. 1...64 for S7-400 Industrial Ethernet CPs)
80A4$_H$	The communication bus connection between the CPU and CP is not established. (Only with newer CPU versions)
80B0$_H$	The module does not recognize the data record.
80B1$_H$	The specified length (in the LEN parameter) is incorrect.
80B2$_H$	The communication bus connection between the CPU and CP is not established.
80C0$_H$	The data record cannot be read.
80C1$_H$	The specified data record is currently being processed.
80C2$_H$	There are too many jobs pending.
80C3$_H$	CPU resources (memory) occupied.
80C4$_H$	Communication error (occurs temporarily and a repetition in the user program will often remedy the problem).
80D2$_H$	Module start address incorrect.

2.2.3 FC8 AG_UNLOCK

2.2.3.1 Meaning and call - AG_UNLOCK

Meaning of the block

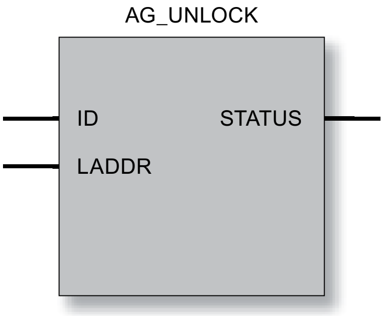
Using the AG_UNLOCK function, the external access to system areas on the S7 CPU with FETCH or WRITE on the connection specified by the ID parameter is released.

The next external FETCH/WRITE job to arrive for the CP can then be processed.

The AG_UNLOCK follows an access lock with AG_LOCK.

Call

Call interface in FBD representation



Example in STL representation

STL	Explanation
call fc 8(	//Block call
ID:=DB 100.DBW 2,	//Connection ID acc. to configuration
LADDR:=W#16#0100,	//=LADDR 256 dec. in HW Config
STATUS:=DB 100.DBW 4);	//Status code

How it works

To release the connection again, the LOCK request bit must be cleared again by the FC. The FC also shows the current status using error messages.

2.2.3.2 Explanation of the formal parameters - AG_UNLOCK

Explanation of the formal parameters

The following table explains all the formal parameters for the AG_UNLOCK function:

Parameter	Declaration	Data type	Possible values	Description
ID	INPUT	INT	1,2...16 for S7300 1,2...64 for S7400	The connection number of the connection is specified in the parameter ID. (See Configuration)
LADDR	INPUT	WORD		Module start address When you configure the CP with STEP 7 HW Config, the module start address is displayed in the configuration table. Specify this address here.
STATUS	OUTPUT	WORD		Status code:

2.2.3.3 Condition codes of AG_UNLOCK

Condition codes

The following table shows the STATUS code that must be evaluated by the user program.

Table 2-3 AG_UNLOCK condition codes

STATUS	Meaning
7000_H	CP is not processing a job
7001_H	FETCH active
7002_H	WRITE active
8183_H	FETCH/WRITE not configured for this connection (S7-400 only)
8186_H	ID number not in permitted range (e.g. 1...64 for S7-400 Industrial Ethernet CPs)
$80A4_H$	The communication bus connection between the CPU and CP is not established. (Only with newer CPU versions)
$80B0_H$	The module does not recognize the data record.
$80B1_H$	The specified length (in the LEN parameter) is incorrect.
$80B2_H$	The communication bus connection between the CPU and CP is not established.
$80C0_H$	The data record cannot be read.
$80C1_H$	The specified data record is currently being processed.
$80C2_H$	There are too many jobs pending.
$80C3_H$	CPU resources (memory) occupied.
$80C4_H$	Communication error (occurs temporarily and a repetition in the user program will often remedy the problem).
$80D2_H$	Module start address incorrect.

2.3 FCs for connection diagnostics

2.3.1 FC 10 AG_CNTRL - meaning and call

How it works

With the AG_CNTRL function, you can diagnose connections. When necessary, you can initialize connection establishment again using the FC.

The following actions are possible by setting commands:

- Reading out connection information

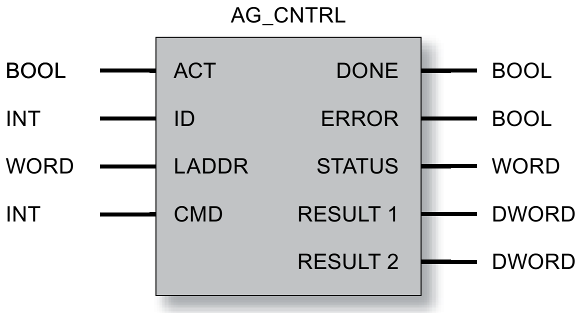
  Based on status information, you can decide whether or not it would be useful to reset all or individual connections of the CP.
- Resetting configured connections

  You can reset individual connections or all connections of a CP.
- Aborting the active connection and establishing it again

The commands of the AG_CNTRL function (FC) are permitted only for SEND/RECV connections based on the ISO / RFC / TCP / UDP protocols.

Call interface

Call interface in FBD representation



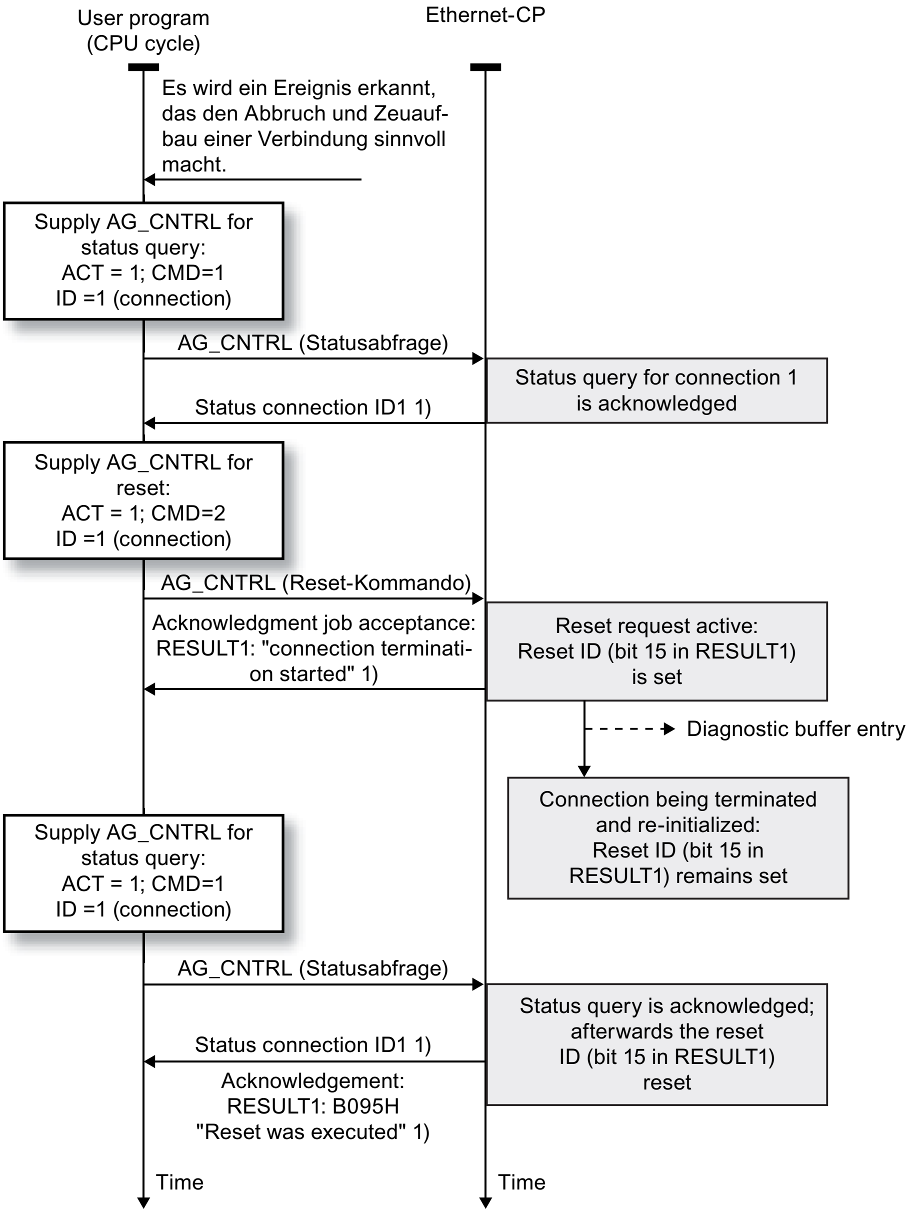
Example in STL representation

STL	Explanation
`call fc 10 (`	`//AG_CNTRL block call`
`ACT:=M1.0,`	`//Job triggered by memory bit`
`ID:=MW8,`	`//Connection ID acc. to configuration`
`LADDR:=W#16#100,`	`//=LADDR 256 dec. in HW Config`
`CMD:=MW6,`	`//=Command ID`
`DONE:=M20.1,`	`//Execution code`
`ERROR:=M20.2,`	`//Error code`
`STATUS:=MW22,`	`//Status code`
`RESULT1:=MD24,`	`//Job result 1`
`RESULT2:=MD28 );`	`//Job result 2`

2.3.2 How AG_CNTRL works

Operating principle

The following diagram shows a typical sequence of AG_CNTRL jobs in the user program.


User program (CPU cycle)
Ethernet-CP
Es wird ein Ereignis erkannt, das den Abbruch und Zeuaufbau einer Verbindung sinnvoll macht.
Supply AG_CNTRL for status query: ACT = 1; CMD=1 ID =1 (connection)
AG_CNTRL (Statusabfrage)
Status query for connection 1 is acknowledged
Status connection ID1 1)
Supply AG_CNTRL for reset: ACT = 1; CMD=2 ID =1 (connection)
AG_CNTRL (Reset-Kommando)
Acknowledgment job acceptance: RESULT1: "connection termination started" 1)
Reset request active: Reset ID (bit 15 in RESULT1) is set
Diagnostic buffer entry
Connection being terminated and re-initialized: Reset ID (bit 15 in RESULT1) remains set
Supply AG_CNTRL for status query: ACT = 1; CMD=1 ID =1 (connection)
AG_CNTRL (Statusabfrage)
Status query is acknowledged; afterwards the reset ID (bit 15 in RESULT1) reset
Status connection ID1 1)
Acknowledgement: RESULT1: B095H "Reset was executed" 1)
Time
Time

1) Parameter transfer DONE, ERROR, STATUS and RESULT1/2

The diagram shows how the connection status is initially queried and then, in a second job, how the connection termination is triggered with the reset command.

The reset ID (bit 15 in RESULT1) is set on the CP. If there is a status query later, it is therefore clearly recognizable whether the connection has been reset due to a reset job. The reset ID is cleared on the CP only after this status query (or as a result of an explicit CN_CLEAR_RESET command)

NOTICE
The block must be called with ACT = 1; if it is called with ACT=0, there is no function call and the block is exited again immediately. Since the job result of FC10 is obtained synchronous to the call, it can be called again in the same cycle.

2.3.3 Explanation of the formal parameters - AG_CNTRL

Explanation of the formal parameters

The following table explains all the formal parameters for the AG_UNLOCK function:

Parameter	Declaration	Data type	Possible values	Description
ACT	INPUT	BOOL	0, 1	The FC must be called with ACT=1. If it is called with ACT=0, there is no function call and the block is exited again immediately.
ID	INPUT	INT	• 1, 2, .., n, or • 0	The connection number of the connection is specified in the parameter ID. The connection number can be found in the configuration. n is the maximum number of connections and is dependent on the product (S7-300 or S7-400). If the call addresses all connections (_ALL function with CMD 3 or 4), 0 must be specified as the ID.
LADDR	INPUT	WORD		Module start address When you configure the CP with STEP 7 HW Config, the module start address is displayed in the configuration table. Specify this address here.
CMD	INPUT	INT		Command to FC AG_CNTRL.
DONE	OUTPUT	BOOL	0: Job still being processed or not yet triggered 1: Job done	This parameter indicates whether or not the job was completed without errors. For the meaning in conjunction with the ERROR and STATUS parameters, refer to the following table. Note: If DONE=1, RESULT can be evaluated
ERROR	OUTPUT	BOOL	0: No error 1: Error	Error code For the meaning in conjunction with the DONE and STATUS parameters, refer to Table 1-6 below.
STATUS	OUTPUT	WORD		Status code For the meaning in conjunction with the DONE and ERROR parameters, refer to Table 1-6 below.

Parameter	Declaration	Data type	Possible values	Description
RESULT1	OUTPUT	DWORD		Information returned according to the command sent to FC AG_CNTRL.
RESULT2	OUTPUT	DWORD		Only to be evaluated for S7-400: Part 2 of information returned according to the command sent to FC AG_CNTRL.

2.3.4 Condition codes of AG_CNTRL

Condition codes

The following table shows the condition codes formed based on DONE, ERROR and STATUS that must be evaluated by the user program.

The command results in the RESULT1/2 parameters must also be evaluated according to " ".

Table 2-4 AG_CNTRL codes

DONE	ERROR	STATUS	Meaning
1	0	0000H	A job (CMD) was transferred successfully to the CP (for example RESET) or a status was read successfully from the CP. The RESULT1/2 parameters can be evaluated.
0	0	0000H	There has been no block call yet or the block was called with ACT=0.
0	0	8181H	Job active The block call must be repeated with the same parameters until DONE or ERROR is signaled.
0	1	8183H	No configuration or the service has not yet started on the Ethernet CP.
0	1	8186H	The ID parameter is invalid. The permitted ID depends on the selected command; see CMD parameter in "".
0	1	8187H	The CMD parameter is invalid.
0	1	8188H	Sequence error in the ACT control (Note: this code does not occur in the product version of the CP / firmware).
0	1	8189H	The CP version / firmware used does not support FC10. The code is set when you call a CP 3431-EX20 with firmware as of V1.3.9; with other CP types, the code 80B0H is set instead. Note: FC10 in version V1.0 is supported by the CPs as of CP 343-1EX21/GX21; this code does not occur with these modules.
0	1	8090H	• No module with this module start address exists. or • The FC being used does not match the system family being used (remember to use different FCs for S7300 and S7400). or • The function is not supported by this module.
0	1	8091H	The module start address is not at a doubleword boundary.

DONE	ERROR	STATUS	Meaning
0	1	80B0$_H$	The module does not recognize the data record.
0	1	80C0$_H$	The data record cannot be read.
0	1	80C1$_H$	The specified data record is currently being processed.
0	1	80C2$_H$	There are too many jobs pending.
0	1	80C3$_H$	CPU resources (memory) occupied.
0	1	80C4$_H$	Communication error The error occurs temporarily; it is usually best to repeat the job in the user program.
0	1	80D2$_H$	The module start address is incorrect.

2.3.5 Commands and job results - AG_CNTRL

Commands and evaluating the job results

The following table shows you the possible commands and the results that can be evaluated in the RESULT1/2 parameters.

Table 2-5 Commands to FC AG_CNTRL

<table>
<tr><th>CMD</th><th colspan="3">Meaning</th></tr>
<tr><td rowspan="5">0</td><td colspan="3">NOP – no operation
The block executes without a job being sent to the CP.</td></tr>
<tr><td colspan="2"><b>RESULT (for CMD = 0)</b></td><td rowspan="2"><b>Meaning</b></td></tr>
<tr><td><b>Parameter</b></td><td><b>Hex value/range</b></td></tr>
<tr><td>RESULT1</td><td>0000 0001$_H$</td><td>Executed without error</td></tr>
<tr><td>RESULT2</td><td>0000 0000$_H$</td><td>Default</td></tr>
</table>

<table>
<tr><th>CMD</th><th colspan="4">Meaning</th></tr>
<tr><td rowspan="5">1</td><td colspan="4">CN_STATUS – connection status
This command returns the status of the connection selected with the ID.
The CP is selected in the LADDR parameter.
If bit 15 (reset ID) is set, this is automatically reset (this action corresponds to the CN_CLEAR_RESET job - see CMD = 5).</td></tr>
<tr><td colspan="3"><b>RESULT (for CMD = 0)</b></td><td rowspan="2"><b>Meaning</b></td></tr>
<tr><td><b>Parameter</b></td><td><b>Hex value/range</b></td><td><b>Bit/value</b></td></tr>
<tr><td rowspan="2">RESULT1</td><td rowspan="2">0000 000*$_H$</td><td colspan="2">Bits 0-3: Codes for the send direction
(excluded values: 0x2)</td></tr>
<tr><td>Bit 0
0
1</td><td>Connection Type
• No send and receive connection
• Connection reserved for send and receive jobs</td></tr>
</table>

CMD	Meaning			
			Bit 1 0 1	Status of current job • No send job being executed • Send job being executed
			Bits 2+3 00 01 10	Previous job: • No information available on previous send job • previous send job completed successfully • previous send job not completed successfully
	RESULT1	0000 00*0$_H$	Bits 4−7: Codes for the receive direction (excluded values: 0x2)	
			Bit 4 0 1	Connection Type • No send and receive connection • Connection reserved for send and receive jobs
			Bit 5 0 1	Status of current job • No receive job being executed • Receive job being executed
			Bits 6+7 00 01 10	Previous job: • No information available on previous receive job • previous receive job completed successfully • previous receive job not completed successfully
	RESULT1	0000 0*00$_H$	Bits 8-11: Codes for FETCH/WRITE (excluded values: 0x3,0x7,0x8,0xB,0xF)	
			Bit 8 0 1	Connection type: • No FETCH connection • Connection reserved for FETCH jobs
			Bit 9 0 1	Connection type: • No WRITE connection • Connection reserved for WRITE jobs
			Bit 10 0 1	Job status (FETCH/WRITE): • Job status OK • Job status NOT OK This ID is set in the following situations: – The job was acknowledged negatively by the CPU – The job could not be forwarded to the CPU because the connection was in the "LOCKED" status. – The job was rejected because the FETCH/WRITE header did not have the correct structure.
			Bit 11 0 1	Status of FETCH/WRITE job • No job active • Job from LAN active
	RESULT1	0000 *000$_H$	Bits 12-15: General CP information (excluded values: 0x3,0xB)	

CMD	Meaning			
			Bit 12 + 13 00 01 10 11	Information on connection status: (only available for SEND/RECV connections based on the ISO/RFC/TCP protocols, with UDP, the corresponding internal information is output) • Connection is terminated • Connection establishment active • Connection termination active • Connection is established
			Bit 14 0 1	CP information: • CP in STOP • CP in RUN
			Bit 15 0 1	Reset ID • FC10 has not yet reset a connection or the reset ID was cleared. • The control block has executed a connection reset.
	RESULT1	**** 0000$_H$		Bits 16-31: Reserved 0 – reserved for later expansions
	RESULT2	0000 0000$_H$		- reserved for later expansions

CMD	Meaning		
2	CN_RESET – connection reset This command resets the connection selected with ID. The CP is selected in the LADDR parameter. Resetting the connection means that a connection is aborted and established again (active or passive depending on the configuration). An entry is also generated in the diagnostics buffer in which the job result can be found.		
	RESULT (for CMD = 2)		**Meaning**
	Parameter	**Hex value/range**	
	RESULT1	0000 0001$_H$	The reset job was transferred to the CP successfully. The connection abort and subsequent connection establishment were triggered.
		0000 0002$_H$	The reset job could not be transferred to the CP because the service has not started on the CP (for example, CP in STOP).
	RESULT2	0000 0000$_H$	Default

<table>
<tr><th>CMD</th><th colspan="3">Meaning</th></tr>
<tr><td rowspan="5">3</td><td colspan="3">CN_STATUS_ALL – all connections status
This command returns the connection status of all connections (established/terminated) in the RESULT1/2 parameters (at total of 8 bytes of group information).
The ID parameter must be set to "0" (checked for 0).
The CP is selected in the LADDR parameter.
When necessary, you can obtain detailed information about a terminated or unconfigured connection using a further connection status call with CMD=1.</td></tr>
<tr><td colspan="2"><b>RESULT (for CMD = 3)</b></td><td rowspan="2"><b>Meaning</b></td></tr>
<tr><td><b>Parameter</b></td><td><b>Hex value/range</b></td></tr>
<tr><td>RESULT1</td><td>**** ****$_H$</td><td>32 bits: Connection 1 - 32
• 0 – connection terminated / not configured
• 1 – connection established</td></tr>
<tr><td>RESULT2</td><td>**** ****$_H$</td><td>32 bits: Connection 33 - 64
• 0 – connection terminated / not configured
• 1 – connection established</td></tr>
</table>

<table>
<tr><th>CMD</th><th colspan="3">Meaning</th></tr>
<tr><td rowspan="6">4</td><td colspan="3">CN_RESET_ALL – all connections reset:
This command resets all connections.
The ID parameter must be set to "0" (checked for 0).
The CP is selected in the LADDR parameter.
Resetting the connections means that connections are aborted and established again (active or passive depending on the configuration).
An entry is also generated in the diagnostics buffer in which the job result can be found.</td></tr>
<tr><td colspan="2"><b>RESULT (for CMD = 4)</b></td><td rowspan="2"><b>Meaning</b></td></tr>
<tr><td><b>Parameter</b></td><td><b>Hex value/range</b></td></tr>
<tr><td>RESULT1</td><td>0000 0001$_H$</td><td>The reset job was transferred to the CP successfully. The connection abort and subsequent connection establishment of all connections were triggered.</td></tr>
<tr><td>RESULT1</td><td>0000 0002$_H$</td><td>The reset job could not be transferred to the CP because the service has not started on the CP (for example, CP in STOP).</td></tr>
<tr><td>RESULT2</td><td>0000 0000$_H$</td><td>Default</td></tr>
</table>

<table>
<tr><th>CMD</th><th colspan="3">Meaning</th></tr>
<tr><td rowspan="3">5</td><td colspan="3">CN_CLEAR_RESET – Clear the reset ID
This command resets the reset ID (bit 15 in RESULT1) for the connection selected with ID.
The CP is selected in the LADDR parameter.
This job executes automatically when the connection status is read (CMD=1); the separate job described here is therefore only required in special situations.</td></tr>
<tr><td colspan="2"><b>RESULT (for CMD = 5)</b></td><td rowspan="2"><b>Meaning</b></td></tr>
<tr><td><b>Parameter</b></td><td><b>Hex value/range</b></td></tr>
</table>

CMD	Meaning		
	RESULT1	0000 0001$_H$	The clear job was transferred to the CP successfully.
	RESULT1	0000 0002$_H$	The Clear job could not be transferred to the CP because the service has not started on the CP (for example, CP in STOP).
	RESULT2	0000 0000$_H$	Default

CMD	Meaning		
6	CN_DISCON − connection disconnect This command resets the connection selected with ID and LADDR. Resetting the connection is achieved by aborting the connection. Any data in the stack is lost without any message being displayed. The connection is not established again automatically afterwards. The connection can be established again with the CN_STARTCON control job. A diagnostics buffer entry is created in which you will find the job result.		
	RESULT (for CMD = 6)		**Meaning**
	Parameter	**Hex value/range**	
	RESULT1	0000 0001$_H$	The job was transferred to the CP successfully. The connection abort was initiated.
	RESULT1	0000 0002$_H$	The job could not be transferred to the CP because the service has not started on the CP (for example, CP in STOP).
	RESULT2	0000 0000$_H$	Default

CMD	Meaning		
7	CN_STARTCON − start connection This command establishes a connection selected with ID and LADDR and aborted earlier with the control job CN_DISCON. A diagnostics buffer entry is created in which you will find the job result.		
	RESULT (for CMD = 6)		**Meaning**
	Parameter	**Hex value/range**	
	RESULT1	0000 0001$_H$	The connection establishment job was transferred to the CP successfully. The connection establishment was initiated.
	RESULT1	0000 0002$_H$	The connection establishment job could not be transferred to the CP because the service has not started on the CP (for example, CP in STOP).
	RESULT2	0000 0000$_H$	Default

2.4 FBs / FCs for FTP services

2.4.1 Overview of FTP

FB and FCs for FTP services (FTP client)

The following list shows the blocks available for the FTP client services. You can change the block numbers used when the blocks ship.

NOTICE
Note that the FTP client services of old SIMATIC S7-300 CPUs, for example the CPU 312 or CPU 315-1AF01, cannot be used because they do not support SFC24. In S7-300 the CP requires the supplied FC5 (AG_SEND) to start the FTP FCs; do not rename the FC5 for this application.

FB/FC	can be used with:		Meaning
	S7-300	S7-400	
FB40	x	x	Handling of complete FTP job sequences
FC40	x	x	Establishing an FTP connection from client to server.
FC41	x	x	Transferring a DB from client to server.
FC42	x	x	Transferring a file from client to server.
FC43	x	x	Deleting a file on the server.
FC44	x	x	Terminating a connection established with the ID.

Using the blocks

Use either FB40 or the functions FC40...44 for the data transfer depending on the module type.

- FB40

  FB40 can be used as of the following module types:

  - As of CP 343-1 Advanced (GX30)
  - As of CP 443-1 Advanced (GX20)

    These module types continue to support the functions FC40...44.

    The following module types, on the other hand, do not support FB40:

  - IT / Advanced CPs up to CP 343-1 Advanced (GX21)
  - IT / Advanced CPs up to CP 443-1 Advanced (EX41)

- FC40...44

  The functions FC40...44 can be used with all Advanced CPs.

Requirement - configured FTP connection

To manage an FTP request sequence between the S7 station as FTP client and an FTP server, you will need to configure an FTP connection. To do this, first configure an unspecified TCP connection with the additional "for FTP protocol" attribute.

See also

Migration from FC 40-44 to FB40 (Page 64)

2.4.2 FB40 FTP_CMD - universal block for FTP services

2.4.2.1 Meaning and call - FTP_CMD

Meaning

Using FB40, you can establish FTP connections and transfer files from and to an FTP server.

FB40 replaces the FTP functions FC40 to FC44 used previously. The differences in these functions are mapped to a command parameter in FB40.

The advantages of FB40 are as follows:

- Simplification in the user program by using a command variable instead of different function calls
- Additional function "APPEND"

  "APPEND" allows data to be appended to an existing file.
- Additional function "RETR_PART"

  "RETR_PART" allows selected data areas to be retrieved from a file.

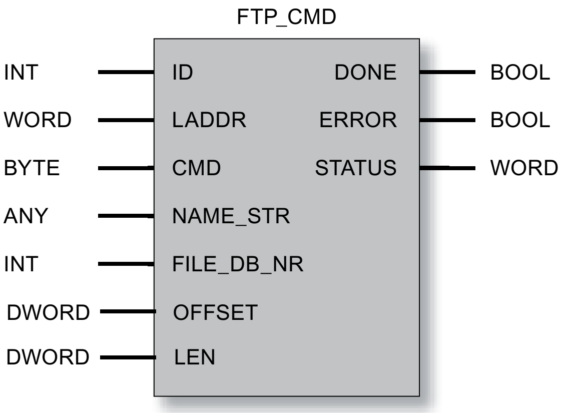
Validity

FB40 can be used as of the following module types:

- As of CP 343-1 Advanced (GX30)
- As of CP 443-1 Advanced (GX20)

Call interface

Call interface in FBD representation



Example of a call in STL representation

STL	Explanation
`CALL FB 40, DB 40 (`	`// Block call`
`ID:= 4,`	`// FTP connection ID according to configuration`
`LADDR := W#16#3FFD,`	`// Module address according to configuration`
`CMD:= B#16#3,`	`// The FTP command to be executed`
`NAME_STR := P#DB44.DBX 170.0 BYTE 220,`	`// Address and length of the destination data area`
`FILE_DB_NR:= 42,`	`// Number of the data block`
`OFFSET:= DW#16#0,`	`// (not relevant in the example)`
`LEN:= DW#16#0,`	`// (not relevant in the example)`
`DONE:= M 420.1,`	`// Status parameter`
`ERROR:= M 420.2,`	`// Error code`
`STATUS:= MW 422);`	`// Status code`

System functions called

The following system functions are called by FB40:

SFC 1, SFC 20, SFC 24, SFC 58, SFC 59

NOTICE
Note that the FTP client services of old SIMATIC S7-300 CPUs, for example the CPU 312 or CPU 315-1AF01, cannot be used because they do not support SFC24.

2.4.2.2 Input parameter - FTP_CMD

Explanation of the input parameters

Each FTP block call must be supplied with the following input parameters:

Table 2-6 Formal parameters of FB40 (FTP_CMD) - input parameters

Parameter	Declaration	Type	Possible values	Meaning / remarks
ID	INPUT	INT	1, 2...64	The FTP jobs are handled on FTP connections. The parameter identifies the connection being used.
LADDR	INPUT	WORD		Module start address When you call an FC, you transfer the module start address of the ADVANCED-CP in the LADDR parameter. You will find the module start address of the ADVANCED CP in the configuration of the ADVANCED CP in "Properties>Addresses>Inputs".
CMD	INPUT	BYTE	0: NOOP 1: CONNECT 2: STORE 3: RETRIEVE 4: DELETE 5: QUIT 6: APPEND 7: RETR_PART	FTP commands executed when FB40 is called. You will find further information following the table. If a command is not supported by the CP firmware, an error message with STATUS = 8F6B$_H$ is output.
NAME_STR	INPUT	ANY	Only "BYTE" is permitted as VARTYPE:	The address references a data block area. Here, you specify the address and length of the data area in which the target data is entered. • When CMD = 1: With this command, the "NAME_STR" parameter specifies the FTP server to be addressed over the FTP connection with the following attributes: - IP address of the FTP server - User name - Password for the login These values must be specified as consecutive strings in the destination range of the ANY pointer. • When CMD = 2, 3, 4, 6, 7: With this command, the "NAME_STR" parameter specifies the file name on the FTP server, in other words, the data source or data destination. The file name is specified as a string in the destination range of the ANY pointer. • When CMD = 5: Parameter not relevant You will find example of content further below.

Parameter	Declaration	Type	Possible values	Meaning / remarks
FILE_DB_NR	INPUT	INT		The data block specified here contains the file DB to be read / written. The parameter is relevant only when CMD = 2, 3, 6 and 7.
OFFSET	INPUT	DWORD		Only when CMD = 7: Offset in bytes starting at which the file will be read.
LEN	INPUT	DWORD		Only when CMD = 7: Sublength in bytes that is read starting at the value specified in "OFFSET". Special features: • If "DW#16#FFFFFFFF" is specified, the available rest of the file will be read. Result OK (DONE = 1, STATUS = 0) if no other error occurred. • When OFFSET > length of the original file: Length of the destination file (ACT_LENGTH in file DB): 0 bytes on the CPU. Result OK (DONE = 1, STATUS = 0) if no other error occurred. • When OFFSET + LEN > length of the original file (and LEN ≠ 0xFFFFFFFF): Length of the destination file (ACT_LENGTH in file DB): Available bytes starting at "OFFSET". Result OK (DONE = 1, STATUS = 0) if no other error occurred.

FTP commands in the "CMD" parameter

The following table explains the meaning of the commands of the "CMD" parameter and which input parameters need to be supplied. The ID and LADDR parameters must always be set to identify the connection.

CMD	Relevant input parameters (in addition to ID and LADDR)	Meaning / handling
0 (NOOP)	-	The called FC does not execute any action. The status codes are set as follows when these parameters are supplied: • DONE=1; ERROR=0; STATUS=0
1 (CONNECT)	NAME_STR	With this command, the FTP client establishes an FTP connection to an FTP server. The connection is available under the connection ID specified here for all further FTP commands. Data is then exchanged with the FTP server specified for this user.
2 (STORE)	NAME_STR FILE_DB_NR	This function call transfers a data block (file DB) from the FTP client (S7-CPU) to the FTP server. Caution: If the file (file DB) already exists on the FTP server, it will be overwritten.

CMD	Relevant input parameters (in addition to ID and LADDR)	Meaning / handling
3 (RETRIEVE)	NAME_STR FILE_DB_NR	This function call transfers a file from the FTP server to the FTP client (S7-CPU). Caution: If the data block (file DB) on the FTP client already contains a file, it will be overwritten.
4 (DELETE)	NAME_STR	With this function call, you delete a file on the FTP server.
5 (QUIT)	No others	With this function call, you establish the FTP connections selected with the ID.
6 (APPEND)	NAME_STR FILE_DB_NR	Similar to "STORE", the "APPEND" command saves a file on the FTP server. With "APPEND", the file on the FTP server is, however, not overwritten. The new content is appended to the existing file.
7 (RETR_PART)	NAME_STR FILE_DB_NR OFFSET LEN	Using the "RETR_PART" command (retrieve part) , you can request a section of a file from the FTP server. If very large files are involved, this allows you to restrict the read to the part you currently require. To do this, you need to know the structure of the file. Enter the required part of the file using the two parameters "OFFSET" and "LEN" in FB40.

Examples of the content of the "NAME_STR" parameter

The parameter record has the following content:

Table 2-7 Content of the parameter record for CMD = 1

Relative address [2)]	Name	Type [1)]	Example	Meaning
0.0	ip_address	STRING[100]	’142.11.25.135’	IP address of the FTP server
102.0	username	STRING[32]	'user'	User name for the login on the FTP server
136.0	password	STRING[32]	'password'	Password for the login on the FTP server
1) The maximum possible string length is specified 2) The specified values relate to the string lengths specified in "Type".				

Table 2-8 Content of the parameter record for CMD = 2, 3, 4, 6, 7

Relative address [2)]	Name	Type [1)]	Example	Meaning
170.0	filename	STRING[220]	'plant1/tank2/press.dat'	File name of the source or destination file
1) The maximum possible string length is specified 2) The specified values relate to the string lengths specified in "Type".				

2.4.2.3 Output parameters and status information - FTP_CMD

Introduction

For status evaluation, the following parameters must be evaluated in the user program:

Table 2-9 Formal parameters of FB40 (FTP_CMD) - output parameters

Parameter	Declaration	Type	Possible values	Meaning / remarks
DONE	OUTPUT	BOOL	0: - 1: Job executed	This parameter indicates whether or not the job was completed without errors.
ERROR	OUTPUT	BOOL	0: - 1: Error	Error code This parameter signals that the job could not be executed error-free.
STATUS	OUTPUT	WORD	See following table	Status code This parameter supplies detailed information about the execution of the job.

The DONE, ERROR and STATUS parameters are updated at every block call.

Example

During job execution, the FB40 returns the following codes:

- DONE=0
- ERROR=0
- STATUS=8181$_H$

Significance: Job still running.

Evaluating status codes

Note

For entries coded with 8FxxH in STATUS, refer to the information in the STEP 7 Standard and System Functions reference manual. The chapter describing error evaluation with the RET_VAL output parameter contains detailed information.

Table 2-10 FB 40: Meaning of the STATUS parameter in conjunction with DONE and ERROR

DONE	ERROR	STATUS	Meaning
0	0	0000$_H$	No job being executed.
1	0	0000$_H$	Job completed without error.

DONE	ERROR	STATUS	Meaning
0	0	8181$_H$	Job active. If 8181$_H$ is indicated permanently: The CP is not released for FB40 (an illegal command for the firmware version (CMD 6 or CMD 7) was called.)
0	1	8090$_H$	• No module with this module start address exists. • The block being used does not match the system family being used (remember to use different blocks for S7300 and S7400).
0	1	8091$_H$	Module start address not at a doubleword boundary
0	1	8092$_H$	Type information in the ANY pointer is not byte
0	1	80A4$_H$	The communication bus connection between the CPU and CP is not established (with newer CPU versions). This can, for example, be caused by the following: • No connection configuration • Maximum number of CPs operating at the same time was exceeded
0	1	80B0$_H$	The module does not recognize the data record.
0	1	80B1$_H$	Destination area invalid; for example, destination area > 240 bytes.
0	1	80B2$_H$	The communication bus connection between the CPU and CP is not established (with older CPU versions). (with newer CPU versions, see 80A4$_H$)
0	1	80C0$_H$	The data record cannot be read.
0	1	80C1$_H$	The specified data record is currently being processed.
0	1	80C2$_H$	There are too many jobs pending.
0	1	80C3$_H$	Resources occupied (memory).
0	1	80C4$_H$	Communication error (occurs temporarily, it is usually best to repeat the job in the user program).
0	1	80D2$_H$	Module start address incorrect.
0	1	8183$_H$	The configuration does not match the job parameters.
0	1	8184$_H$	Illegal data type specified for the NAME_STR parameter.
0	1	8186$_H$	ID parameter invalid. ID = 1, 2....64
0	1	8F22$_H$	Source area invalid, for example: Area does not exist in the DB
0	1	8F24$_H$	Area error reading a parameter
0	1	8F28$_H$	Alignment error reading a parameter
0	1	8F32$_H$	Parameter contains a DB number that is too high.
0	1	8F33$_H$	DB number error
0	1	8F3A$_H$	Area not loaded (DB)
0	1	8F50$_H$	File DB DB 0 or DB does not exist
0	1	8F51$_H$	Specified file DB data area larger than existing data area
0	1	8F52$_H$	File DB in write-protected memory
0	1	8F53$_H$	File DB max. length < current length
0	1	8F54$_H$	File DB does not contain any valid data.
0	1	8F55$_H$	Header status bit: Locked
0	1	8F56$_H$	The NEW bit in the file DB header was not reset

DONE	ERROR	STATUS	Meaning
0	1	8F57$_H$	The FTP client does not have write access to the file DB but rather the FTP server (header status bit: WriteAccess).
0	1	8F60$_H$	Bad user data, for example bad IP address of the FTP server
0	1	8F61$_H$	FTP server not obtainable
0	1	8F62$_H$	Job not supported or rejected by FTP server
0	1	8F63$_H$	File transfer aborted by the FTP server
0	1	8F64$_H$	Error on the FTP control connection; data could not be sent or received; the FTP control connection must be established again after such an error.
0	1	8F65$_H$	Error on the FTP data connection; data could not be sent or received. The job must be called again. This error can, for example, be caused by RETRIEVE (CMD=3) when the addressed file is already open on the FTP server.
0	1	8F66$_H$	Error reading/writing data from/to the CPU (for example DB does not exist or too short)
0	1	8F67$_H$	Error in the FTP client on the ADVANCED CP; for example attempting to open more than 10 FTP connections.
0	1	8F68$_H$	The job was rejected by the FTP client. This error can, for example, be caused by RETRIEVE (CMD=3) when the value for the parameter MAX_LENGTH was selected too low in the file DB header.
0	1	8F69$_H$	The FTP connection in an incorrect status, for example: • The connection is called without a previous connection termination (with the same NetPro ID) • There is a connection termination for a connection that has already been terminated: • A STORE command was sent on a connection that is not established.
0	1	8F6A$_H$	No new socket could be opened / temporary resource problem: Repeat the block call.
0	1	8F6B$_H$	Possible causes: • Bad value for the CMD parameter Values from 0 to 15 are permitted. • An FB40 command is not supported. Possible cause: Wrong firmware on the CP Remedy: Firmware update (with older CPs, use the functions FC 40...FC 44 instead of FB 40.)
0	1	8F7F$_H$	Internal error, for example illegal ANY reference

2.4.2.4 Migration from FC 40-44 to FB40

Comparison of the function block FB40 with older functions FC40...44

All CPs with FTP functionality support the functions FC40...44. This means that existing user programs can be used unchanged.

If you want to convert from the FTP functions FC40...44 to FB40, you will need to modify your user program.

The following table shows the FB40 commands used to replace the functions FC40...44.

- Correlation is indicated by "X".
- Where there is no correlation, this is indicated by "-".

	Commands of the "CMD" parameter of FB40						
Older FTP functions FC40...44	**CMD = 1**	**CMD = 2**	**CMD = 3**	**CMD = 4**	**CMD = 5**	**CMD = 6**	**CMD = 7**
FC40	X [1]						
FC41		X [2]					
FC42			X [3]				
FC43				X [4]			
FC44					X		
						-	
							-
[1]...[4] The parameters of FC40...43 and CMD 1...4 (FB40) are not identical. (See table below)							

The corresponding parameters that specify a particular function in the functions FC40...FC43 or in the commands of FB40 are listed in the following table.

Parameters of the FC			Parameters in FB40 (with CMD 1...4)	
FC40:	LOGIN	→	CMD = 1:	NAME_STR
FC41:	FILE_NAME	→	CMD = 2:	NAME_STR
FC42:	FILE_NAME	→	CMD = 3:	NAME_STR
FC43:	FILE_NAME	→	CMD = 4:	NAME_STR
FC40...43:	BUFFER_DB_NR	→	Omitted (replaced by instance DB)	

2.4.3 FC40 FTP_CONNECT

2.4.3.1 Meaning and call - FTP_CONNECT

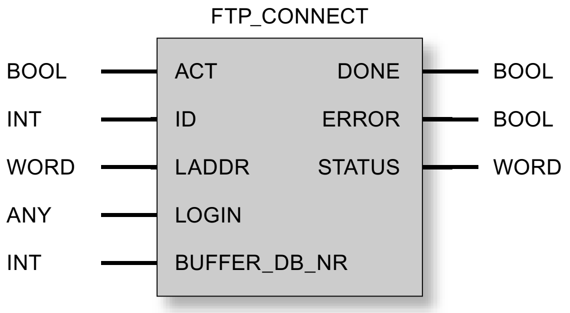
Meaning

With this function call, the FTP client establishes an FTP connection to an FTP server.

The IP address of the FTP server, the user identification (username) and (if necessary) the password for user identification must be transferred to the FTP server.

All further access by the FTP client is then based on this user identification if you use the same FTP connection ID. Data is then exchanged with the FTP server specified for this user.

Call interface



Example of a call in STL representation

STL	Explanation
`call fc40 (`	`//FTP_CONNECT block call`
`ACT:= M 420.0,`	`// Job triggered by memory bit`
`ID:= 4,`	`// FTP connection ID acc. to configuration`
`LADDR:= W#16#3FFD,`	`// Module address acc. to configuration`
`LOGIN := P#DB40.DBX 0.0 BYTE 170,`	`// Information for LOGIN in DB40`
`BUFFER_DB_NR:= 9,`	`// Buffer area for FTP service`
`DONE:= M 420.1,`	
`ERROR:= M 420.2,`	
`STATUS:= MW 422);`	

2.4.3.2 Explanation of the formal parameters - FTP_CONNECT

Explanation of the general call parameters

The general parameters have the same significance in every FTP function call; they are therefore described in one section.

Explanation of the formal parameters specific to the call

Table 2-11 Formal parameters for FTP_CONNECT

Parameter	Declaration	Data type	Description
LOGIN	INPUT	ANY (only the following are permitted as VARTYPE: BYTE)	This parameter specifies the FTP server to be accessed on the FTP connection. (for further details, refer to the following table) Here, you specify the address and length of the data area in which the target data are entered. The address references a data block area. The ANY pointer data type is used to address this area. For more detailed information on this data type, refer to the STEP 7 online help under the appendix topic "Format of the parameter type ANY".
BUFFER_DB_NR	INPUT	INT	Here, you enter a data block required as a buffer area by the FTP client for FTP transfer. You can use the same data block as the buffer area for all FTP jobs. Note: The length of the reserved DB must be **at least 255 bytes**!

LOGIN parameter

This parameter record has the following content for FTP_CONNECT:

Relative address [2]	Name	Type [1]	Example	Meaning
0.0	ip_address	STRING[100]	'142.11.25.135'	IP address of the FTP server.
102.0	username	STRING[32]	'user'	User name for the login on the FTP server.
136.0	password	STRING[32]	'password'	Password for the login on the FTP server.
170.0	filename	STRING[220]	'plant1/tank2/pressure.dat'	Name of the destination or source file

[1] in each case, the maximum possible string length is specified

[2] The specified values relate to the string lengths specified in "Type".
Note: The rows shown on a gray background are irrelevant for this call.

2.4.4 FC41 FTP_STORE

2.4.4.1 Meaning and call - FTP_STORE

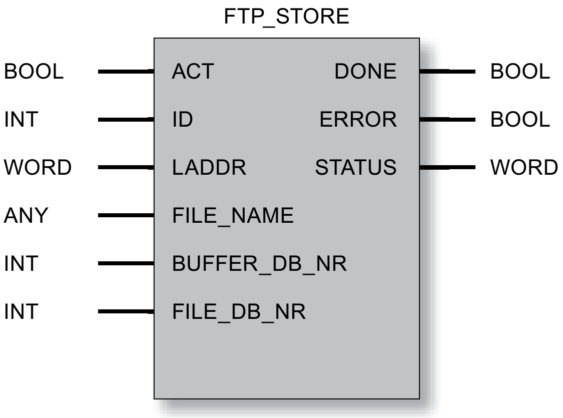
Meaning

This function call transfers a data block (file DB) from the FTP client (S7-CPU) to the FTP server.

You need to specify the data block that contains the file. You will also have to select the path/file name under which the file will be created on the FTP server.

If the file (file DB) already exists on the FTP server, it will be overwritten.

Call interface



Example of a call in STL representation

STL	Explanation
call fc41 (	//FTP_STORE block call
ACT:= M 420.0,	// Job triggered by memory bit
ID:= 4,LADDR:= W#16#3FFD,	// FTP connection ID acc. to configuration
FILE_NAME := P#DB40.DBX 170.0 BYTE 220,	// Module address acc. to configuration
BUFFER_DB_NR:= 9,	// Information for destination file in DB40
FILE_DB_NR:= 42,	// Buffer area for FTP service
DONE:= M 420.1,	// DB no. of source file
ERROR:= M 420.2,	
STATUS:= MW 422);	

2.4.4.2 Explanation of the formal parameters - FTP_STORE

Explanation of the general call parameters

The general parameters have the same significance in every FTP function call; they are therefore described in one section.

Explanation of the formal parameters specific to the call

Table 2-12 Formal parameters for FTP_STORE

Parameter	Declaration	Data type	Description
FILE_NAME	INPUT	ANY (only the following are permitted as VARTYPE: BYTE)	This parameter specifies the data destination. (for further details, refer to the following table) Here, you specify the address and length of the data area in which the target data are entered. The address references a data block area. The ANY pointer data type is used to address this area. For more detailed information on this data type, refer to the STEP 7 online help under the appendix topic "Format of the parameter type ANY".
BUFFER_DB_ NR	INPUT	INT	Here, you enter a data block required as a buffer area by the FTP client for FTP transfer. You can use the same data block as the buffer area for all FTP jobs. Note: The length of the reserved DB must be at least 255 bytes!
FILE_DB_NR	INPUT	INT	The data block specified here contains the file DB to be read.

Parameter FILE_NAME

This parameter record has the following content for FTP_STORE:

Relative address [2]	Name	Type [1]	Example	Meaning
0.0	ip_address	STRING[100]	'142.11.25.135'	IP address of the FTP server.
102.0	username	STRING[32]	'user'	User name for the login on the FTP server.
136.0	password	STRING[32]	'password'	Password for the login on the FTP server.
170.0	filename	STRING[220]	'plant1/tank2/pressure.dat'	Name of the destination or source file

[1] in each case, the maximum possible string length is specified

[2] The specified values relate to the string lengths specified in "Type".
Note: The rows shown on a gray background are irrelevant for this call.

2.4.5 FC42 FTP_RETRIEVE

2.4.5.1 Meaning and call - FTP_RETRIEVE

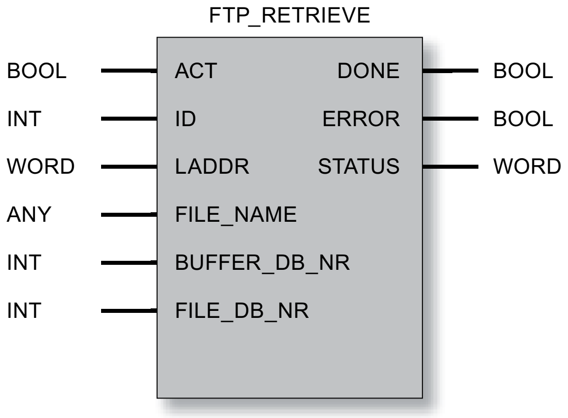
Meaning

This function call transfers a file from the FTP server to the FTP client (S7-CPU).

You need to specify the data block in which the file will be entered. You will also have to select the path/file name under which the file is located on the FTP server.

If the data block (file DB) on the FTP client already contains a file, it will be overwritten.

Call interface



Example of a call in STL representation

STL	Explanation
`call fc42 (`	`//FTP_RETRIEVE block call`
`ACT:= M 420.0,`	`// Job triggered by memory bit`
`ID:= 4,`	`// FTP connection ID acc. to configuration`
`LADDR:= W#16#3FFD,`	`// Module address acc. to configuration`
`FILE_NAME := P#DB40.DBX 170.0 BYTE 220,`	`// Information for source file in DB40`
`BUFFER_DB_NR:= 9,`	`// Buffer area for FTP service`
`FILE_DB_NR:= 42,`	`// DB no. of destination file`
`DONE:= M 420.1,`	
`ERROR:= M 420.2,`	
`STATUS:= MW 422);`	

2.4.5.2 Explanation of the formal parameters - FTP_RETRIEVE

Explanation of the general call parameters

The general parameters have the same significance in every FTP function call; they are therefore described in one section.

Explanation of the formal parameters specific to the call

Table 2-13 Formal parameters for FTP_RETRIEVE

Parameter	Declaration	Data type	Description
FILE_NAME	INPUT	ANY (only the following are permitted as VARTYPE: BYTE)	This parameter specifies the data source. (for further details, refer to the following table) Here, you specify the address and length of the data area in which the target data are entered. The address references a data block area. The ANY pointer data type is used to address this area. For more detailed information on this data type, refer to the STEP 7 online help under the appendix topic "Format of the parameter type ANY".
BUFFER_DB_ NR	INPUT	INT	Here, you enter a data block required as a buffer area by the FTP client for FTP transfer. You can use the same data block as the buffer area for all FTP jobs. Note: The length of the reserved DB must be **at least 255 bytes**!
FILE_DB_NR	INPUT	INT	The data block specified here contains the file DB to be written (data destination).

FILE_NAME parameter

This parameter record has the following content for FTP_RETRIEVE:

Relative address 2)	Name	Type 1)	Example	Meaning
0.0	ip_address	STRING[100]	'142.11.25.135'	IP address of the FTP server.
102.0	username	STRING[32]	'user'	User name for the login on the FTP server.
136.0	password	STRING[32]	'password'	Password for the login on the FTP server.
170.0	filename	STRING[220]	'plant1/tank2/pressure.dat'	Name of the destination or source file

1) in each case, the maximum possible string length is specified

2) The specified values relate to the string lengths specified in "Type".
Note: The rows shown on a gray background are irrelevant for this call.

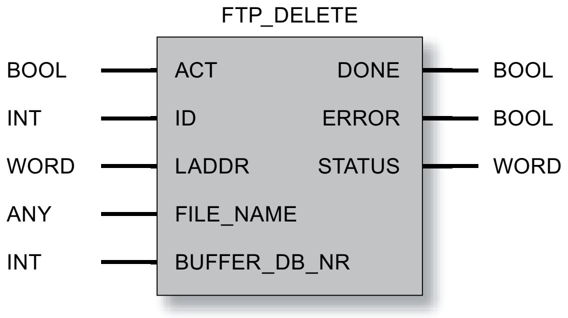
2.4.6 FC43 FTP_DELETE

2.4.6.1 Meaning and call - FTP_DELETE

Meaning

With this function call, you delete a file on the FTP server.

Call interface



Example of a call in STL representation

STL	Explanation
`call fc43 (`	`//FTP_DELETE block call`
`ACT:= M 420.0,`	`// Job triggered by memory bit`
`ID:= 4,`	`// FTP connection ID acc. to configuration`
`LADDR:= W#16#3FFD,`	`// Module address acc. to configuration`
`FILE_NAME := P#DB40.DBX 170.0 BYTE 220,`	`// Information for destination file in DB40`
`BUFFER_DB_NR:= 9,`	`// Buffer area for FTP service`
`DONE:= M 420.1,`	
`ERROR:= M 420.2,`	
`STATUS:= MW 422);`	

2.4.6.2 Explanation of the formal parameter - FTP_DELETE

Explanation of the general call parameters

The general parameters have the same significance in every FTP function call; they are therefore described in one section.

Explanation of the formal parameters specific to the call

Table 2-14 Formal parameters for FTP_DELETE

Parameter	Declaration	Data type	Description
FILE_NAME	INPUT	ANY (only the following are permitted as VARTYPE: BYTE)	This parameter specifies the data destination. (for further details, refer to the following table) Here, you specify the address and length of the data area in which the target data are entered. The address references a data block area. The ANY pointer data type is used to address this area. For more detailed information on this data type, refer to the STEP 7 online help under the appendix topic "Format of the parameter type ANY".
BUFFER_DB_NR	INPUT	INT	Here, you enter a data block required as a buffer area by the FTP client for FTP transfer. You can use the same data block as the buffer area for all FTP jobs. Note: The length of the reserved DB must be **at least 255 bytes**!

LOGIN parameter

This parameter record has the following content for FTP_DELETE:

Relative address [2]	Name	Type [1]	Example	Meaning
0.0	ip_address	STRING[100]	'142.11.25.135'	IP address of the FTP server.
102.0	username	STRING[32]	'user'	User name for the login on the FTP server.
136.0	password	STRING[32]	'password'	Password for the login on the FTP server.
170.0	filename	STRING[220]	'plant1/tank2/pressure.dat'	Name of the destination or source file

[1] in each case, the maximum possible string length is specified

[2] The specified values relate to the string lengths specified in "Type".
Note: The rows shown on a gray background are irrelevant for this call.

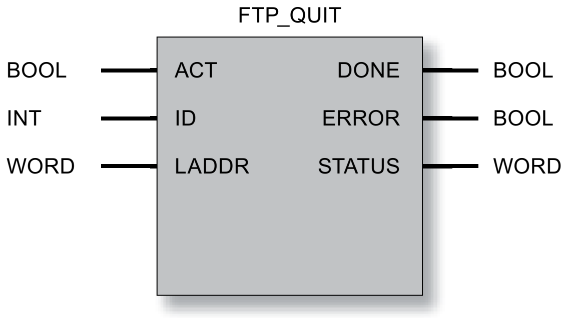
2.4.7 FC44 FTP_QUIT

2.4.7.1 Meaning and call - FTP_QUIT

Meaning

With this function call, you establish the FTP connections selected with the ID.

Call interface



Example of a call in STL representation

STL	Explanation
`call fc44 (`	`//FTP_QUIT block call`
`ACT:= M 420.0,`	`// Job triggered by memory bit`
`ID:= 4,`	`// FTP connection ID acc. to configuration`
`LADDR:= W#16#3FFD,`	`// Module address acc. to configuration`
`DONE:= M 420.1,`	
`ERROR:= M 420.2,`	
`STATUS:= MW 422);`	

NOTICE

The output of FC44 must be assigned a memory word as value. If you enter DBx.DWy, an error message is displayed (applies only to S7-300).

2.4.7.2 Explanation of the formal parameters - FTP_QUIT

Explanation of the general call parameters

The general parameters have the same significance in every FTP function call; they are therefore described in one section.

2.4.8 Parameters for CP and connection assignment (input parameters)

Parameters for CP and connection assignment (input parameters)

Apart from the input parameters specific to the jobs started with each FTP block call, the following general input parameters must also have values supplied to them:

Parameter	Declaration	Type [1)]	Possible values	Meaning
ACT	INPUT	BOOL	0,1	The parameter contains the initialization bit for triggering the job. • If ACT = 1, the job is executed. During job execution, the FC returns the following codes: – DONE=0 – ERROR=0 – STATUS=8181$_H$ • If ACT = 0, the called FC does not execute any actions; the status codes are then as follows for these parameters settings: – DONE=0 – ERROR=1 – STATUS=8F70$_H$ Note / recommendation: You should execute the FTP calls conditionally in your application, for example by evaluating the codes. It is not a good idea to control the call using the ACT bit. The ACT bit must be set to 1 until the execution is signaled by the DONE bit.
ID	INPUT	INT	1,2...64	The FTP jobs are handled on FTP connections. The parameter identifies the connection being used.
LADDR	INPUT	WORD		Module start address When you call an FC, you transfer the module start address of the ADVANCED-CP in the LADDR parameter. You will find the module start address of the ADVANCED-CP in the configuration of the properties of the ADVANCED-CP in "Addresses > Inputs".

CAUTION

Make sure that only one FTP client block is called per ID used as long as ACT = 1 is set.

For example, FC STORE and FC RETRIEVE must not be active on the same FTP connection at the same time. This corresponds to normal FTP functionality. If this is nevertheless attempted, you cannot rely on the output parameters (DONE bit, ERROR bit and STATUS word) being correct.

2.4.9 Status information (output parameters)

Status information (output parameters)

For status evaluation, the following parameters must be evaluated in the user program:

Parameter	Declaration	Type [1)]	Possible values	Meaning
DONE	OUTPUT	BOOL	0: - 1: Job done	This parameter indicates whether or not the job was completed without errors.
ERROR	OUTPUT	BOOL	0: - 1: Error	Error code This parameter signals that the job could not be executed error-free.
STATUS	OUTPUT	WORD	See following table	Status code This parameter supplies detailed information about the execution of the job.

NOTICE
For FC FTP_QUIT, use only the data type memory word for the STATUS parameter (applies only to the CP 343-1 IT).

Example

During job execution, the FC returns the following codes:

- DONE=0
- ERROR=0
- STATUS=8181$_H$

Evaluating status codes

Remember that the status codes DONE, ERROR, STATUS are updated at each block call.

Note

For entries coded with 8FxxH in STATUS, refer to the information in the STEP 7 Standard and System Functions reference manual. The chapter describing error evaluation with the RET_VAL output parameter contains detailed information

DONE	ERROR	STATUS	Meaning
1	0	0000$_H$	Job completed without error.
0	0	0000$_H$	No job being executed.
0	0	8181$_H$	Job active.
0	1	8090$_H$	• No module with this module start address exists. • The FC being used does not match the system family being used (remember to use different FCs for S7300 and S7400).
0	1	8091$_H$	Module start address not at a doubleword boundary.
0	1	8092$_H$	Type information in the ANY pointer is not byte.
0	1	80A4$_H$	The communication bus connection between the CPU and CP is not established. (with newer CPU versions). This can, for example, be caused by the following: • No connection configuration; • The maximum number of CPs that can be operated at the same time has been exceeded.
0	1	80B0$_H$	The module does not recognize the data record.
0	1	80B1$_H$	• Destination area invalid. for example, destination area > 240 bytes.
0	1	80B2$_H$	The communication bus connection between the CPU and CP is not established (with older CPU versions; otherwise 80A4$_H$; for further information, refer to this code)
0	1	80C0$_H$	The data record cannot be read.
0	1	80C1$_H$	The specified data record is currently being processed.
0	1	80C2$_H$	There are too many jobs pending.
0	1	80C3$_H$	Resources occupied (memory).
0	1	80C4$_H$	Communication error (occurs temporarily, it is usually best to repeat the job in the user program).
0	1	80D2$_H$	Module start address incorrect.
0	1	8183$_H$	The configuration does not match the job parameters.
0	1	8184$_H$	• Bad data type specified for the FILE_NAME / LOGIN parameter.
0	1	8186$_H$	ID parameter invalid. ID != 1,2....64.
0	1	8F22$_H$	Source area invalid, for example: Area does not exist in the DB
0	1	8F24$_H$	Area error reading a parameter.
0	1	8F28$_H$	Alignment error reading a parameter.
0	1	8F32$_H$	Parameter contains a DB number that is too high.
0	1	8F33$_H$	DB number error.
0	1	8F3A$_H$	Area not loaded (DB).
0	1	8F50$_H$	File DB DB 0 or DB does not exist
0	1	8F51$_H$	Specified file DB data area larger than existing data area
0	1	8F52$_H$	File DB in write-protected memory
0	1	8F53$_H$	File DB max. length < current length
0	1	8F54$_H$	File DB does not contain any valid data
0	1	8F55$_H$	Header status bit: Locked
0	1	8F56$_H$	The NEW bit in the file DB header was not reset
0	1	8F57$_H$	The FTP client does not have write access to the file DB but rather the FTP server (header status bit: WriteAccess)

DONE	ERROR	STATUS	Meaning
0	1	8F5A$_H$	Buffer DB DB 0 or DB does not exist
0	1	8F5B$_H$	Buffer DB data area too short
0	1	8F5C$_H$	Buffer DB in write-protected memory
0	1	8F60$_H$	Bad user data, for example bad IP address of the FTP server
0	1	8F61$_H$	FTP server not obtainable
0	1	8F62$_H$	Job not supported or rejected by FTP server
0	1	8F63$_H$	File transfer aborted by the FTP server
0	1	8F64$_H$	Error on the FTP control connection; data could not be sent or received; the FTP control connection must be established again after such an error.
0	1	8F65$_H$	Error on the FTP data connection; data could not be sent or received; the job (FTP_STORE or FTP_RETRIEVE) must be called again. This error can, for example, be caused by FTP_RETRIEVE when the addressed file is already open on the FTP server.
0	1	8F66$_H$	Error reading/writing data from/to the CPU (for example DB does not exist or too short)
0	1	8F67$_H$	Error in the FTP client on the IP-CP; for example attempting to open more than 10 FTP connections.
0	1	8F68$_H$	The job was rejected by the FTP client This error can, for example, be caused by FTP_RETRIEVE when the value for the parameter MAX_LENGTH was selected too low in the file DB header.
0	1	8F69$_H$	FTP connection in the incorrect status for this call, for example a double connect call or when attempting to retrieve without previously connecting (using the same NetPro ID)
0	1	8F6A$_H$	No new socket could be opened, temporary resource problem, repeat the block call.
0	1	8F70$_H$	Calling an FTP client block with ACT = 0
0	1	8F7F$_H$	Internal error, for example illegal ANY reference

2.5 FBs for programmed connections

2.5.1 FB 55 IP_CONFIG - meaning and call

Meaning of the block

Connections can be specified in a DB (configuration DB) and transferred to the CP with an FB.

You can use this variant of programmed communication connections as an alternative to connection configuration with STEP 7.

With the function block FB55, a configuration data block (CONF_DB) is transferred to the CP. The configuration data block contains all the connection data for an Ethernet CP.

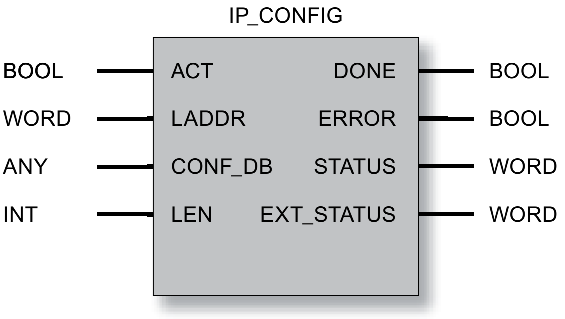
Depending on the size of the configuration DB, the data may be transferred to the CP in several segments. This means that the FB must continue to be called until the FB signals complete transfer by setting the DONE bit to 1.

Note

Read the description of the configuration data block CONF_DB.

Call

Call interface in FBD representation



Example in STL representation

STL	Explanation
call fb 55(	//IP_CONFIG block call
ACT:=M 10.0,	//Job initiated by memory bit
LADDR:=W#16#0100,	//=LADDR 256 dec. in hardware configuration
CONF_DB:= P#db99.dbx10.0 byte 240,	//Data block with connection data
LEN:=MW 14,	//Length info for the connection data
DONE:=M 10.1,	//Execution code
ERROR:=M 10.2,	//Error code
STATUS:=MW 16,	//Status code
EXT_STATUS :=MW 18);	//Cause of error in connection data

2.5.2 How IP_CONFIG works

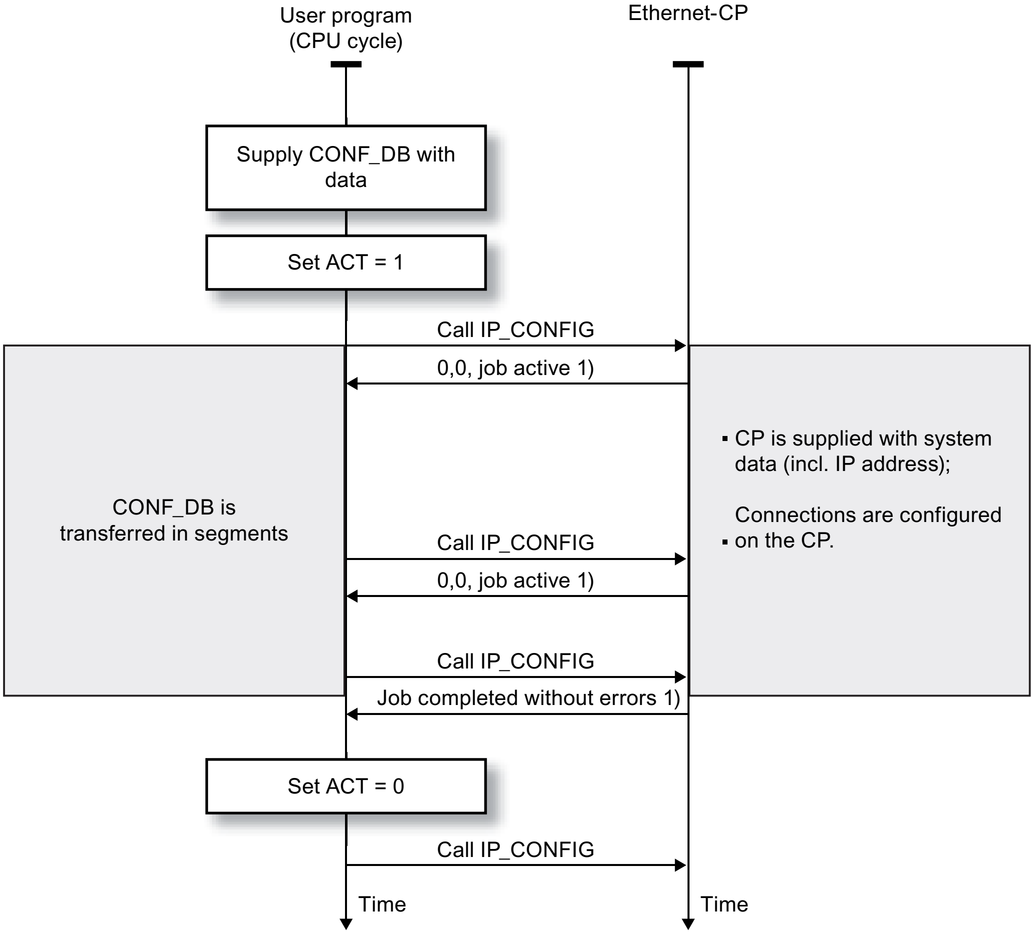
Operating principle

The following diagram illustrates the normal sequence of connection configuration triggered by an IP_CONFIG in the user program.

The job executes as soon as the parameter ACT is transferred with value 1.

Due to the segmented transfer of the CONF_DB, you will need to repeat the job with ACT = 1 until completion of the job is indicated in the parameters DONE, ERROR, and STATUS.

If you want to transfer a connection configuration again later, the parameter ACT must first be transferred with value 0 in at least one further call.



1) Parameter transfer DONE, ERROR, STATUS

NOTICE
The data transferred with the configuration DB are stored in volatile memory on the CP and must be downloaded to the CP again following a power down!

2.5.3 Explanation of the formal parameters - IP_CONFIG

Explanation of the formal parameters

The following table explains the formal parameters for the call interface of the IP_CONFIG function block :

Parameter	Declaration	Data type	Possible values	Description
ACT	INPUT	BOOL	0,1	When the FB is called with ACT = 1, the DBxx is sent to the CP. If the FB is called with ACT = 0, only the status codes DONE, ERROR and STATUS are updated.
LADDR	INPUT	WORD		Module start address When you configure the CP with STEP 7 HW Config, the module start address is displayed in the configuration table. Specify this address here.
CONF_DB	INPUT	ANY		The parameter points to the start address of the configuration data area in a data block (data type: byte).
LEN	INPUT	INT		Length information in bytes for the configuration data area.
DONE	OUTPUT	BOOL	0: - 1: Job completed with data transfer.	The parameter indicates whether the configuration data area was completely transferred. Remember that it may be necessary to call the FB several times depending on the size of the configuration data area (in several cycles) until the DONE parameter is set to 1 to signal completion of the transfer. For the meaning of this parameter in conjunction with the ERROR and STATUS parameters, refer to the following table.
ERROR	OUTPUT	BOOL	0: - 1: Error	Error code For the meaning of this parameter in conjunction with the DONE and STATUS parameters, refer to the following table.
STATUS	OUTPUT	WORD	See following table	Status code For the meaning of this parameter in conjunction with the DONE and ERROR parameters, refer to the following table.
EXT_ Status	OUTPUT	WORD		If an error occurs in the execution of a job, the parameter indicates which parameter was detected as the cause of the error in the configuration DB. High byte: Index of the parameter field Low byte: Index of the subfield within the parameter field

2.5.4 Reserved port numbers - IP_CONFIG

Reserved Port Numbers

The following local port numbers are reserved; do not use these in the connection project engineering.

Table 2-15 Reserved Port Numbers

Protocol	Port number	Service
TCP	20, 21	FTP
TCP	25	SMTP
TCP	80	HTTP
TCP	102	RFC1006
TCP	135	RPC-DCOM
TCP	502	ASA application protocol
UDP	161	SNMP_REQUEST
UDP	34964	PN IO
UDP	65532	NTP
UDP	65533	NTP
UDP	65534	NTP
UDP	65535	NTP

2.5.5 Condition codes of the IP_CONFIG block

Condition codes

The following table shows the condition codes formed based on DONE, ERROR and STATUS that must be evaluated by the user program.

Table 2-16 Condition codes for FB55 IP_CONFIG

DONE	ERROR	STATUS	Meaning
General codes relating to job execution			
1	0	0000_H	Job completed without errors
0	0	8181_H	Job active
Errors detected on the interface between CPU and CP.			
0	1	$80A4_H$	• Communication error on the K-bus or • Data error: Configuration by the user program is not set.
0	1	$80B1_H$	The number of data bytes to be sent exceeds the upper limit for this service. (upper limit = 16 Kbytes)

DONE	ERROR	STATUS	Meaning
0	1	$80C4_H$	Communication error The error can occur temporarily; it is usually best to repeat the job in the user program.
0	1	$80D2_H$	Configuration error The module you are using does not support this service.
Errors detected in the evaluation of the FB in the CPU or on the interface between CPU and CP.			
0	1	8183_H	The CP rejects the requested data record number.
0	1	8184_H	System error or bad parameter type. (data type of the ANY pointer CONF_DB not OK) (Currently only the byte data type is accepted)
0	1	8185_H	The value of the LEN parameter is larger than the CONF_DB less the reserved header (4 bytes) or the length information is incorrect.
0	1	8186_H	Illegal parameter detected The ANY pointer CONF_DB does not point to a data block.
0	1	8187_H	Illegal status of the FB Data in the header of CONF_DB was possibly overwritten.
Further errors detected on the interface between the CPU and CP.			
0	1	$8A01_H$	The status code in the data record is invalid (value is >= 3).
0	1	$8A02_H$	There is no job running on the CP; the FB, however, expected an acknowledgment for a completed job.
0	1	$8A03_H$	There is no job running on the CP and the CP is not ready; the FB triggered the first job to read a data record.
0	1	$8A04_H$	There is no job running on the CP and the CP is not ready; the FB nevertheless expected an acknowledgment for a completed job.
0	1	$8A05_H$	There is a job running, but there was no acknowledgment; the FB nevertheless triggered the first job for a read data record job.
0	1	$8A06_H$	A job is complete but the FB nevertheless triggered the first job for a read data record job.
Errors detected when evaluating the FB on the CP.			
0	1	$8B01_H$	Communication error The DB could not be transferred.
0	1	$8B02_H$	Parameter error Double parameter field
0	1	$8B03_H$	Parameter error The subfield in the parameter field is not permitted.
0	1	$8B04_H$	Parameter error The length specified in the FB does not match the length of the parameter fields / subfields.
0	1	$8B05_H$	Parameter error The length of the parameter field is invalid.
0	1	$8B06_H$	Parameter error The length of the subfield is invalid.
0	1	$8B07_H$	Parameter error The ID of the parameter field is invalid
0	1	$8B08_H$	Parameter error The ID of the subfield is invalid

DONE	ERROR	STATUS	Meaning
0	1	8B09$_H$	System error The connection does not exist
0	1	8B0A$_H$	Data error The content of the subfield is not correct.
0	1	8B0B$_H$	Structure error A subfield exists twice.
0	1	8B0C$_H$	Data error The parameter does not contain all the necessary parameters.
0	1	8B0D$_H$	Data error The CONF_DB does not contain a parameter field for system data.
0	1	8B0E$_H$	Data error / structure error The CONF_DB type is invalid.
0	1	8B0F$_H$	System error The CP does not have enough resources to process CONF_DB completely.
0	1	8B10$_H$	Data error Configuration by the user program is not set.
0	1	8B11$_H$	Data error The specified type of the parameter field is invalid.
0	1	8B12$_H$	Data error Too many connections were specified (either in total or too many for a specific type; for example, only one Email connection is possible).
0	1	8B13$_H$	CPinternal error
0	1	8B14$_H$	The active protection level does not permit the change that will result from the action.
Further errors detected on the program interfaces within the CPU (SFC errors).			
0	1	8F22$_H$	Area length error reading a parameter (e.g. DB too short).
0	1	8F23$_H$	Area length error writing a parameter (e.g. DB too short).
0	1	8F24$_H$	Area error reading a parameter.
0	1	8F25$_H$	Area error writing a parameter.
0	1	8F28$_H$	Alignment error reading a parameter.
0	1	8F29$_H$	Alignment error writing a parameter.
0	1	8F30$_H$	The parameter is in the writeprotected first current data block.
0	1	8F31$_H$	The parameter is in the writeprotected second current data block.
0	1	8F32$_H$	The parameter contains a DB number that is too high.
0	1	8F33$_H$	DB number error
0	1	8F3A$_H$	The target area was not loaded (DB).
0	1	8F42$_H$	Timeout reading a parameter from the I/O area.
0	1	8F43$_H$	Timeout writing a parameter to the I/O area.
0	1	8F44$_H$	Access to a parameter to be read during block execution is prevented.
0	1	8F45$_H$	Access to a parameter to be written during block execution is prevented.
0	1	8F7F$_H$	Internal error For example, an illegal ANY reference was detected.

2.6 Configuration limits / resources required by the FCs and FBs (Ethernet)

Required resources

NOTICE
Please note the version information of the blocks. Blocks with other versions have different resource requirements.

Table 2-17 Information for FCs / FBs with an S7-400

NAME	Version	FC/FB no.	Load memory bytes	Work memory bytes	MC7 bytes	Local data bytes
AG_SEND	1.1	FC5	732	576	540	20
AG_RECV	1.1	FC6	656	522	486	20
AG_LOCK	1.0	FC7	272	200	164	6
AG_UNLOCK	1.0	FC8	256	186	150	6
AG_LSEND	3.0	FC50	1044	846	810	52
AG_LRECV	3.0	FC60	1190	992	956	58
AG_SSEND	1.0	FC53	1642	1386	1350	118
AG_SRECV	1.0	FC63	1600	1356	1320	122
FTP_CMD	1.0	FB40	1998	1726	1690	58
FTP_CONNECT	1.0	FC40	1482	1236	1200	86
FTP_STORE	1.0	FC41	1794	1514	1478	102
FTP_RETRIEVE	1.0	FC42	1934	1642	1606	106
FTP_DELETE	1.0	FC43	1478	1232	1196	86
FTP_QUIT	1.0	FC44	968	796	760	46

Table 2-18 Information for FCs / FBs with an S7-300

NAME	Version	FC/FB no.	Load memory bytes	Work memory bytes	MC7 bytes	Local data bytes
AG_SEND	4.2	FC5	1976	1664	1628	50
AG_RECV	4.7	FC6	1440	1206	1170	40
AG_LOCK	4.0	FC7	748	636	600	34
AG_UNLOCK	4.0	FC8	712	604	568	32
AG_CNTRL	1.0	FC10	1402	1138	1102	82
IP_CONFIG	1.1	FB55	2478	2056	2020	62
FTP_CMD	1.0	FB40	2590	2240	2204	70
FTP_CONNECT	1.1	FC40	928	774	738	68
FTP_STORE	1.1	FC41	1232	1046	1010	74

NAME	Version	FC/FB no.	Load memory bytes	Work memory bytes	MC7 bytes	Local data bytes
FTP_RETRIEVE	1.1	FC42	1310	1118	1082	84
FTP_DELETE	1.1	FC43	922	770	734	68
FTP_QUIT	1.1	FC44	452	370	334	28

FCs / FBs for PROFINET 3

3.1 FBs for PROFINET CBA

3.1.1 FB88 PN_InOut / FB90 PN_InOut_Fast - meaning and call

How It works

The task of function block FB88 / FB90 is to transfer data from the interface DB to the CP and from the CP to the interface DB. The interface DB itself is the interface to the user program.

FB88 / FB90 is called cyclically. It is also possible to call FB88 /FB90 more than once in a cycle.

At its interface, FB88 /FB90 only needs to be supplied with the module address of the CP.

To ensure data consistency, you can only modify the data to be transferred or start to read the received data when the job is completed (DONE=1 or ERROR=1).

As soon as DONE=1 or ERROR=1 is set, the transfer is complete or has been terminated with an error message. Data can now be evaluated or set again. Data will only be transferred with the next call.

In your user program, make sure that on completion of data transfer FB88 /FB90 is called again only after all the input data has been read and all output data has been written to the interface DB.

Calling the FB88 / FB90 blocks time-driven is permitted in principle. Refer to the notes on this mode later in this chapter.

Differences between FB88 and FB90

The activities of the function blocks FB90 and FB88 on the interface to the user program are largely identical. You can use FB90 with certain CP/CPU types with an S7-400; refer to the information in the manual of the CP.

If FB90 is approved for the CP type being used, we recommend that you use it. As a result, you can achieve shorter reaction times than with FB88. You should, however, remember the constraints regarding its use.

The following points apply:

- The interface parameters are identical;
- For FB90, there are several additional codes in the STATUS parameter;

- With some errors, there are different codes in the STATUS parameter of FB88 and FB90;
- There are differences in the configuration limits of the interface DB (see relevant manual).

Note

For more detailed information on handling the interface DB, refer to the SIMATIC iMap documentation.

NOTICE
When you reload user program blocks, data consistency is only guaranteed if the CPU is first changed to STOP.

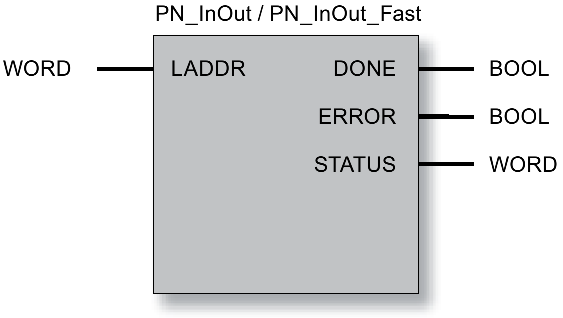
How supplied - block library

FB88 and FB90 are supplied with SIMATIC iMap. There are different block types for S7-300 and S7-400.

After installation, the blocks are available in the PROFINET Library under "PROFINET_System_Library/CP300 or /CP400".

Call interface

Call interface in FBD representation



Example of a call in STL representation

STL	Explanation
`Call FB 88 , DB88 (`	`//Block call with instance DB88`
`LADDR:=W#16#0120,`	
`DONE:=M 99.1,`	
`ERROR:=M 99.0,`	
`STATUS:=MW 104);`	

3.1.2 Explanation of the formal parameters - PN_InOut / PN_InOut_Fast

Explanation of the formal parameters

The following table explains all the formal parameters for FB88 / FB90:

Parameter	Declaration	Data type	Description
LADDR	INPUT	WORD	Module start address When you configure the CP, the module start address is displayed in the configuration table. Specify this address here. Do not modify the parameter until the job is completed (DONE=1 or ERROR=1).
DONE	OUTPUT	BOOL	Reports the (positive) completion of a job.
ERROR	OUTPUT	BOOL	This indicates that the job could not be executed errorfree.
STATUS	OUTPUT	WORD	This parameter supplies detailed information about the execution of the job. Status codes can be returned during execution of the job (DONE=0 and ERROR=0).

3.1.3 Condition codes of the PN_InOut and PN_InOut_Fast blocks

Evaluating status codes

Remember that the status codes DONE, ERROR, STATUS are updated at each block call.

The following table shows the condition codes formed based on DONE, ERROR and STATUS that must be evaluated by the user program.

Table 3-1 Codes for PN_InOut (FB88) and PN_InOut_Fast (FB90)

DONE	ERROR	STATUS	Meaning
1	0	0000_H	Job completed without error.
0	0	0000_H	No job being processed; the block can be called.
0	0	8181_H	• Job active. or • (only with FB90): Connection establishment to addressed module active (see also information under 8090_H).
0	1	8183_H	(S7-300 only) The service has not yet started; data acceptance is not yet possible.
0	1	8184_H	• Bad instance DB, generally triggered by illegal writing of the instance DB by the user program. or • (only with FB90) Bad send or receive job.
0	1	8085_H	(only with FB90) Bad interface DB.

DONE	ERROR	STATUS	Meaning
0	1	8090_H	(S7-400 only) Parameter assignment error An incorrect module address was specified; the address points to an empty slot. Note (only with FB90): In the following cases, the value 8181H is shown in STATUS (job active); In actual fact, there is no communication: • The address points to a slot that contains a different module type. • The addressed module is configured for PROFINET CBA operation.
0	1	$80A1_H$	(only with FB90) Possible communications errors: • Stationinternal connection to addressed module being terminated; • The configuration limits for connections of the CPU has been exceeded; • The interface is being reinitialized.
0	1	$80B0_H$	(S7-300 only) Block error: The data record number is wrong. This status can also occur after the following actions: • Cold or warm restart after power DOWN/UP • Cold or warm restart on the CPU
0	1	$80B1_H$	(S7-300 only) Block error: Data record length or offset wrong.
0	1	$80B3_H$	(S7-300 only) Parameter error: Wrong CP address.
0	1	$80C1_H$	(S7-300 only) Temporary error: The specified data record is currently being processed.
0	1	$80C2_H$	(S7-300 only) Temporary error: There is a job bottleneck; the data record cannot be read yet.
0	1	$80C3_H$	(S7-300 only) Temporary error: Resources occupied (memory).
0	1	$80C4_H$	(S7-300 only) Communication error: Occurs temporarily and a repetition in the user program will often remedy the problem.
0	1	$80D0_H$	(S7-300 only) Configuration error: The maximum number of blocks of input and output data has been exceeded; the interface DB is too large.
0	1	$80D1_H$	(S7-300 only) Configuration error Possible causes: • The interface of the configured component does not match the one used in the program (outputs). • The wrong module was inserted; The PROFINET service is not supported.

DONE	ERROR	STATUS	Meaning
0	1	80D2$_H$	(S7-300 only) Configuration error Possible causes: • The interface of the configured component does not match the one used in the program (inputs). • The wrong module was inserted; The PROFINET service is not supported. • Parameter error: Wrong CP address
0	1	8322$_H$	(only with FB90) Bad interface DB.
0	1	8332$_H$	(only with FB90) The number of the interface DB is too high.
0	1	833A$_H$	(only with FB90) Access to the interface DB is not possible (possibly because the interface DB was deleted).
0	1	8623$_H$	(only with FB90) Bad interface DB.
0	1	863A$_H$	(only with FB90) Access to the interface DB is not possible (possibly because the interface DB was deleted).

To find out which SFCs are used that are relevant for error evaluation, display the properties dialog of the FB described here in the "Calls" tab.

Note

For entries with the coding 8FxxH (for S7-300) or 8xxxH (for S7-400) under STATUS, note the information in the Reference Manual STEP 7 Standard and System Functions. The chapter describing error evaluation with the RET_VAL output parameter contains detailed information

Status codes during CP startup

With a complete restart or restart of the PROFINET CP (after activating a switch on the module), the output parameters of the block are reset as follows:

- DONE = 0
- ERROR = 0
- STATUS = 8181$_H$

3.1.4 Timedriven PN_InOut / PN_InOut_Fast call - recommendation on application

Timedriven call - recommendation on application

If your application requires timedriven transfer of the CBA data instead of cyclic or event-driven processing, we recommend the following procedure to call the FB88 / FB90 blocks.

If you use a timedriven call, remember that the block must be called repeatedly after it has been started until the DONE flag is set. To allow the CBA data to be copied between the CPU and CP without any longer interruptions, the followon calls should be as fast as possible and independent of the timing.

Note the following recommendations for programming:

- The timing is provided by a timer OB; the timer OB should only cause the first call for the PROFINET CBA blocks FB88 or FB90 not by calling the block directly but, for example, by setting a start flag.
- The call of the PROFINET CBA blocks FB88 and FB90 should then always be made in OB1; OB1 starts the call as soon as the start flag is set by the timer OB.
- After the first block call, this is repeated in OB1 until the DONE bit is set (or until an error occurs); the start flag must then be reset.

Result:
The CBA user data can be copied between the CPU and CP without any significant interruption thanks to separating the timer OB from the actual block calls in OB1. You can select the interval between the first calls depending on the requirements of your application.

3.2 FCs/FBs for PROFINET IO (S7-300)

3.2.1 Overview of FCs/FBs and their use

Overview

The FCs listed below are available for transferring data cyclically on the PROFINET IO interface. The significance of the FCs differs depending on how you use the CP (as a PROFINET IO controller or PROFINET IO device) in an S7 station.

FC	can be used with:		Meaning
	S7-300	S7-400	
PNIO_SEND (FC11)	x	-	Depending on the mode of the CP: • For a PROFINET IO controller Sending output data to the PROFINET IO devices. • On a PROFINET IO device Forwarding process input data to the PROFINET IO controller.
PNIO_RECV (FC12)	x	-	Depending on the mode of the CP: • On a PROFINET IO controller Receiving process input data from the PROFINET IO devices. • On a PROFINET IO device Receiving process output data from the PROFINET IO controller.

For CPs operating as PROFINET IO controller and IO device at the same time, the FCs as of version 2.0 are available.

The FBs listed below are available for transferring data (data records, alarm information) acyclically on the PROFINET IO interface. The two blocks can only be used in PROFINET IO controller mode.

FB	can be used with:		Meaning
	S7-300	S7-400	
PNIO_RW_REC (FB52)	x	-	• Read data record (from a PROFINET IO device) • Write data record (to a PROFINET IO device)
PNIO_ALARM (FB54)	x	-	Receive alarm information from the PROFINET IO devices

3.2.2 FC11 PNIO_SEND

3.2.2.1 Meaning and call - PNIO_SEND

How It works

The PNIO_SEND FC is used to transfer data in the PROFINET IO controller or PROFINET IO device modes of the CP.

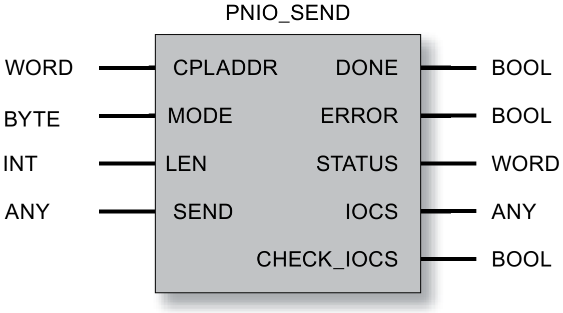
- Operating as PROFINET IO controller
  The block transfers the process data (outputs) of a specified output area to the CP to be forwarded to PROFINET IO devices and as status display returns the IO consumer status (IOCS) of the outputs of the PROFINET IO devices.
- Operating as PROFINET IO device
  The block reads the preprocessed process inputs of the CPU on the PROFINET IO device and transfers them to the PROFINET IO controller (configured I addresses); the block also returns the IO Consumer Status (IOCS) of the PROFINET IO controller as a status display.

The preprocessed process data is available in a DB or bit memory area.

For CPs operating as PROFINET IO controller and IO device at the same time, the FC as of version 2.0 is available. With the additional MODE parameter, you set the mode for which the FC will be called.

Call interface (block version 2.0)

Call interface in FBD representation



Example in STL representation

STL	Explanation
call fc 11(	//PNIO_SEND block call
CPLADDR:=W#16#0100,	//Module address from hardware configuration
MODE:=0,	//Controller mode or device mode
LEN:=20,	//Length of the data area
IOCS:=P#DB10.DBX20.0 BYTE 3,	//Per send data byte one bit status in DB10
DONE:=M 70.0,	//Address for return parameter DONE
ERROR:=M 70.1,	//Address for return parameter ERROR
STATUS:=MW 72,	//Address for return parameter STATUS

```
STL                                    Explanation
CHECK_IOCS:=M 70.2,                    //Address for return parameter CHECK_IOCS
SEND:=P#DB10.DBX0.0 BYTE 20 );         //Data area to be transferred from DB10
                                       //(20 bytes)
```

3.2.2.2 Explanation of the formal parameters - PNIO_SEND

Explanation of the formal parameters

The following table explains all the formal parameters for FC11:

Parameter	Declaration	Data type	Possible values	Description
CPLADDR	INPUT	WORD	-	Module start address
MODE (parameters version 2.0 or later)	INPUT	BYTE	• 0: – IO controller mode – IO device operation (without parallel operation) – Not compatible with FC in version 1.0 • 1: IO device mode (both modes at same time)	Specifies the mode of the CP Notes on compatibility;: • The version 1.0 FC can continue to be used as long as the CP is not being operated as an IO controller and IO device at the same time. • When MODE=0, the FC as of version 2.0 behaves like the FC version 1.0.

Parameter	Declaration	Data type	Possible values	Description
SEND	IN_OUT	ANY (as VARTYPE only BYTE is permitted)	The address of the data area points to one of the alternatives: • Memory bit area • Data block area	Specifies the address and length IO controller mode: The length should match the total length of the distributed IO configured, whereby address gaps are also transmitted. The length can also be shorted than the total length of the distributed IO, for example when the block is called more than once in one OB. It must, however, have the total length in at least one call. IO device mode: The data structure results from the order of the slots of the input modules configured for this PROFINET IO device on the PROFINET IO controller line and their length without address gaps. (Please note the more extensive explanations or examples for your CP in the devicespecific Part B of this manual) Notes: • The block begins to transfer the data at address 0 regardless of how you configured the addresses (regardless of the lowest configured address). • Specifying an I/O area is not permitted since you must first check the IOCS for GOOD before data can be accepted in the I/O.

Parameter	Declaration	Data type	Possible values	Description
LEN	INPUT	INT	Value > 0 The maximum total length of the data areas to be transferred can be found in the devicespecific Part B of this manual in the "Performance data" chapter. This may differ for controller or device mode.	Length of the data area to be transferred in bytes. The transfer of the data always begins with address 0 regardless of the configuration. Please note that the IO address "0" with a length of 1 is included. IO controller mode: • The highest configured address of the devices must be specified here. The individual areas are not grouped together. If the block is called more than once, LEN can also be shorter than the highest address. The highest address should be specified in at least one call (compare "SEND" parameter). • The data is transferred in the order of the logical addresses (as with PROFIBUS DP). IO device mode: • The data is transferred in the order of the slots as the configured input modules on the PROFINET IO controller chain for this PROFINET IO device. Note: Make sure that the length programmed here and the configuration of the PROFINET IO controller are consistent. The entire data area length including any gaps is transferred for the device.
DONE	OUTPUT	BOOL	0: - 1: New data accepted	This parameter indicates whether or not the job was completed without errors.
ERROR	OUTPUT	BOOL	0: -1: Error	Error code
STATUS	OUTPUT	WORD	-	Status code
CHECK_IOCS	OUTPUT	BOOL	0: All IOCS set to GOOD 1: At least one IOCS set to BAD	Auxiliary bit that indicates whether or not it is necessary to evaluate the IOCS status area.

Parameter	Declaration	Data type	Possible values	Description
IOCS	OUTPUT	ANY (as VARTYPE only BYTE is permitted)	The address of the data area points to one of the alternatives: • Memory bit area • Data block area Length: For the maximum value, refer to the devicespecific Part B of this manual in the section "Performance data". This may differ for controller or device mode.	A status bit is transferred per byte of user data. The length information depends on the length in the LEN parameter (one bit per byte) = (Length LEN + 7/ 8) Controller mode: Address gaps are also transferred according to the SEND parameter. Address gaps are transferred with the status GOOD. Device mode: Address gaps are not transferred. The block begins the transfer of the status for address 0. Note: The minimum length of the ANY pointer is (length LEN + 7/8)

Note

Remember that all output parameters may only be evaluated when the block signals either DONE = 1 or ERROR = 1.

NOTICE

You must assume that the returned IOCS status does not arrive timesynchronized with the data (SEND parameter) but delayed by one user program cycle. This means: User data and IOCS are not consistent.

3.2.2.3 Condition codes of PNIO_SEND

Condition codes

The following table shows the condition codes formed based on DONE, ERROR and STATUS that must be evaluated by the user program.

Note

For entries coded with 8FxxH in STATUS, refer to the information in the STEP 7 Standard and System Functions reference manual. The chapter describing error evaluation with the RET_VAL output parameter contains detailed information.

To find out which SFCs are used and are relevant for error evaluation, display the properties dialog of the FC described here in the "Calls" tab.

Table 3-2 Condition codes PNIO_SEND

DONE	ERROR	STATUS	Meaning
0	0	8180$_H$	• Data transfer active; or • The CP is in STOP mode.
0	0	8181$_H$	Module does not support block version 2.0. Remedy: Use block version 1.0.
1	0	0000$_H$	New data transferred without error.
0	1	8183$_H$	• PROFINET IO configuration missing; or • wrong CPLADDR; or • The CP is in STOP mode. or • Interconnection of MODE does not match module configuration or incorrect interconnection with MODE > 1 Extra in device mode: • The connection between PROFINET IO controller and PROFINET IO device is down, or • PROFINET IO controller not reachable or • Total lengths (configuration and LEN parameter) are not consistent.
0	1	8184$_H$	System error or bad parameter type.
0	1	8185$_H$	Parameter LEN is greater than source area SEND or target buffer (IOCS) is too small.
0	1	8F22$_H$	Area length error reading a parameter (e.g. DB too short).
0	1	8F23$_H$	Area length error writing a parameter (e.g. DB too short).
0	1	8F24$_H$	Range error when reading a parameter.
0	1	8F25$_H$	Range error when writing a parameter.
0	1	8F28$_H$	Alignment error when reading a parameter.
0	1	8F29$_H$	Alignment error when writing a parameter.
0	1	8F30$_H$	Parameter is in the write-protected 1st current data block.
0	1	8F31$_H$	Parameter is in the write-protected 2nd current data block.
0	1	8F32$_H$	Parameter contains a DB number that is too high.
0	1	8F3A$_H$	Destination area is not loaded (DB).
0	1	8F42$_H$	Timeout reading a parameter from the I/O area.
0	1	8F43$_H$	Timeout writing a parameter to the I/O area.
0	1	8F44$_H$	Access to a parameter to be read during block execution is prevented.
0	1	8F45$_H$	Access to a parameter to be written during block execution is prevented.
0	1	8F7F$_H$	Internal error, e.g. illegal ANY reference.
0	1	8090$_H$	Module with this address does not exist.
0	1	80A0$_H$	Negative acknowledgment writing to the module.

DONE	ERROR	STATUS	Meaning
0	1	$80A1_H$	Negative acknowledgment writing to the module.
0	1	$80B0_H$	The module does not recognize the data record.
0	1	$80B1_H$	• The specified data record length is incorrect. or • The CP changes to STOP.
0	1	$80C0_H$	The data record cannot be read.
0	1	$80C1_H$	The specified data record is currently being processed.
0	1	$80C2_H$	There are too many jobs pending.
0	1	$80C3_H$	Resources occupied (memory).
0	1	$80C4_H$	Communication error (occurs temporarily, it is usually best to repeat the job in the user program).

3.2.3 FC12 PNIO_RECV

3.2.3.1 Meaning and call - PNIO_RECV

How It works

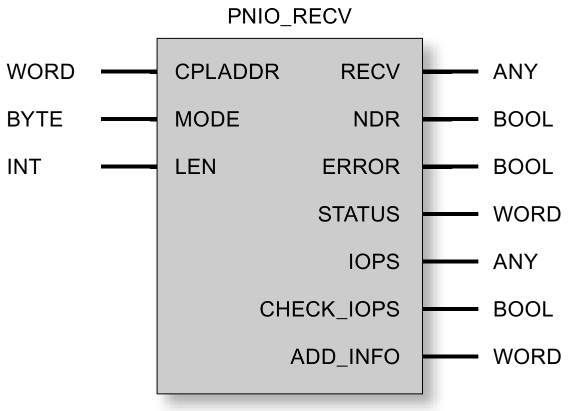
The PNIO_RECV FC is used to accept data in the PROFINET IO controller or PROFINET IO device CP modes.

- Operating as PROFINET IO controller
  The block accepts the process data from PROFINET IO devices (inputs of the controller) and transfers the IO provider status (IOPS) from the PROFINET IO devices to the specified input areas.
- Operating as PROFINET IO device
  The block receives the data transferred by the PROFINET IO controller (configured O addresses) as well as the IO Provider Status (IOPS) of the PROFINET IO controller and writes them to the data areas on the CPU of the PROFINET IO device reserved for the process outputs.

For CPs operating as PROFINET IO controller and IO device at the same time, the FC as of version 2.0 is available. With the additional MODE parameter, you set the mode for which the FC will be called.

Call interface (block version 2.0)

Call interface in FBD representation



Example in STL representation

STL	**Explanation**
`call fc 12(`	`//PNIO_RECV block call`
`CPLADDR:=W#16#0100,`	`//Module address from hardware configuration`
`MODE:=0,`	`//Controller mode or device mode`
`LEN:=7,`	`//Length of the data area`
`IOPS:=P#DB11.DBX7.0 BYTE 1,`	`//Per receive data byte one status bit in DB11`
`NDR:=M 74.0,`	`//Address for return parameter NDR`
`ERROR:=M 74.1,`	`//Address for return parameter ERROR`
`STATUS:=MW76,`	`//Address for return parameter STATUS`
`CHECK_IOPS:=M74.2,`	`//Address for return parameter CHECK_IOPS`
`ADD_INFO:=MW 26,`	`//Diagnostic information`
`RECV:=P#DB11.DBX0.0 BYTE 7 );`	`//Received data in DB11 (7 bytes)`

See also

Data consistency (Page 107)

Substitute values (Page 108)

3.2.3.2 Explanation of the formal parameters - PNIO_RECV

Explanation of the formal parameters

The following table explains all the formal parameters for FC12:

Parameter	Declaration	Data type	Possible values	Description
CPLADDR	INPUT	WORD	-	Module start address
MODE (parameters version 2.0 or later)	INPUT	BYTE	• 0: – IO controller mode – IO device operation (without parallel operation) – Not compatible with FC in version 1.0 • 1: IO device mode (both modes at same time)	Specifies the mode of the CP. Notes on compatibility;: • The version 1.0 FC can continue to be used as long as the CP is not being operated as an IO controller and IO device at the same time. • When MODE=0, the FC as of version 2.0 behaves like the FC version 1.0.
RECV	IN_OUT	ANY (as VARTYPE only BYTE is permitted)	The address of the data area points to one of the alternatives: • Memory bit area • Data block area	Specifies the address and length IO controller mode: The length should match the total length of the distributed IO configured, whereby address gaps are also transmitted. The length can also be shorted than the total length of the distributed IO, for example when the block is called more than once in one OB. It must, however, have the total length in at least one call. IO device mode: The data structure results from the order of the slots of the output modules configured for this PROFINET IO device on the PROFINET IO controller line and their length without address gaps. Notes: • The block begins to transfer the data at address 0 regardless of how you configured the addresses (regardless of the lowest configured address). • Specifying an I/O area is not permitted since you must first change the IOPS for GOOD before data can be accepted in the I/O.

Parameter	Declaration	Data type	Possible values	Description
LEN	INPUT	INT	Value > 0 The maximum total length of the data to be transferred can be found in the devicespecific Part B of this manual in the section "Performance data". This may differ for controller or device mode.	Length of the data area to be transferred in bytes. The transfer of the data always begins with address 0 regardless of the configuration. Please note that the IO address "0" with a length of 1 is included. IO controller mode: • The highest configured address of the devices must be specified here. The individual areas are not grouped together. If the block is called more than once, LEN can also be shorter than the highest address. The highest address should be specified in at least one call (compare "RECV" parameter). • The data is transferred in the order of the logical addresses (as with PROFIBUS DP). IO device mode: • The data is transferred in the order of the slots corresponding to the configuration of the input modules on the PROFINET IO controller line for this PROFINET IO device. • Note: Make sure that the length programmed here and the configuration of the PROFINET IO controller are consistent. The entire data area length including any gaps is transferred for the device.
NDR	OUTPUT	BOOL	0: - 1: Data accepted	This parameter indicates whether or not the job was completed without errors.
ERROR	OUTPUT	BOOL	0: - 1: Error	Error code
STATUS	OUTPUT	WORD	-	Status code
CHECK_ IOPS	OUTPUT	BOOL	0: All IOPS set to GOOD 1: At least one IOPS set to BAD	Auxiliary bit that indicates whether or not it is necessary to evaluate the IOPS status area.

Parameter	Declaration	Data type	Possible values	Description
IOPS	OUTPUT	ANY (as VARTYPE only BYTE is permitted)	The address of the data area points to one of the alternatives: • Memory bit area • Data block area Length: For the maximum value, refer to the devicespecific Part B of this manual in the section "Performance data". This may differ for controller or device mode.	A status bit is transferred per byte of user data. The length information depends on the length in the RECV parameter (one bit per byte) = (Length LEN + 7/ 8) Controller mode: Address gaps are also transferred according to the RECV parameter. Address gaps are transferred with the status GOOD. Device mode: Address gaps are not transferred. The block begins the transfer of the status for address 0. Note: • The minimum length of the ANY pointer is (length LEN + 7/8)
ADD_INFO	OUTPUT	WORD	Additional Diagnostic Information In controller mode: • 0: No alarm • >0: Number of pending alarms In device mode, the parameter is always = 0.	Parameter expansion Note: The ADD_INFO parameter is also updated when there are no INPUT addresses configured on the PROFINET IO controller. In this case, the PNIO_RECV block is called with a length LEN > 0 (for example LEN = 1 byte). It then transfers an address gap of 1 byte. The parameter expansion can be used for CPs as of the following firmware version: • CP 343−1 (EX30) as of firmware V2.0 • CP 343−1 Lean (CX10) as of firmware V2.0 • CP 343−1 Advanced (GX30) as of firmware V1.0 In older firmware versions, the parameter is reserved.

Note

Remember that all output parameters may only be evaluated when the block signals either NDR = 1 or ERROR = 1.

3.2.3.3 Condition codes of PNIO_RECV

Condition codes

The following table shows the codes formed by the NDR, ERROR and STATUS parameters that must be evaluated by the user program.

Note

For entries coded with 8FxxH in STATUS, refer to the information in the STEP 7 Standard and System Functions reference manual. The chapter describing error evaluation with the RET_VAL output parameter contains detailed information.

To find out which SFCs are used and are relevant for error evaluation, display the properties dialog of the FC described here in the "Calls" tab.

Table 3-3 Condition codes PNIO_RECV

NDR	ERROR	STATUS	Meaning
0	0	8180$_H$	• Data acceptance active; or • The CP is in STOP mode.
0	0	8181$_H$	Module does not support block version 2.0. Remedy: Use block version 1.0.
1	0	0000$_H$	New data accepted without error.
0	1	8183$_H$	• PROFINET IO configuration missing; or • wrong CPLADDR; or • The CP is in STOP mode. or • Interconnection of MODE does not match module configuration or incorrect interconnection with MODE > 1. Extra in device mode: • The connection between PROFINET IO controller and PROFINET IO device is down, or • PROFINET IO controller not reachable or • Total lengths (configuration and LEN parameter) are not consistent
0	1	8184$_H$	System error or bad parameter type.
0	1	8185$_H$	Destination buffer (RECV of IOCS) is too small.
0	1	8F22$_H$	Area length error reading a parameter (e.g. DB too short).
0	1	8F23$_H$	Area length error writing a parameter (e.g. DB too short).
0	1	8F24$_H$	Range error when reading a parameter.

NDR	ERROR	STATUS	Meaning
0	1	8F25$_H$	Range error when writing a parameter.
0	1	8F28$_H$	Alignment error when reading a parameter.
0	1	8F29$_H$	Alignment error when writing a parameter.
0	1	8F30$_H$	Parameter is in the write-protected 1st current data block.
0	1	8F31$_H$	Parameter is in the write-protected 2nd current data block.
0	1	8F32$_H$	Parameter contains a DB number that is too high.
0	1	8F3A$_H$	Destination area is not loaded (DB).
0	1	8F42$_H$	Timeout reading a parameter from the I/O area.
0	1	8F43$_H$	Timeout writing a parameter to the I/O area.
0	1	8F44$_H$	Access to a parameter to be read during block execution is prevented.
0	1	8F45$_H$	Access to a parameter to be written during block execution is prevented.
0	1	8F7F$_H$	Internal error, e.g. illegal ANY reference.
0	1	8090$_H$	Module with this address does not exist.
0	1	80A0$_H$	Negative acknowledgment writing to the module.
0	1	80A1$_H$	Negative acknowledgment writing to the module.
0	1	80B0$_H$	The module does not recognize the data record.
0	1	80B1$_H$	• The specified data record length is incorrect. or • The CP changes to STOP.
0	1	80C0$_H$	The data record cannot be read.
0	1	80C1$_H$	The specified data record is currently being processed.
0	1	80C2$_H$	There are too many jobs pending.
0	1	80C3$_H$	Resources occupied (memory).
0	1	80C4$_H$	Communication error (occurs temporarily, it is usually best to repeat the job in the user program).

3.2.4 General characteristics of the FCs for PROFINET IO

IO Consumer Status (IOCS) and IO Provider Status (IOPS)

For both communication partners - CPU/CP on the one hand and IO device on the other - there is the status information GOOD or BAD for the data. This status information is transferred parallel to the data. The status of the partner that sends the data is called IOPS (IO Provider Status), the status of the receiving partner is called IOCS (IO Consumer Status).

The IOPS and IOCS status are not necessarily identical. It is, for example, possible that the S7-300 CPU is in STOP mode (output disable or no PROFINET IO blocks active). In this case, the CP as PROFINET IO controller transfers the BAD status to the IO devices.

Relationship between block call and IO data

- Operation as PROFINET IO controller
  As a PROFINET IO controller, the CP does not monitor the cyclic calls of the PNIO_SEND/RECV blocks. If the blocks are not called, the last transferred IO data and IOCS/IOPS data are taken as valid.
- Operation as PROFINET IO device
  FC11 and FC12 each have their own watchdog. Depending on the CPU cycle time, the connection to the PROFINET IO controller is terminated if one of the two blocks is no longer called following the initialization phase.

Optimizing data transfer (only when operating as PROFINET IO controller)

It is possible to call the blocks with a length (LEN parameter) that is shorter than the configured total length of the IO data on the PNIO chain.

You can use this so that timecritical data is transferred in every CPU cycle whereas non critical data is not transferred in every cycle.

Example:
You could, for example, transfer only the first data area (timecritical data) in every cycle and the total length of the configured IO data in every second cycle. To do this, place the time-critical data in the lower area (starting at IO address 0) during configuration.

3.2.5 Data consistency

The entire input or output data area of the PNIO controller is always transferred in its entirety and is therefore consistent.

- Operating as PROFINET IO controller
  Regardless of this, using the length information in the block call, you can also read or output an input or output area smaller than the configured area consistently.

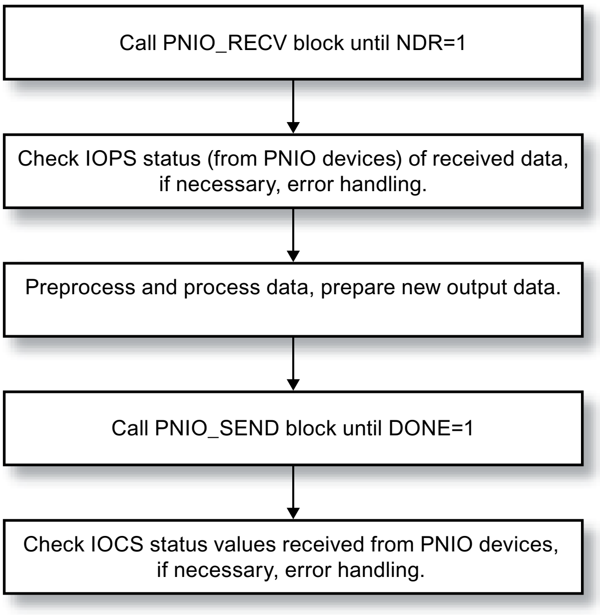
Note: You should, however, bear in mind that in terms of the "IO user data" within a PROFINET IO system, data consistency can only be guaranteed within the individual IO slots. This applies regardless of the fact that consistent data transfer between CPU and IO controller is guaranteed for the blocks described here.

Block call

To guarantee data consistency, you may, however, only access the IO data when the block has completed free of errors (output parameter NDR = TRUE). You must also check that the IOCS or IOPS status for the data is GOOD.

Example

In a normal situation (depending on the total length of the IO data), the block will run over several user program cycles until the condition code DONE/NDR = 1 is signaled.


Call PNIO_RECV block until NDR=1
Check IOPS status (from PNIO devices) of received data, if necessary, error handling.
Preprocess and process data, prepare new output data.
Call PNIO_SEND block until DONE=1
Check IOCS status values received from PNIO devices, if necessary, error handling.

Note: The user program cycle and the cycle of the IO data exchange between the PNIO controller and PNIO devices are independent of each other.

3.2.6 Substitute values

Operational situations

The setting of substitute values is supported for the two following operational situations:

- Substitute values during startup (mode change on the CPU from STOP to RUN)
- Substitute values if problems are detected (remove/insert or station failure/return)

Substitute values during startup

You can initialize the outputs with substitute values by setting a memory bit ("startup" memory bit) in the startup OB. In cyclic mode (OB1), evaluate this "startup" memory bit to call PNIO_SEND with the initialization values when appropriate.

Substitute values if a problem occurs (only when operating as PROFINET IO controller)

If there is a fault (device/submodule failed), you can find out which submodules have failed by querying the status information IOCS / IOPS status. You then have the option of setting substitute values.

3.2.7 FB52 PNIO_RW_REC

3.2.7.1 Meaning and call - PNIO_RW_REC

How It works

FB52 is used both for the "read data record" and the "write data record" function in PROFINET IO controller mode. FB52 can only execute one of the functions at any one time. The "read data record" or "write data record" function is controlled by the WRITE_REC parameter.

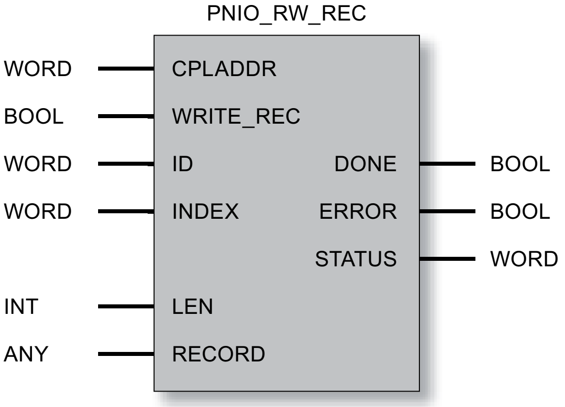
Example: The CP can be informed of the location ID and plant designation using the "write data record" function (if this parameter was not already set in the properties dialog of the CP in STEP 7). This is done using the maintenance data record "IM1" with index AFF1H.

You will find details of the supported data records and their structure at the following Internet address:

`http://support.automation.siemens.com/WW/view/en/19289930`

Call interface

Call interface in FBD representation:



Example in STL representation:

```
STL                                          Explanation
CALL FB 52, DB 52(                           //PNIO_RW_REC block call
                                              // (read/write data record)
CPLADDR:=W#16#0110,                          //Module address from hardware configuration
WRITE_REC:=M 1.1,                            //TRUE: Write data record;
                                              //FALSE: Read data record
ID:=W#16#86A,                                //Logical address of the submodule to be addressed
INDEX:=W#16#8000,                            //Data record number
DONE:=M 1.3,                                 //Address for return parameter DONE
ERROR:=M 1.1,                                //Address for return parameter ERROR
STATUS:=MW 12,                               //Address for return parameter STATUS
LEN:=MW 16,                                  //Length of the read data record /
                                              // data record to be written in bytes
RECORD:=P#DB3.DBX0.0 BYTE 80 );              //Destination or source of the data record
                                              // to be transferred (here max. 80 bytes)
```

3.2.7.2 Explanation of the formal parameters - PNIO_RW_REC

Explanation of the formal parameters

The following table explains all the formal parameters for FB52:

Parameter	Declaration	Data type	Possible values	Description
CPLADDR	INPUT	WORD	-	Module start address
WRITE_REC	INPUT	BOOL	0: Read data record 1: Write data record	Job type; The parameter must not be changed while the block is executing.
ID	INPUT	WORD		Logical address of the PROFINET IO component (module or submodule). For an output module, bit 15 is set (example of output address 5: ID:=DW#16#8005). For a mixed module, the lower of the two addresses must be specified.
INDEX	INPUT	WORD	See vendor information for the data record numbers supported by the module.	Data record number that the user wants to read or write.
DONE	OUTPUT	BOOL	0: - 1: Data record transferred successfully	This parameter indicates whether or not the job was completed without errors.
ERROR	OUTPUT	BOOL	0: - 1: Error	Error code
STATUS	OUTPUT	WORD	0: No error Different value: Error (see "")	Status code

Parameter	Declaration	Data type	Possible values	Description
LEN	IN_OUT	INT	The maximum length is 480 bytes.	• Read data record: OUTPUT parameter only; after a successful read, the length of the read data record is indicated; otherwise 0. • Write data record: INPUT parameter only; length of the data record to be written is entered here by the user. The length must match the definition of the data record.
RECORD	IN_OUT	ANY (as VARTYPE, BYTE, WORD and DWORD are permitted)	The address of the data area points to one of the alternatives: • Memory bit area • Data block area The length of the ANY pointer must be greater than or equal to the definition of the data record.	• Read data record: OUTPUT parameter only; after a successful read, the data of the data record is stored here. If the ANY pointer is too short, as much data as possible is transferred. • Write data record: INPUT parameter only; the data to be written from the data record is stored here by the user. The ANY pointer must be at least as long as specified in the LEN parameter.

3.2.7.3 Condition codes of PNIO_RW_REC

Condition codes

The following table shows the condition codes formed based on DONE, ERROR and STATUS that must be evaluated by the user program.

Note

For entries with the coding 8FxxH under STATUS, note the information in the Reference Manual "STEP 7 - System and Standard Functions for S7-300 and S7-400". The chapter describing error evaluation with the RET_VAL output parameter contains detailed information.

Table 3-4 PNIO_RW_REC condition codes

DONE	ERROR	STATUS	Meaning
0	0	8180_H	Data transfer active
1	0	0000_H	Data record transferred successfully
0	1	8183_H	• No PNIO controller configuration, • wrong CPLADDR or • CP in STOP mode
0	1	8184_H	System error or illegal parameter type
0	1	8185_H	Destination buffer (RECORD) is too short
0	1	$8F22_H$	Area length error reading a parameter (e.g. DB too short)

DONE	ERROR	STATUS	Meaning
0	1	8F23$_H$	Area length error writing a parameter (e.g. DB too short)
0	1	8F24$_H$	Area error reading a parameter
0	1	8F25$_H$	Area error writing a parameter
0	1	8F28$_H$	Orientation error when reading a parameter
0	1	8F29$_H$	Alignment error writing a parameter
0	1	8F30$_H$	Parameter is in the write-protected first active data block
0	1	8F31$_H$	Parameter is in the write-protected second active data block
0	1	8F32$_H$	The DB number in the parameter is too high
0	1	8F3A$_H$	Destination area not loaded (DB)
0	1	8F42$_H$	Timeout reading a parameter from the I/O area
0	1	8F43$_H$	Timeout writing a parameter to the I/O area
0	1	8F44$_H$	Address of the parameter to be read is disabled in the accessed rack
0	1	8F45$_H$	Address of the parameter to be written is disabled in the accessed rack
0	1	8F7F$_H$	Internal error, e.g. illegal ANY reference
0	1	8090$_H$	Module with this address does not exist
0	1	80A0$_H$	Negative acknowledgment reading from the module
0	1	80A1$_H$	Negative acknowledgment writing to the module
0	1	80A3$_H$	General PROFINET IO context management error
0	1	80A9$_H$	PROFINET IO device or module reports an illegal type
0	1	80B0$_H$	Module does not recognize the data record
0	1	80B1$_H$	• The specified data record length is incorrect or • The CP changes to STOP
0	1	80B2$_H$	The logical address or the configured slot is not in use
0	1	80B4$_H$	PROFINET IO device or module signaling access to an illegal area
0	1	80B6$_H$	PROFINET IO device or module denies access
0	1	80B8$_H$	The module is signaling an illegal parameter
0	1	80C0$_H$	The data record cannot be read
0	1	80C1$_H$	The specified data record is being processed
0	1	80C2$_H$	Too many jobs pending
0	1	80C3$_H$	Resources (memory) occupied
0	1	80C4$_H$	Communication error (occurs temporarily, it is usually best to repeat the job in the user program).

3.2.8 FB54 PNIO_ALARM

3.2.8.1 Meaning and call - PNIO_ALARM

How It works

FB54 is used for alarm evaluation by a CP 3431 operating as PROFINET IO controller and should be called in its user program when the ADD_INFO parameter in FC12 is not equal to 0. After complete and errorfree transfer of all OUTPUT parameters of FB54, the received alarms are acknowledged automatically.

The alarms are forwarded to the user program in the chronological order in which they were signaled. Older alarms that have not yet been signaled to the user program and that become invalid due to more recent alarms are not deleted by the newer alarms.

Note

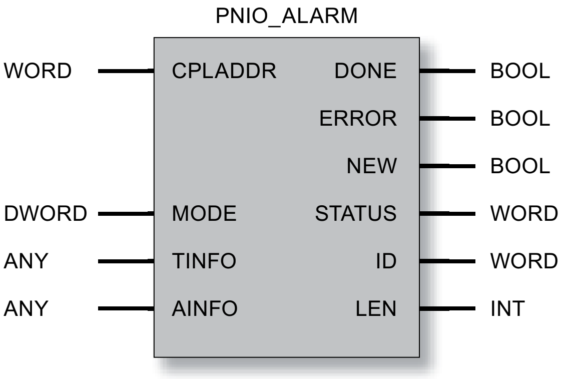
As long as the block has not yet been called, the alarms are acknowledged automatically in the CP.

If FB54 has been called (at least) once in the user program, it must continue to be called to acknowledge pending alarms. This is the situation when FC12 signals a value not equal to "0" in the ADD_INFO parameter.

If FB54 is no longer called after it has been called once or more in the user program, alarms are not acknowledged and there is no guarantee that the IO image will be updated correctly. The can occur, for example, following a station return alarm. The need to call FB54 can only be reset by restarting the CP (power cycle).

Call interface

Call interface in FBD representation



Example in STL representation:

```
STL                                          Explanation
CALL FB 54, DB 54(                           //PNIO_ALARM block call
                                             // (read/write data record)
CPLADDR:=W#16#0110,                          //Module address from hardware configuration
DONE:=M 1.1,                                 //Address for return parameter DONE
ERROR:=M 1.2,                                //Address for return parameter ERROR
NEW:=M 1.3,                                  //TRUE: A new alarm was received
STATUS:=MW 12,                               //Error code of the SFB or PNIOCtrl
ID:=MW14,                                    //Logical base address of the component
                                             //(module or submodule) from which an alarm
                                             //was received
LEN:=MW 16,                                  //Length of the received alarm information (AINFO)
MODE:=MD 18,                                 //RESERVED (value always = 0)
TINFO:=P#DB4.DBX0.0 BYTE 32,                 //(task information) destination area for OB
                                             //start information and management information;
                                             // fixed length 32 bytes of diagnostic information
AINFO:= P#DB4.DBX32.0 BYTE 532 );            //(alarm information) destination area for
                                             //header information and additional alarm information
```

3.2.8.2 Explanation of the formal parameters - PNIO_ALARM

Explanation of the formal parameters

The following table explains all the formal parameters for FB54:

Parameter	Declaration	Data type	Possible values	Description
CPLADDR	INPUT	WORD	-	Start address of the module that caused the error
DONE	OUTPUT	BOOL	0: - 1: Alarm information transferred successfully	This parameter indicates whether or not the job was completed without errors. If DONE = 1, the NEW parameter must also be checked.
ERROR	OUTPUT	BOOL	0: - 1: Error	Error code
NEW	OUTPUT	BOOL	0: Data transfer active or no new alarm 1: New alarm received and acknowledged	If DONE = 1 and NEW = 1, a new received alarm is signaled.
STATUS	OUTPUT	WORD	0: No error Different value: Error (see Table 1–14)	Status code
ID	OUTPUT	WORD		Logical start address of the PNIO component that triggers the alarm (module or submodule). For an output module, bit 15 is set (example of output address 5: ID:=DW#16#8005). For a mixed module, the lower of the two addresses is specified.
LEN	OUTPUT	INT		Length of the received alarm information (AINFO)
MODE	IN_OUT	DWORD	0	Reserved

Parameter	Declaration	Data type	Possible values	Description
TINFO	IN_OUT	ANY (as VARTYPE , BYTE, WORD and DWORD are permitted)	The address of the data area points to one of the alternatives: • Memory bit area • Data block area The length of the ANY pointer must be >= 32 bytes.	task information Destination area for the alarm management information. The error OB start information (OB header = byte 0...19 of TINFO) is reproduced as far as possible by the CP firmware. See also 1)
AINFO	IN_OUT	ANY (as VARTYPE , BYTE, WORD and DWORD are permitted)	The address of the data area points to one of the alternatives: • Memory bit area • Data block area The length of the ANY pointer must be greater than or equal to the maximum additional alarm information that can be expected, maximum 1432 bytes (see LEN parameter)	alarm information Destination area for header information and additional alarm information. If the ANY pointer AINFO is too low, the information will be truncated. See also 1)

1) Reference Manual "STEP 7 - System and Standard Functions for S7-300 and S7-400", receiving an alarm with SFB54 "RALRM"

3.2.8.3 Condition codes of PNIO_ALARM

Condition codes

The following table shows the condition codes formed by the DONE, NEW, ERROR and STATUS parameters that must be evaluated by the user program.

Note

For entries with the coding 8FxxH under STATUS, note the information in the Reference Manual "STEP 7 - System and Standard Functions for S7-300 and S7-400". The chapter describing error evaluation with the RET_VAL output parameter contains detailed information.

DONE	NEW	ERROR	STATUS	Meaning
0	0	0	8180$_H$	Data transfer active
1	1	0	0000$_H$	Alarm data successfully transferred and alarm acknowledged
1	0	0	0000$_H$	No alarm data exist

DONE	NEW	ERROR	STATUS	Meaning
0	0	1	8183$_H$	• No PNIO controller configuration, • wrong CPLADDR or • CP in STOP mode
0	0	1	8184$_H$	System error or illegal parameter type
0	0	1	8185$_H$	Destination buffer (TINFO or AINFO) is too short
0	0	1	8F22$_H$	Area length error reading a parameter (e.g. DB too short)
0	0	1	8F23$_H$	Area length error writing a parameter (e.g. DB too short)
0	0	1	8F24$_H$	Area error reading a parameter
0	0	1	8F25$_H$	Area error writing a parameter
0	0	1	8F28$_H$	Orientation error when reading a parameter
0	0	1	8F29$_H$	Alignment error writing a parameter
0	0	1	8F30$_H$	Parameter is in the write-protected first active data block
0	0	1	8F31$_H$	Parameter is in the write-protected second active data block
0	0	1	8F32$_H$	The DB number in the parameter is too high
0	0	1	8F3A$_H$	Destination area not loaded (DB)
0	0	1	8F42$_H$	Timeout reading a parameter from the I/O area
0	0	1	8F43$_H$	Timeout writing a parameter to the I/O area
0	0	1	8F44$_H$	Address of the parameter to be read is disabled in the accessed rack
0	0	1	8F45$_H$	Address of the parameter to be written is disabled in the accessed rack
0	0	1	8F7F$_H$	Internal error, e.g. illegal ANY reference
0	0	1	8090$_H$	Module with this address does not exist
0	0	1	80A0$_H$	Negative acknowledgment reading from the module
0	0	1	80A1$_H$	Negative acknowledgment writing to the module
0	0	1	80B0$_H$	Module does not recognize the data record
0	0	1	80B1$_H$	• The specified data record length is incorrect or • The CP changes to STOP
0	0	1	80C0$_H$	The data record cannot be read
0	0	1	80C1$_H$	The specified data record is being processed
0	0	1	80C2$_H$	Too many jobs pending
0	0	1	80C3$_H$	Resources (memory) occupied
0	0	1	80C4$_H$	Communication error (occurs temporarily, it is usually best to repeat the job in the user program).

3.2.9 Configuration limits / resources required by the FCs and FBs (PROFINET)

Required resources

NOTICE
Please note the version information of the blocks. Blocks with other versions have different resource requirements.

Table 3-5 Information for FCs / FBs with an S7-400

NAME	Version	FC/FB no.	Load memory bytes	Work memory bytes	MC7 bytes	Local data bytes
PN_InOut	1.3	FB88	2678	2234	2198	48
PN_InOut_Fast	1.0	FB90	2906	2266	2230	48

Table 3-6 Information for FCs / FBs with an S7-300

NAME	Version	FC/FB no.	Load memory bytes	Work memory bytes	MC7 bytes	Local data bytes
PN_InOut	1.5	FB88	2470	2066	2030	54
PNIO_SEND	1.0	FC11	1272	1058	1022	42
PNIO_SEND	2.0	FC11	1342	1116	1080	42
PNIO_RECV	1.0	FC12	1122	928	892	42
PNIO_RECV	2.0	FC12	1192	986	950	42
PNIO_RW_REC	1.1	FB52	1636	1378	1342	62
PNIO_ALARM	1.1	FB54	1168	960	924	62

FCs / FBs for PROFIBUS 4

4.1 FCs for S5-compatible communication (SEND/RECEIVE interface)

4.1.1 Overview of FCs and their use

Overview

The following FCs are available for the SEND/RECEIVE interface for transferring data on configured FDL connections:

FC	Can be used with 1)		Meaning
	S7-300	S7-400	
AG_SEND (FC5)	x	x	for sending data
AG_RECV (FC6)	x	x	for receiving data
AG_LSEND (FC50)		x	for sending data
AG_LRECV (FC60)		x	for receiving data

1) Notes on the FCs for an S7300 and S7400

To ensure the compatibility of PROFIBUS and Ind. Ethernet on the interface in the user program, you can use the FCs AG_LSEND and AG_LRECV on PROFIBUS as alternatives to AG_SEND and AG_RECV. There is no difference in the interface or the way they function. On PROFIBUS, however, you can only transfer data up to a maximum of 240 bytes even with these FCs although they are intended for longer data records on Industrial Ethernet.

This is only possible if the block type and block version are permitted for the CP type you are using.

With the S7 CPs for S7300, only the FCs AG_SEND and AG_RECV are used; on Industrial Ethernet even for the transfer of longer data records.

The manuals contain information on the compatibility of the S7-CPs and the corresponding blocks (FCs / FBs). You will find an overview of the versions of the FCs/FBs in the documentation and block history.

Application

The following diagram illustrates the use of the FCs AG_SEND / AG_LSEND and AG_RECV / AG_LRECV for bi-directional data transfer on one configured FDL connection.
With certain connection types, a job header should be included in the user data area.

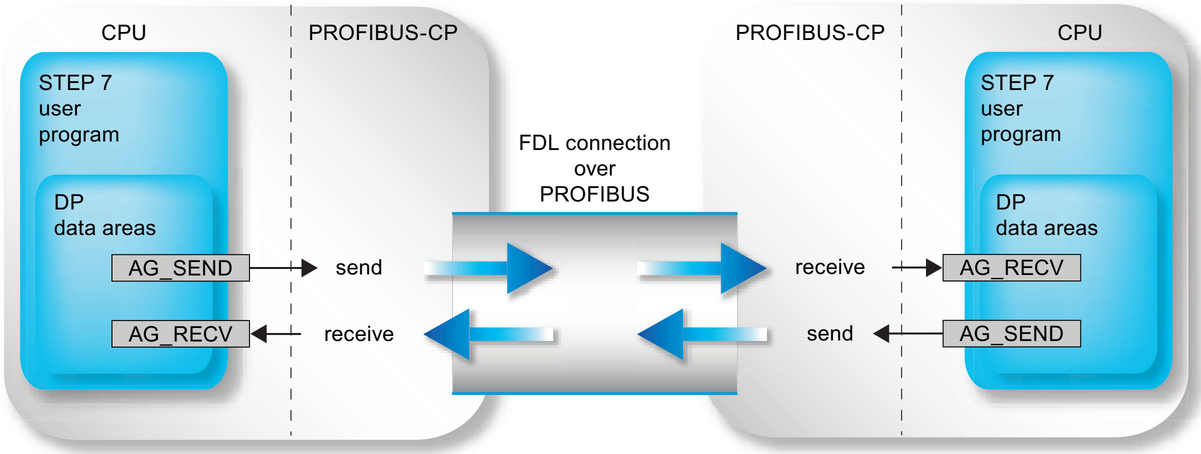


Figure 4-1 Using AG_SEND and AG_RECV on both communications partners

Application without job header

With a specified FDL connection, the address and job parameters are specified by the configuration of the connection. The user program only provides the user data in the FDL data area when sending with AG_SEND / AG_LSEND or receives the data with AG_RECV / AG_LRECV.

Up to 240 bytes of user data can be transferred. This applies to PROFIBUS for both the AG_SEND or AG_LSEND functions.

Working with the job header

The following connection types require a job header in the FDL (user) data area:

- Unspecified FDL connection with free layer 2 access
- FDL connection with broadcast
- FDL connection with multicast

The following schematic illustrates the structure of the job buffer and the meaning and location of the parameters in the job header.

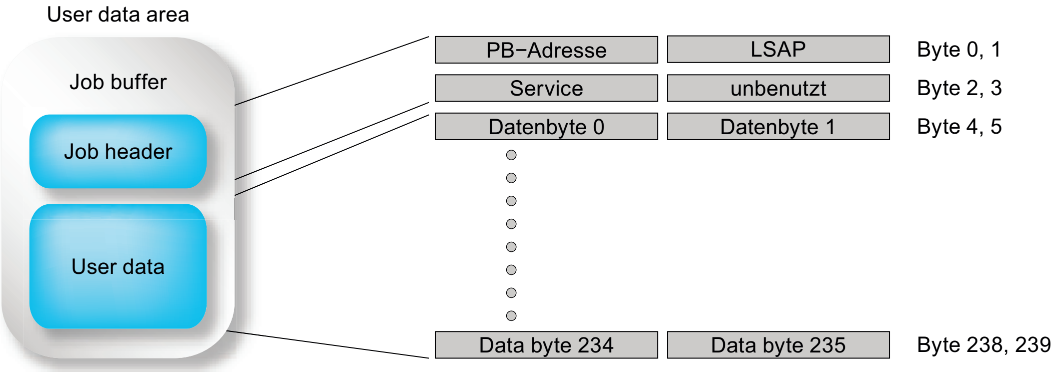


Figure 4-2 Sending and receiving via an FDL connection with programmed broadcast addressing

|The user data area can be up to 240 bytes. Up to 236 bytes of user data can be transferred. 4 bytes are reserved for the job header.

Please note that the data length specified in the block call (LEN parameter) must include the header and the user data!

4.1.2 FC5 AG_SEND / FC50 AG_LSEND

4.1.2.1 Meaning and call - AG_SEND / AG_LSEND

Meaning of the block

FC AG_SEND / AG_LSEND transfers data to the PROFIBUS CP for transmission on a configured FDL connection.

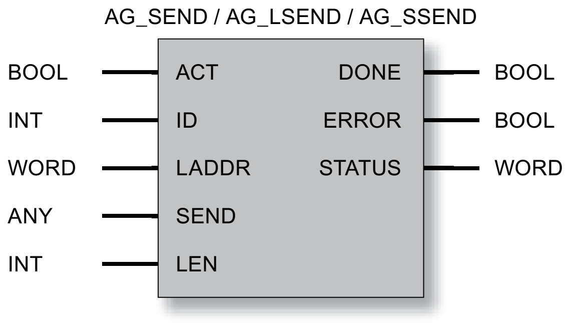
The selected data area can be a process image area, a memory bit area or a data block area.

Error free execution of the function is indicated when the entire FDL data area could be sent on PROFIBUS.

Note:
Unless otherwise stated, all the following information applies equally to the FCs AG_SEND and AG_LSEND.

Call

Call interface in FBD representation



Example in STL representation

```
STL                                           Explanation
call fc 5(                                    //AG_SEND / AG_LSEND block call
ACT:=M 20.0,                                  //Job triggered by memory bit
ID:=MW 22,                                    //Connection ID acc. to configuration
LADDR:=W#16#0100,                             //=LADDR 256 dec. in HW Config
SEND:= P#db99.dbx10.0 byte 240,               //Buffer with send data
LEN:=MW 24,                                   //Length for send data
DONE:=M 20.1,                                 //Execution code
ERROR:=M 20.2,                                //Error code
STATUS:=MW 26 );                              //Status code
```

Calls with job header

The following table shows the connection types and job types for which parameters must be supplied in the job header.

The job header is located in the FDL (user) data area. It occupies the first 4 bytes and must be added to the length specified in the LEN parameter. The maximum user data length is therefore reduced for jobs with a job header to 236 bytes.

Table 4-1 Supplying the job header in the user data area

Parameter	FDL connection type		
	Unspecified: free layer 2 2)	Broadcast	Multicast
PB address	Address of the destination station Range of values: 0..126 depending on node / 127 for broadcast/multicast	For AG_SEND no relevance; but area must be reserved.	For AG_SEND no relevance; but area must be reserved.
LSAP	LSAP of the destination station Range of values: 0..62 depending on node / 63 for broadcast	No significance but area must be reserved.	No significance but area must be reserved.

	FDL connection type		

Service [1]	SDA (Send Data with Acknowledge):
	Value: 00_H
	SDN (Send Data with No Acknowledge):
	Value: 01_H
		No significance but area must be reserved.	No significance but area must be reserved.

[1] for broadcast and multicast, only the SDN service is possible.

[2] The information on broadcast and multicast in this column is relevant only when an unspecified FDL connection is used for broadcast or multicast. On a configured FDL connection (recommended application) with broadcast or multicast as the connection partner, the address parameters are assigned automatically according to the configuration.

4.1.2.2 How AG_SEND / AG_LSEND work

Operating principle

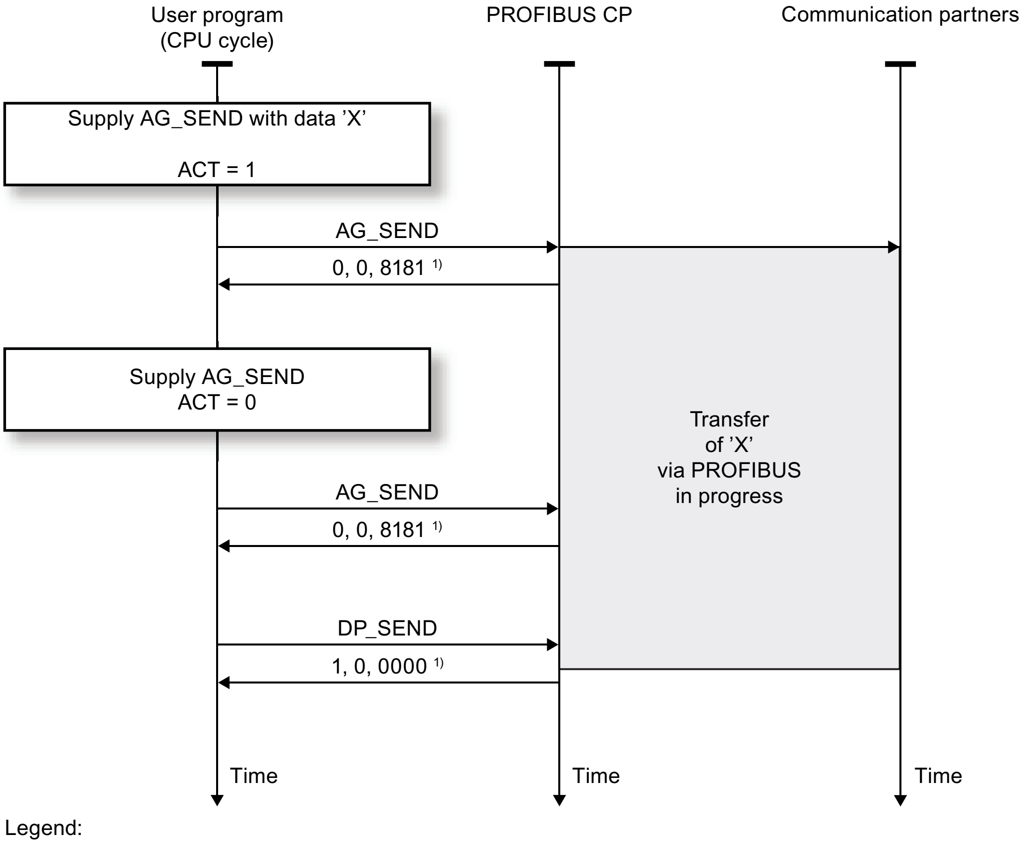
The following diagram illustrates the normal sequence of data transmission triggered in the user program using AG_SEND.

The send job in executed as soon as the parameter ACT = 1 is passed.

Following this, the parameter ACT = 0 must be passed in at least one further call.

The status code in the output parameters DONE, ERROR and STATUS is updated in each block call and can be evaluated. To update the status code without starting a new send job, start a new block call with the parameter ACT = 0.

Refer to the sample program at the end of this section.



Legend:
1) Parameter transfer DONE, ERROR, STATUS

4.1.2.3 Explanation of the formal parameters - AG_SEND / AG_LSEND

Explanation of the formal parameters

The following table explains all the formal parameters for the AG_SEND /AG_LSEND functions:

Parameter	Declaration	Data type	Possible values	Description
ACT	INPUT	BOOL	0,1	If an FC is called with ACT=1, LEN bytes are sent from the ISO transport data area specified with the SEND parameter. If an FC is called with ACT = 0, the status codes DONE, ERROR and STATUS are updated.
ID	INPUT	INT	1,2...64 (S7-400) 1,2...16 (S7-300)	The connection number of the FDL connection is specified in the parameter ID.
LADDR	INPUT	WORD		Module start address When you configure the CP, the module start address is displayed in the configuration table. Specify this address here.
SEND	INPUT	ANY (only the following are permitted as VARTYPE: WORD and DWORD are permitted)		Specifies the address and length The address of the data area points to one of the alternatives: • PI area • Memory bit area • Data block area With a call with job header, the FDL data area contains the job header and the user data.
LEN	INPUT	INT	1,2, to 240 (or up to "length specified for SEND parameter")	Number of bytes to be sent from the FDL data area with this job. The possible values range from 1 to length specified for the SEND parameter. In a call, with job header, the length information is made up of the job header (4 bytes) + user data (1 to 236 bytes). Therefore LEN >= 4 !
DONE	OUTPUT	BOOL	0: - 1: new data	The status parameter indicates whether or not the job was completed without errors. For the meaning in conjunction with the ERROR and STATUS parameters, refer to the following table.
ERROR	OUTPUT	BOOL	0: - 1: Error	Error code For the meaning in conjunction with the DONE and STATUS parameters, refer to the following table.
STATUS	OUTPUT	WORD	See following table	Status code For the meaning in conjunction with the DONE and ERROR parameters, refer to the following table.

4.1.2.4 Condition codes of AG_SEND and AG_LSEND

Condition codes

The following table shows the condition codes formed based on DONE, ERROR and STATUS that must be evaluated by the user program.

Note

For entries coded with 8Fxx$_H$ in STATUS, refer to the information in the STEP 7 Standard and System Functions reference manual. The chapter describing error evaluation with the RET_VAL output parameter contains detailed information.

To find out which SFCs are used and are relevant for error evaluation, display the properties dialog of the FC described here in the "Calls" tab.

Table 4-2 AG_SEND condition codes

DONE	ERROR	STATUS	Meaning
1	0	0000$_H$	Job completed without error.
0	0	0000$_H$	No job being executed.
0	0	8181$_H$	Job active.
0	1	7000$_H$	The condition code is possible only with S7-400: The FC was called with ACT=0; the job has not yet been processed.
0	1	8183$_H$	No configuration or the FDL service has not yet started on the PROFIBUS CP.
0	1	8184$_H$	• Illegal data type specified for the SEND parameter. • FDL connection without job buffer: System error. • FDL connection with job buffer: Parameter LEN<4 or illegal parameter in job header (with free layer 2 access).
0	1	8185$_H$	LEN parameter longer than SEND source area.
0	1	8186$_H$	ID parameter invalid. ID!=1,2 to 15,16.
0	1	8301$_H$	SAP not activated on destination station.
0	1	8302$_H$	No receive resources on the destination station; the receiving station cannot process received data quickly enough or has not prepared any receive resources.
0	1	8303$_H$	The PROFIBUS service (SDA Send Data with Acknowledge) is not supported on this SAP by the destination station. This condition code can also occur temporarily when connections or gateways are downloaded "in RUN".
0	1	8304$_H$	The FDL connection is not established.
0	1	8311$_H$	The destination station is not obtainable at the specified PROFIBUS address or the service is not possible for the specified PROFIBUS address.
0	1	8312$_H$	PROFIBUS error on the CP: for example, bus short-circuit, own station not in ring.
0	1	8315$_H$	Internal parameter error on an FDL connection with job header: Parameter LEN<4 or illegal parameter in job header (with free layer 2 access).
0	1	8F22$_H$	Source area invalid, e.g.: Area does not exist in the DB LEN parameter < 0

DONE	ERROR	STATUS	Meaning
0	1	8F24$_H$	Area error reading a parameter.
0	1	8F28$_H$	Alignment error reading a parameter.
0	1	8F32$_H$	Parameter contains a DB number that is too high.
0	1	8F33$_H$	DB number error.
0	1	8F3A$_H$	Area not loaded (DB).
0	1	8F42$_H$	Timeout reading a parameter from the I/O area.
0	1	8F44$_H$	Address of the parameter to be read is disabled in the access track.
0	1	8F7F$_H$	Internal error, e.g. illegal ANY reference e.g. parameter LEN=0
0	1	8090$_H$	• No module with this module start address exists. • The FC being used does not match the system family being used (remember to use different FCs for S7300 and S7400).
0	1	8091$_H$	Module start address not at a doubleword boundary.
0	1	8092$_H$	In the ANY reference, a type other than BYTE is specified. (S7-400 only)
0	1	80A4$_H$	The communication bus connection between the CPU and CP is not established. (with newer CPU versions). This can, for example, be caused by the following: • No connection configuration; • The maximum number of CPs that can be operated at one time has been exceeded (for further information, refer to the CP manual).
0	1	80B0$_H$	The module does not recognize the data record.
0	1	80B1$_H$	• Destination area is invalid. The amount of data to be sent exceeds the upper limit permitted for this service (e.g. destination area > 240 bytes).
0	1	80B2$_H$	The communication bus connection between the CPU and CP is not established (with older CPU versions; otherwise 80A4$_H$; for further information, refer to this code)
0	1	80C0$_H$	The data record cannot be read.
0	1	80C1$_H$	The specified data record is currently being processed.
0	1	80C2$_H$	There are too many jobs pending.
0	1	80C3$_H$	Resources occupied (memory).
0	1	80C4$_H$	Communication error (occurs temporarily, it is usually best to repeat the job in the user program).
0	1	80D2$_H$	Module start address incorrect.

4.1.3 FC6 AG_RECV / FC60 AG_LRECV

4.1.3.1 Meaning and call - AG_RECV / AG_LRECV

Meaning of the block

The AG_RECV / AG_LRECV function receives the data transferred on a configured FDL connection from the PROFIBUS CP.

The data area specified for the receive data can be a process image area, a bit address area or a data block area.

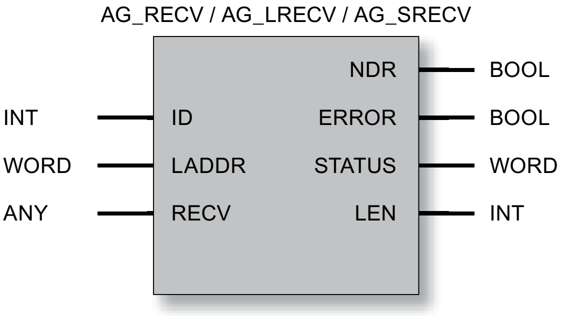
Errorfree execution is indicated when the data could be received from the PROFIBUS CP.

Note:

Unless otherwise stated, all the following information applies equally to the FCs AG_SEND and AG_LSEND.

Call interface

Call interface in FBD representation



Example in STL representation

STL	Explanation
call fc 6(	//AG_RECV / AG_LRECV block call
ID:=MW 30,	//Connection ID acc. to configuration
LADDR:=W#16#0100,	//=LADDR 256 dec.in HW Config
RECV:=P#M 10.0 BYTE 100,	//Buffer for received data
NDR:=DB 100.DBX 0.6,	//Receive code
ERROR:=DB 100.DBX 0.7,	//Execution code
STATUS:=DB 100.DBW 2,	//Error code
LEN:=DB 100.DBW 4);	//Status code

Calls with job header

Table 4-3 Return parameters in the job header in the FDL (user) data area

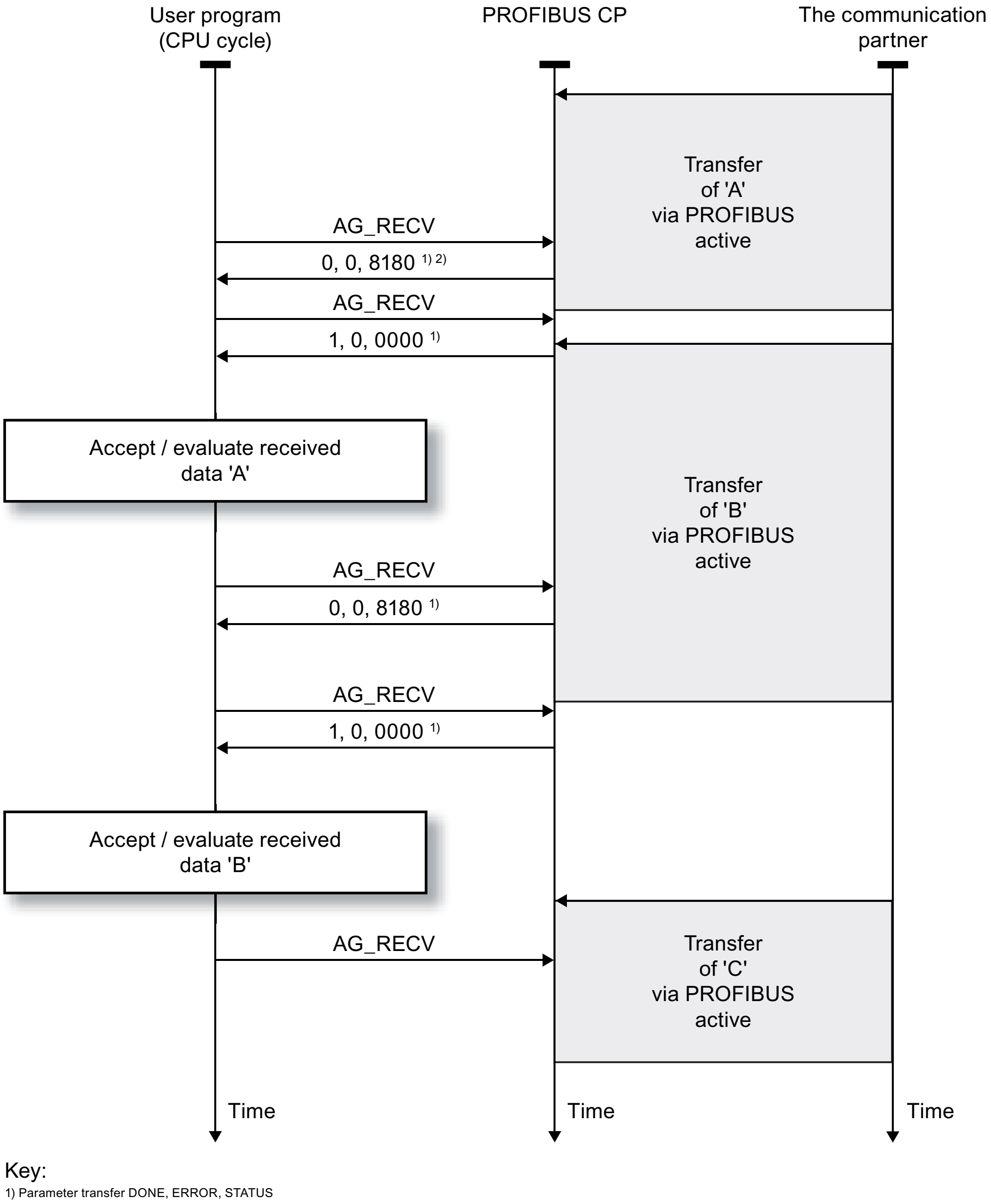
Parameter	FDL connection type		
	Unspecified: free layer 2)	**Broadcast**	**Multicast**
PB address	Address of the sender Values: 0 to 126 depending on node		
LSAP	LSAP of the sender Values: 0 to 63 depending on node		
Service	SDN indication (Send Data with No Acknowledge - Indication): Value: 01_H or SDA indication (Send Data with Acknowledge - Indication): Value: 00_H	SDN indication (Send Data with No Acknowledge - Indication): Value: $7F_H$	SDN indication (Send Data with No Acknowledge - Indication): Value: $7F_H$

4.1.3.2 How AG_RECV / AG_LRECV work

Operating principle

The following diagram illustrates the normal sequence of data acceptance triggered by an AG_RECV in the user program.

Each AG_RECV job in the user program is acknowledged by the Ethernet CP with an entry in the output parameters NDR, ERROR and STATUS.


User program
(CPU cycle)
PROFIBUS CP
The communication partner
Transfer of 'A' via PROFIBUS active
AG_RECV
0, 0, 8180 1) 2)
AG_RECV
1, 0, 0000 1)
Accept / evaluate received data 'A'
Transfer of 'B' via PROFIBUS active
AG_RECV
0, 0, 8180 1)
AG_RECV
1, 0, 0000 1)
Accept / evaluate received data 'B'
AG_RECV
Transfer of 'C' via PROFIBUS active
Time
Time
Time
Key:
1) Parameter transfer DONE, ERROR, STATUS

4.1.3.3 Explanation of the formal parameters - AG_RECV / AG_LRECV

Explanation of the formal parameters

The following table explains all the formal parameters for the AG_RECV / AG_LRECV function:

Parameter	Declaration	Data type	Possible values	Description
ID	INPUT	INT	1,2...16 (S7-300) 1,2...32 (S7-400)	The connection number of the FDL connection is specified in the parameter ID.
LADDR	INPUT	WORD		Module start address When you configure the CP, the module start address is displayed in the configuration table. Specify this address here.
RECV	INPUT	ANY (only the following are permitted as VARTYPE: WORD and DWORD are permitted)		Specifies the address and length The address of the FDL data area points to one of the alternatives: • PI area • Memory bit area • Data block area With a call with job header, the FDL data area contains the job header and the user data.
LEN	OUTPUT	INT	1,2,...240	Specifies the number of bytes to be received in the FDL data area from the PROFIBUS CP. In a call, with job header, the length information is made up of the job header (4 bytes) + user data (1 to 236 bytes). Therefore LEN >= 4 !
NDR	OUTPUT	BOOL	0: - 1: new data	This parameter indicates whether new data were received. For the meaning in conjunction with the ERROR and STATUS parameters, refer to the following table.
ERROR	OUTPUT	BOOL	0: - 1: Error	Error code For the meaning of this parameter in conjunction with the NDR and STATUS parameters, refer to the following table.
STATUS	OUTPUT	WORD	See following table	Status code For the meaning in conjunction with the NDR and ERROR parameters, refer to the following table.

4.1.3.4 Condition codes of AG_RECV and AG_LRECV

Condition codes

The following table shows the codes formed by the NDR, ERROR and STATUS parameters that must be evaluated by the user program.

Note

For entries coded with 8FxxH in STATUS, refer to the information in the STEP 7 Standard and System Functions reference manual. The chapter describing error evaluation with the RET_VAL output parameter contains detailed information.

To find out which SFCs are used and are relevant for error evaluation, display the properties dialog of the FC described here in the "Calls" tab.

Table 4-4 AG_RECV / AG_LRECV condition codes

NDR	ERROR	STATUS	Meaning
1	0	0000_H	New data accepted.
0	0	8180_H	• There is no data available yet. The configuration is missing or the FDL service has not started on the PROFIBUS CP (occurs here instead of the code 0,1,8183$_H$).
0	0	8181_H	Job active.
0	1	8183_H	No configuration or the FDL service has not yet started on the PROFIBUS CP.
0	1	8184_H	• Illegal data type specified for the RECV parameter. • System error.
0	1	8185_H	Destination buffer (RECV) is too short.
0	1	8186_H	ID parameter invalid. ID!=1,2 to 15,16.
0	1	8303_H	The PROFIBUS service (SDA - Send Data with Acknowledge) is not supported on this SAP. This condition code can also occur temporarily when connections or gateways are downloaded "in RUN".
0	1	8304_H	The FDL connection is not established.
0	1	$8F23_H$	Source area invalid, e.g.: Area does note exist in the DB.
0	1	$8F25_H$	Area error writing a parameter.
0	1	$8F29_H$	Alignment error writing a parameter
0	1	$8F30_H$	Parameter is in the writeprotected 1st current data block.
0	1	$8F31_H$	Parameter is in the writeprotected 2nd current data block.
0	1	$8F32_H$	Parameter contains a DB number that is too high.
0	1	$8F33_H$	DB number error.
0	1	$8F3A_H$	Destination area not loaded (DB).
0	1	$8F43_H$	Timeout writing a parameter to the I/O area.
0	1	$8F45_H$	Address of the parameter to be read is disabled in the access track.

NDR	ERROR	STATUS	Meaning
0	1	$8F7F_H$	Internal error, e.g. illegal ANY reference.
0	1	8090_H	• No module with this module start address exists. • The FC being used does not match the system family being used (remember to use different FCs for S7300 and S7400).
0	1	8091_H	Module start address not at a doubleword boundary.
0	1	8092_H	In the ANY reference, a type other than BYTE is specified. (S7-400 only)
0	1	$80A0_H$	Negative acknowledgment reading from the module.
0	1	$80A4_H$	The communication bus connection between the CPU and CP is not established. (with newer CPU versions). This can, for example, be caused by the following: • No connection configuration; • The maximum number of CPs that can be operated at one time has been exceeded (for further information, refer to the CP manual).
0	1	$80B0_H$	The module does not recognize the data record.
0	1	$80B1_H$	• Destination area invalid. The destination area is too short.
0	1	$80B2_H$	The communication bus connection between the CPU and CP is not established.
0	1	$80C0_H$	The data record cannot be read.
0	1	$80C1_H$	The specified data record is currently being processed.
0	1	$80C2_H$	There are too many jobs pending.
0	1	80C3H	Resources occupied (memory).
0	1	$80C4_H$	Communication error (occurs temporarily, it is usually best to repeat the job in the user program).
0	1	$80D2_H$	Module start address incorrect.

4.2 FCs for DP (distributed I/O) with S7-300

4.2.1 Overview of FCs and their use

Overview

The following FCs are available for the DP master and DP slave modes with an S7300:

FC	can be used with:		Meaning
	DP master	DP slave	
DP_SEND (FC1)	X	X	for sending data
DP_RECV (FC2)	X	X	for receiving data
DP_DIAG (FC3)	X	-	for diagnostic functions initiated by the DP master
DP_CTRL (FC4)	X	-	for control functions

Application

The following diagram illustrates the use of the DP_SEND and DP_RECV FCs on the DP master and DP slave.

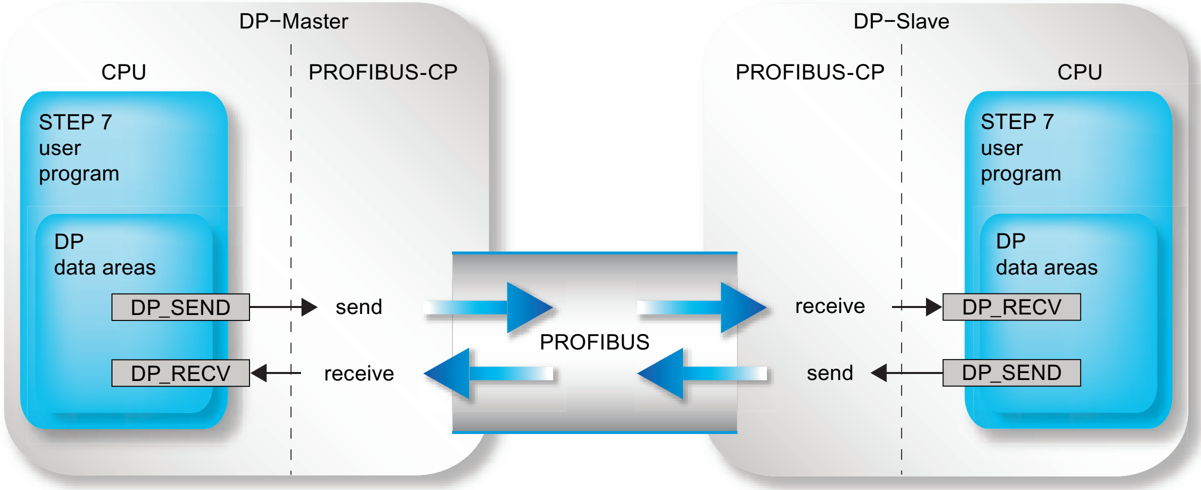


Figure 4-3 Using the FCs DP_SEND and DP_RECV with DP master and DP slave

4.2.2 FC1 DP_SEND

4.2.2.1 Meaning and call - DP_SEND

Meaning

FC DP_SEND transfers data to the PROFIBUS CP. Depending on the mode of the PROFIBUS CP, DP_SEND has the following significance:

- When used in the DP master

The block transfers the data of a specified DP output area to the PROFIBUS CP for output to the distributed I/O system.

- When used in the DP slave

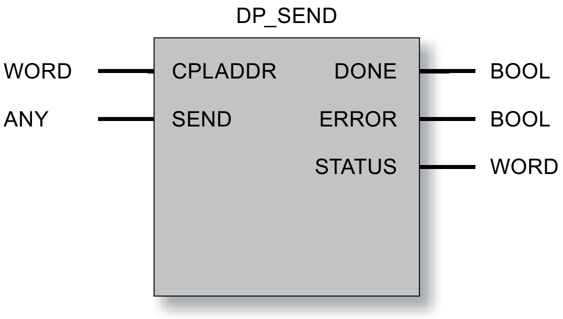
The block transfers the input data of the DP slave to the PROFIBUS CP for transfer to the DP master

The selected data area can be a process image area, a memory bit area or a data block area.

Correct execution is signaled when the entire DP data area could be accepted by the PROFIBUS CP.

Note that FC DP_SEND must then be called successfully at least once for the DP slave in the user program if inputs were configured for this slave. Please read the information in the manual as well.

Call interface



Example of a call in STL representation

STL	Explanation

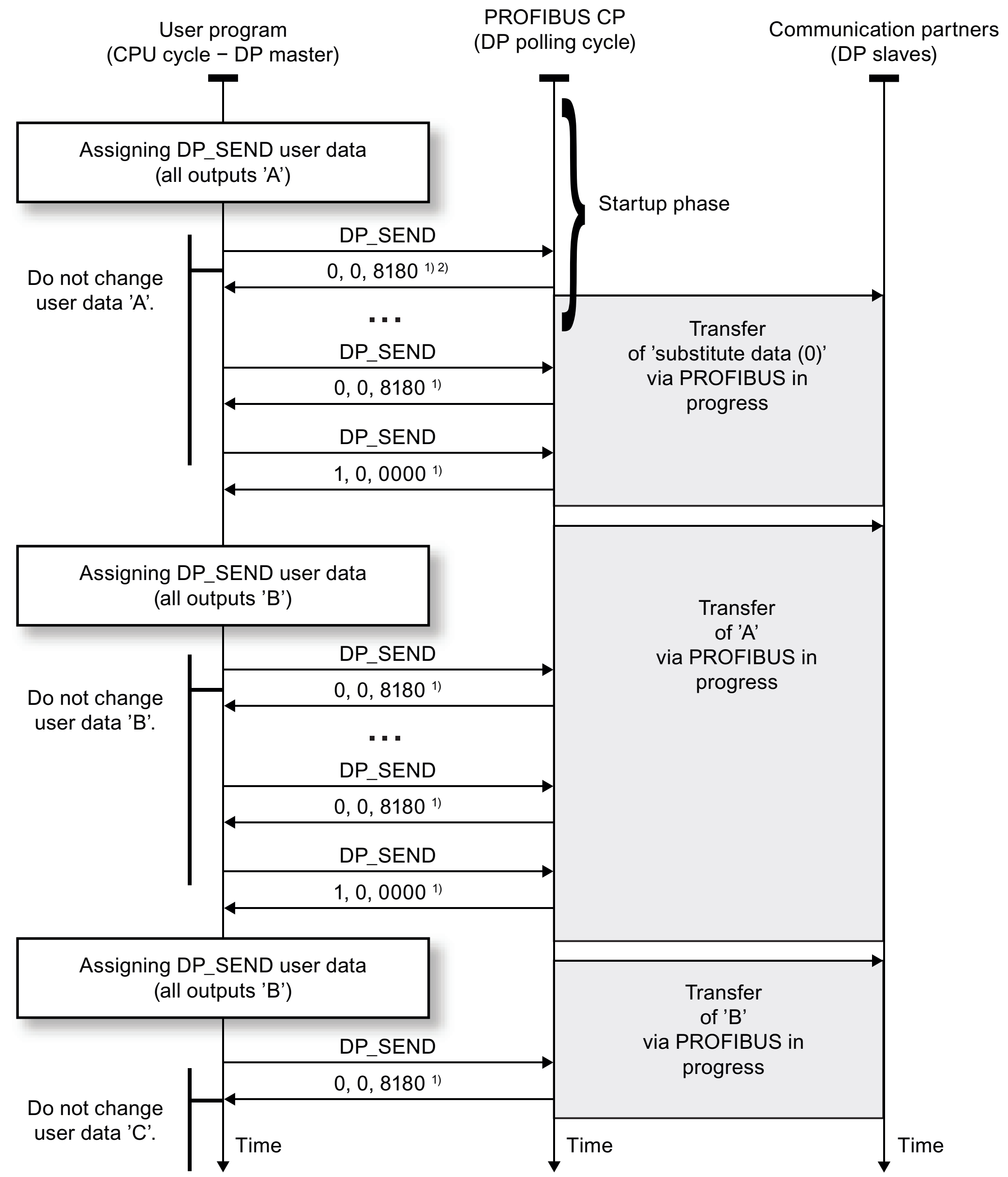
call fc 1(
CPLADDR:=W#16#0120,
SEND:=P#db17.dbx0.0 byte 103,
DONE:=M 99.1,
ERROR:=M 99.0,
STATUS:=MW 104);
	//DP_SEND function call

4.2.2.2 How DP_SEND works

Operating principle

The following flow diagram illustrates the normal sequence of data transfer triggered with DP_SEND in the user program.

Each DP_SEND job in the user program is acknowledged by the PROFIBUS CP setting values in the DONE, ERROR and STATUS output parameters.



Legend:
1) Parameter transfer DONE, ERROR, STATUS
2) Display 8183H is available during startup for older CP types

Guarantee of data transfer

The diagram also shows that with the confirmation DONE=1, ERROR=0 and STATUS=0000, data transfer to the communications partner is functioning correctly.

The latest send data transferred to the PROFIBUS CP is always passed on to the communications partner. For this reason, new user data must only be entered in the send buffer following a positive acknowledgment (DONE=1, ERROR=0, STATUS=0000).

4.2.2.3 Explanation of the formal parameters - DP_SEND

Explanation of the formal parameters

The following table explains all the formal parameters for the DP_SEND function:

Parameter	Declaration	Data type	Possible values	Description
CPLADDR	INPUT	WORD		Module start address When you configure the CP, the module start address is displayed in the configuration table. Specify this address here.
SEND	INPUT	ANY (only the following are permitted as VARTYPE: With FC1 as of V3: BYTE With FC1 up to V2.x: BYTE, WORD and DWORD)		Specifies the address and length The address of the DP data area points to one of the alternatives: • PI area • Memory bit area • Data block area The length must be set for • DP master: 1...21600 • DP slave: 1...240
DONE	OUTPUT	BOOL	0: - 1: new data	The status parameter indicates whether or not the job was completed without errors. For the meaning in conjunction with the ERROR and STATUS parameters, refer to ""..
ERROR	OUTPUT	BOOL	0: - 1: Error	Error code For the meaning in conjunction with the DONE and STATUS parameters, refer to "".
STATUS	OUTPUT	WORD	See " "	Status code For the meaning in conjunction with the DONE and ERROR parameters, refer to "".

4.2.2.4 Condition codes of the DP_SEND block

Condition codes

The following table shows the condition codes formed based on DONE, ERROR and STATUS that must be evaluated by the user program.

Note

For entries coded with 8FxxH in STATUS, refer to the information in the STEP 7 Standard and System Functions reference manual. The chapter describing error evaluation with the RET_VAL output parameter contains detailed information.

To find out which SFCs are used and are relevant for error evaluation, display the properties dialog of the FC described here in the "Calls" tab.

Table 4-5 DP_SEND condition codes

DONE	ERROR	STATUS	Meaning
0	0	8180_H	DP not started due to: • CP STOP or • "no parameter assignment" (occurs here instead of the code 0,1,8183_H).
1	0	0000_H	New data transferred without error.
0	1	8183_H	No configuration or the DP service has not yet started on the PROFIBUS CP.
0	1	8184_H	System error or bad parameter type.
0	1	$8F22_H$	Area length error reading a parameter (e.g. DB too short).
0	1	$8F23_H$	Area length error writing a parameter (e.g. DB too short).
0	1	$8F24_H$	Area error reading a parameter.
0	1	$8F25_H$	Area error writing a parameter.
0	1	$8F28_H$	Alignment error reading a parameter.
0	1	$8F29_H$	Alignment error writing a parameter.
0	1	$8F30_H$	Parameter is in the writeprotected 1st current data block.
0	1	$8F31_H$	Parameter is in the writeprotected 2nd current data block.
0	1	$8F32_H$	Parameter contains a DB number that is too high.
0	1	$8F33_H$	DB number error.
0	1	$8F3A_H$	Destination area not loaded (DB).
0	1	$8F42_H$	Timeout reading a parameter from the I/O area.
0	1	$8F43_H$	Timeout writing a parameter to the I/O area.
0	1	$8F44_H$	Address of the parameter to be read is disabled in the access track.
0	1	$8F45_H$	Address of the parameter to be written is disabled in the access track.
0	1	$8F7F_H$	Internal error, e.g. illegal ANY reference.
0	1	8090_H	No module with this address exists.
0	1	8091_H	Logical base address not at a double word boundary.
0	1	$80A1_H$	Negative acknowledgment writing to the module.

DONE	ERROR	STATUS	Meaning
0	1	$80B0_H$	The module does not recognize the data record.
0	1	$80B1_H$	The number of data bytes to be sent exceeds the upper limit for this service (applies to DP master and DP slave mode).
0	1	$80C0_H$	The data record cannot be read.
0	1	$80C1_H$	The specified data record is currently being processed.
0	1	$80C2_H$	There are too many jobs pending.
0	1	$80C3_H$	Resources occupied (memory).
0	1	$80C4_H$	Communication error (occurs temporarily, it is usually best to repeat the job in the user program).
0	1	$80D2_H$	Logical base address incorrect.

4.2.3 FC2 DP_RECV

4.2.3.1 Meaning and call - DP_RECV

Meaning

The DP_RECV function (FC) receives data over PROFIBUS. DP_RECV has the following significance depending on the mode of the PROFIBUS CP:

- When used in the DP master
  DP_RECV receives the process data from the distributed I/O along with status information and enters this in a specified DP input area.
- When used on the DP slave
  DP_RECV accepts the output data transferred by the DP master in the DP data area specified in the block.

The data area specified for the receive data can be a process image area, a bit address area or a data block area.

Errorfree execution of the function is signaled when the entire DP data input area could be transferred by the PROFIBUS CP.

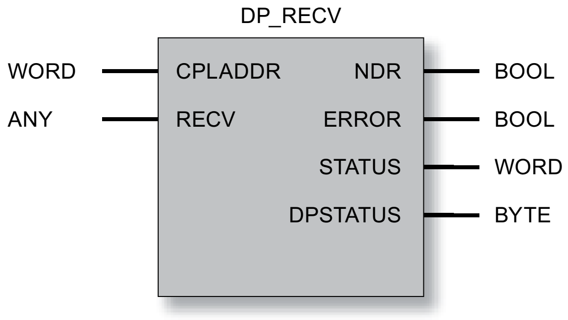
Note that FC DP_RECV must be called successfully at least once on the DP slave in the user program if output data was configured for this DP slave. Please read the information in the manual.

Additional task: Entering the status byte

The DP_RECV function has the following additional task:

- Updating the DP status byte DPSTATUS. This means that DP_RECV handles tasks for DP diagnostics
  If no receive data is configured, DP_RECV must be called with a length of 1 to update the DPSTATUS status byte (applies only to DP masters; with DP slaves, the status byte cannot be read without data).
  Please read the information in the manual as well.
- Enabling the station list (see DP_DIAG).

Call interface



Example in STL representation

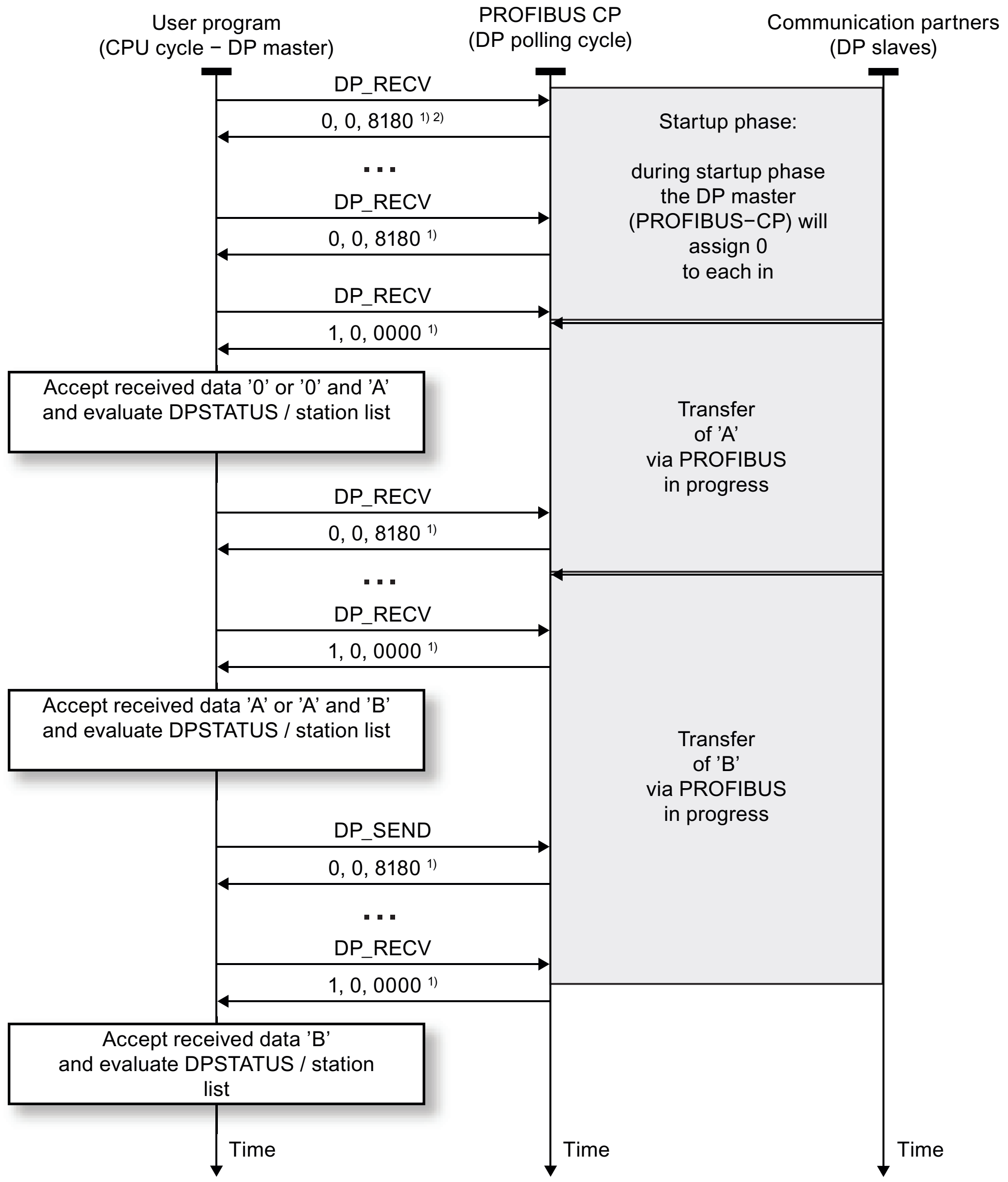
STL	Explanation
`call fc 2(`	`//DP_RECV function call`
`CPLADDR:=W#16#0120,`	
`RECV:= P#db17.dbx240.0 byte 103,`	
`NDR:=M 99.1,`	
`ERROR:=M 99.0,`	
`STATUS:=MW 104,`	
`DPSTATUS:=MB 0 );`	

4.2.3.2 How DP_RECV works

Operating principle

The following flow diagram illustrates the normal sequence of data transfer triggered with DP_RECV in the user program.

Each DP_RECV job in the user program is acknowledged by the PROFIBUS CP setting values in the NDR, ERROR and STATUS output parameters.


User program
(CPU cycle − DP master)
PROFIBUS CP
(DP polling cycle)
Communication partners
(DP slaves)
DP_RECV
0, 0, 8180 1) 2)
...
DP_RECV
0, 0, 8180 1)
DP_RECV
1, 0, 0000 1)
Startup phase:
during startup phase the DP master (PROFIBUS−CP) will assign 0 to each in
Accept received data ’0’ or ’0’ and ’A’ and evaluate DPSTATUS / station list
Transfer of ’A’ via PROFIBUS in progress
DP_RECV
0, 0, 8180 1)
...
DP_RECV
1, 0, 0000 1)
Accept received data ’A’ or ’A’ and ’B’ and evaluate DPSTATUS / station list
Transfer of ’B’ via PROFIBUS in progress
DP_SEND
0, 0, 8180 1)
...
DP_RECV
1, 0, 0000 1)
Accept received data ’B’ and evaluate DPSTATUS / station list
Time
Time
Time

Legend:

1) Parameter transfer DONE, ERROR, STATUS

2) Display 8183H is available during startup for older CP types

Guarantee of data acceptance

The diagram also shows that the confirmation NDR=1, ERROR=0 and STATUS=0000 indicates reliable data reception. Requirement: The DP master and the DP slaves are in the data transfer phase.

Note the following:

- In DP master mode:
  If a DB slave is not in the data transfer phase, the corresponding received data is set to 0.
  If the DP master is neither in the RUN nor CLEAR state (bits 4 and 5 in DPSTATUS), all the received data is set to 0.
  If data has been received from the DP slave several times since the last DP_RECV function call, only the last received data is fetched with the next DP_RECV.
- In DP slave mode:
  If the DP slave is not in the data transfer phase (bit 1 in DPSTATUS) or the DP master is in the CLEAR state (bit 2 in DPSTATUS), the received data is set to 0.
  If data has been received from the DP master several times since the last DP_RECV function call, only the last received data is fetched with the next DP_RECV.

4.2.3.3 Explanation of the formal parameters - DP_RECV

Explanation of the formal parameters

The following table explains all the formal parameters for the function DP_RECV:

Parameter	Declaration	Data type	Possible values	Description
CPLADDR	INPUT	WORD		Module start address When you configure the CP, the module start address is displayed in the configuration table. Specify this address here.
RECV	INPUT	ANY (only the following are permitted as VARTYPE: With FC1 as of V3: BYTE With FC1 up to V2.x: BYTE, WORD and DWORD)		Specifies the address and length The address of the DP data area points to one of the alternatives: • PI area • Memory bit area • Data block area The length must be set for: • DP master: 1...2160 • DP slave: 1...240 • DP master; only read status byte: 1 (see also CP manual)
NDR	OUTPUT	BOOL	0: - 1: New data accepted	The status parameter indicates whether or not new data was accepted. For the meaning in conjunction with the ERROR and STATUS parameters, refer to "".
ERROR	OUTPUT	BOOL	0: - 1: Error	Error code For the meaning in conjunction with the NDR and STATUS parameters, refer to "".

Parameter	Declaration	Data type	Possible values	Description
STATUS	OUTPUT	WORD	See ""	Status code For the meaning in conjunction with the NDR and ERROR parameters, refer to "".
DPSTATUS	OUTPUT	Byte	For coding, see below under DPSTATUS	DP status code

4.2.3.4 Condition codes of the DP_RECV block

Condition codes

The following table shows the codes formed by the NDR, ERROR and STATUS parameters that must be evaluated by the user program.

Note

For entries coded with 8Fxx$_H$ in STATUS, refer to the information in the STEP 7 Standard and System Functions reference manual. The chapter describing error evaluation with the RET_VAL output parameter contains detailed information.

To find out which SFCs are used and are relevant for error evaluation, display the properties dialog of the FC described here in the "Calls" tab.

NDR	ERROR	STATUS	Meaning
0	0	8180$_H$	DP not started due to: • CP STOP or • "no parameter assignment" (occurs here instead of the code 0,1,8183$_H$).
1	0	0000$_H$	New data accepted without error.
0	1	8183$_H$	No configuration or the DP service has not yet started on the PROFIBUS CP.
0	1	8184$_H$	System error or bad parameter type.
0	1	8F22$_H$	Area length error reading a parameter (e.g. DB too short).
0	1	8F23$_H$	Area length error writing a parameter (e.g. DB too short).
0	1	8F24$_H$	Area error reading a parameter.
0	1	8F25$_H$	Area error writing a parameter.
0	1	8F28$_H$	Alignment error reading a parameter.
0	1	8F29$_H$	Alignment error writing a parameter.
0	1	8F30$_H$	Parameter is in the writeprotected 1st current data block.
0	1	8F31$_H$	Parameter is in the writeprotected 2nd current data block.
0	1	8F32$_H$	Parameter contains a DB number that is too high.
0	1	8F33$_H$	DB number error.
0	1	8F3A$_H$	Destination area not loaded (DB).

NDR	ERROR	STATUS	Meaning
0	1	8F42$_H$	Timeout reading a parameter from the I/O area.
0	1	8F43$_H$	Timeout writing a parameter to the I/O area.
0	1	8F44$_H$	Address of the parameter to be read is disabled in the access track.
0	1	8F45$_H$	Address of the parameter to be read is disabled in the access track.
0	1	8F7F$_H$	Internal error, e.g. illegal ANY reference.
0	1	8090$_H$	No module with this address exists.
0	1	8091$_H$	Logical base address not at a double word boundary.
0	1	80A0$_H$	Negative acknowledgment writing to the module.
0	1	80B0$_H$	The module does not recognize the data record.
0	1	80B1$_H$	The number of data bytes to be sent exceeds the upper limit for this service (applies to DP master and DP slave mode).
0	1	80C0$_H$	The data record cannot be read.
0	1	80C1$_H$	The specified data record is currently being processed.
0	1	80C2$_H$	There are too many jobs pending.
0	1	80C3$_H$	Resources occupied (memory).
0	1	80C4$_H$	Communication error (occurs temporarily, it is usually best to repeat the job in the user program).
0	1	80D2$_H$	Logical base address incorrect.

4.2.3.5 DPSTATUS - DP_RECV

DPSTATUS

The coding of the DPSTATUS output parameter is different for the DP master mode and DP slave mode.

DP master mode

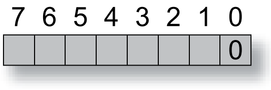


Table 4-6 Meaning of the bits in DPSTATUS in DP master mode

Bit	Meaning
7	not used
6	This bit is not set. Please read the information in the manual as well.

Bit	Meaning
5,4	Values for DPSTATUS of the DP master: 00 RUN 01 CLEAR 10 STOP (this is now the OFFLINE mode) 11 OFFLINE Please read the information in the manual as well.
3	Value 1: Cyclic synchronization is active
2	Value 0: No new diagnostic data exists Value 1: Evaluation of diagnostic list useful; at least one station has new diagnostic data
1	Value 0: All DP slaves are in the data transfer phase Value 1: Evaluating the station list is useful
0	DP mode Value 0: DP master mode The other bits only have the specified meaning when this bit is not set.

DP slave mode

7	6	5	4	3	2	1	0
							1

Table 4-7 Meaning of the bits in DPSTATUS in DP slave mode

Bit	Meaning
7-5	not used
4	This bit is not set. Please read the information in the manual as well.
3	This bit is not set. Please read the information in the manual as well.
2	Value 1: DP master 1 is in the CLEAR mode. The DP slave receives the value 0 in the DP data intended for the outputs. This has no effect on the send data.
1	Value 1: The configuration/parameter assignment is not yet completed.
0	Value 1: DP slave mode. **The other bits only have the specified meaning when this bit is set.**

NOTICE

Please note, that DPSTATUS must not be evaluated until the return parameter NDR=1 is set.

4.2.4 FC3 DP_DIAG

4.2.4.1 Meaning and call - DP_DIAG

Meaning of the block

FC DP_DIAG is used to request diagnostic information. The following types of job are possible:

- Request DP station list
- Request DP diagnostics list;
- Request DP single status;
- Read input/output data of a DP slave acyclically
- Read older DP single diagnostic information
- Read DP status.
- Read DP mode for PLC/CP stop
- Read current status of the DP slave.

Diagnostics data can also be requested for a specific slave by specifying a station address.

To transfer the diagnostic data to the CPU, you should reserve a memory area in the CPU and specify this area in the call. This memory area can be a data block area or a bit memory area. The maximum length of the available memory area must also be specified in the job.

Note

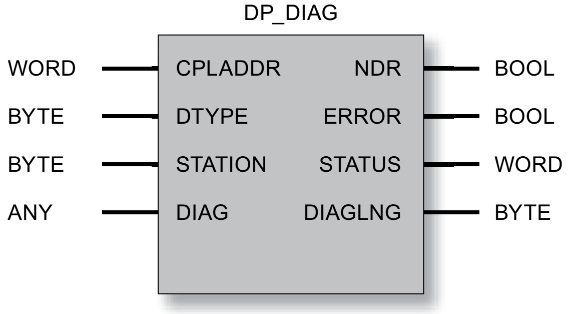
FC DP_DIAG is only of practical use in the DP master mode.

Exclusion

As long as this block is running, it must not be supplied with new job data.

Exception: Requesting the DP station list or DP diagnostics list.

Call interface



Example in STL representation

STL	Explanation

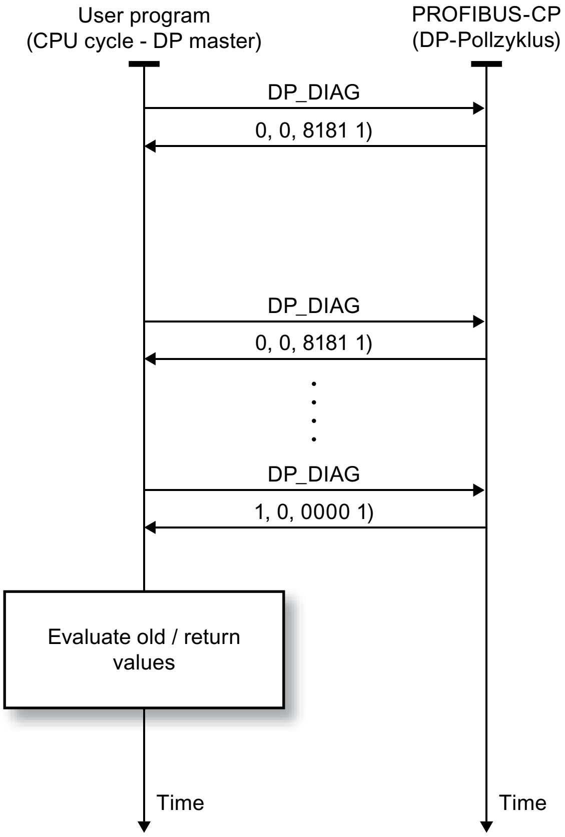
call fc 3(
CPLADDR:=W#16#0120,
DTYPE:=B#16#00,
STATION:=B#16#03,
DIAG:=P#db18.dbx0.0 byte 16,
NDR:=M 70.0,
ERROR:=M 70.1,
STATUS:=MW 72,
DIAGLNG:=MB 20);
	//DP_DIAG function call

4.2.4.2 How DP_DIAG works

Sequence / handling on the call interface

The DP_DIAG function call is processed during cyclic execution of the user program as follows:

The job is triggered with the first call. Diagnostic data is only returned in the acknowledgment of one of the subsequent calls.



1) Parameter transfer NDR, ERROR, STATUS

Note

Please note the following special feature of the job types read_DP_station_list and read_DP_diagnostic_list:

- The diagnostic job supplies the diagnostic data available at the time of the last DPRECV call. Reading a list prevents the data from being read out again (return value 0x8182).
- The lists are released again after a new diagnostic event followed by a DPRECV call.

After calling DP_DIAG, you obtain information indicating one of the situations below:

- NDR=0, ERROR=0, STATUS=8181
  As long as the code combination NDR=0, ERROR=0 and STATUS=8181 is set, the job parameters must not be modified.
- NDR=1
  The parameter value NDR=1 indicates that valid diagnostic data is available. Additional information is possible in the STATUS parameter.
- NDR=0, ERROR=1
  An error has occurred. The diagnostic data is invalid. The error message is located in STATUS.

4.2.4.3 Explanation of the formal parameters - DP_DIAG

Explanation of the formal parameters

The following table explains all the formal parameters for the function DP_DIAG:

Parameter	Declaration	Data type	Possible values	Description
CPLADDR	INPUT	WORD		Module start address When you configure the CP, the module start address is displayed in the configuration table. Specify this address here.
DTYPE	INPUT	BYTE	0: Station list 1: Diagnostic list 2: Current diagnostic info 3: Older diagnostic info 4: Read status 5: Read status for CPU STOP 6: Read status for CP STOP 7: Read input data (acyclically) 8: Read output data (acyclically) 10: Read current status of the DP slave	Diagnostic type
STATION	INPUT	BYTE		Station address of the DP slave

Parameter	Declaration	Data type	Possible values	Description
DIAG	INPUT	ANY (only the following are permitted as VARTYPE: BYTE, WORD and DWORD)	The length must be set from 1 to 240	Specifies the address and length Address of the data area. References the following alternatives: • PI area • Memory bit area • Data block area Note: If more diagnostic data exist than can be entered in the DIAG area, only as much data as specified in the DIAG length will be transferred. The actual length is indicated in DIAGLNG.
NDR	OUTPUT	BOOL	0: - 1: new data	This parameter indicates whether or not new data were accepted. For the meaning in conjunction with the ERROR and STATUS parameters, refer to "".
ERROR	OUTPUT	BOOL	0: - 1: Error	Error code For the meaning in conjunction with the NDR and STATUS parameters, refer to "".
STATUS	OUTPUT	WORD	See list	Status code For the meaning in conjunction with the NDR and ERROR parameters, refer to "".
DIAGLNG	OUTPUT	BYTE	See list	This contains the actual length (in bytes) of the data made available by the PROFIBUS CP, regardless of the buffer size specified in the DIAG parameter.

4.2.4.4 Job types - DP_DIAG

Job types

The following overview of the specifications for DTYPE, STATION and DIAGLNG shows the permitted or useful entries.

Table 4-8 Job types for DP_DIAG

DTYPE	Corresponds to job	Parameter STATION	DIAGLNG	Acknowledgement code (contained in the STATUS parameter; shown in Table "DP_DIAG codes")
0	Read DP station list	---	- ignored -	Module start address When you configure the CP, the module start address is displayed in the configuration table. Specify this address here.
1	Read DP diagnostic list	---	- ignored -	The DP diagnostics list informs the CPU program about the DP slaves with new diagnostics data.

DTYPE	Corresponds to job	Parameter STATION	DIAGLNG	Acknowledgement code (contained in the STATUS parameter; shown in Table "DP_DIAG codes")
2	Read current DP single diagnostic data	1...126	>=6	The current DP single diagnostics informs the CPU program of the current diagnostic data of a DP slave.
3	Read older DP single diagnostic information	1...126	>=6	The older DP single diagnostics informs the CPU program of the older diagnostic data of a DP slave. This data is stored on the PROFIBUS CP and read according to the "last in - first out" principle in the ring buffer. The structure of the ring buffer is explained below. If changes occur quickly in the DP slave diagnostic data, this function allows the diagnostic data of a DP slave to be acquired and evaluated in the CPU program of the DP master.
4	Read the requested operating mode with the DP-CTRL job (CYTPE=4)		>=0	With this job, you can read the DP status. The following statuses are possible: : • RUN • CLEAR • STOP (now the OFFLINE mode) • OFFLINE Please read the information in the manual as well.
5	Read DP status for CPU STOP		>=0	With this job you can find out the DP status to which the PROFIBUS CP changes if the CPU changes to STOP: • RUN • CLEAR • STOP (now the OFFLINE mode) • OFFLINE As default, the PROFIBUS CP changes to the DP status CLEAR if the CPU changes to STOP. Please read the information in the manual as well.
6	Read DP status for CP STOP		>=0	With this job you can find out the DP status to which the PROFIBUS CP changes if the CP changes to STOP: • STOP (now the OFFLINE mode) • OFFLINE As default, the PROFIBUS CP changes to the DP status OFFLINE if the CP changes to STOP. Please read the information in the manual as well.
7	Read input data	1...126	>=1	With this job, the DP master (class 2) reads the input data of the DP slave. This function is also known as shared input.
8	Read output data	1...126	>=1	With this job, the DP master (class 2) reads the output data of a DP slave. This function is also known as shared output.

DTYPE	Corresponds to job	Parameter STATION	DIAGLNG	Acknowledgement code (contained in the STATUS parameter; shown in Table "DP_DIAG codes")
10	Read current status of the DP slave	1...126	>=0	With this job, you can read out the current status of the DP slave. The following statuses are possible : • The DP master exchanges data with the DP slave cyclically. • The DP master reads the input data of the DP slave cyclically. • The DP master reads the output data of the DP slaves cyclically. • The DP master is not currently processing this DP slave cyclically.

4.2.4.5 Ring buffer for diagnostics data - DP_DIAG

Ring Buffer for Diagnostic Data

The following diagram illustrates how diagnostic data is read using the "read older DP single diagnostic data" function. The first access reads the **most recent of the older diagnostic data**.

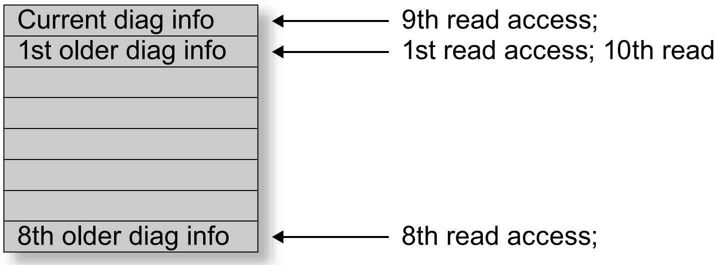


Figure 4-4 Ring Buffer for Diagnostic Data

When the current diagnostic data is read out, the read pointer is reset to the first older diagnostic data.

4.2.4.6 Condition codes of DP_DIAG

Condition codes

The following table shows the codes formed by the NDR, ERROR and STATUS parameters that must be evaluated by the user program.

Note

For entries coded with $8Fxx_H$ in STATUS, refer to the information in the STEP 7 Standard and System Functions reference manual. The chapter describing error evaluation with the RET_VAL output parameter contains detailed information.

To find out which SFCs are used and are relevant for error evaluation, display the properties dialog of the FC described here in the "Calls" tab.

Table 4-9 DP_DIAG codes

NDR	ERROR	STATUS	Possible with DTYPE	Meaning
0	0	8181_H	2-10	Job active. DP master not started due to • CP STOP or • "no parameter assignment" (occurs here instead of the code 0,1,8183_H).
0	0	8182_H	0	Triggering job pointless. DP master not started due to • CP STOP or • "no parameter assignment" (occurs here instead of the code 0,1,8183_H).
0	0	8182_H	1	No new diagnostic data exist. DP master not started due to • CP STOP or • "no parameter assignment" (occurs here instead of the code 0,1,8183_H).
1	0	0000_H	0-10	Job completed without error.
1	0	8222_H	7,8	Job completed without error. The length of the DP slave data that was read is not the same as the data length expected by the DP master based on the module list of the DP slave in the CP database.
1	0	8227_H	7,8	Job completed without error. Message: No data exists.
1	0	8231_H	4,5,6	Job completed without error. Message: The DP status is already "RUN"
1	0	8232_H	4,5,6	Job completed without error. Message: The DP status is already "CLEAR"
1	0	8233_H	4,5,6	Job completed without error. Message: The DP status is already STOP The STOP status is now the OFFLINE status (here code 8234_H). Please read the information in the manual as well.
1	0	8234_H	4,5,6	Job completed without error. Message: The DP status is already "OFFLINE"

NDR	ERROR	STATUS	Possible with DTYPE	Meaning
1	0	823A$_H$	2,3,7,8	Job completed without error. Message: 241 or 242 bytes of data were read. 240 bytes of data are available.
1	0	8241$_H$	2,3,10	Job completed without error. Message: The specified DP slave was not configured.
1	0	8243$_H$	2,3,10	Job completed without error. Message: The module list of the DP slave in the CP database only contains empty modules.
1	0	8245$_H$	2,3,10	Job completed without error. Message: The DP slave is in the "read input data cyclically" mode.
1	0	8246$_H$	2,3,10	Job completed without error. Message: The DP slave is in the "read output data cyclically" mode.
1	0	8248$_H$	2,3,10	Job completed without error. Message: The module list of the DP slave in the CP database contains input, output or I/O modules.
1	0	8249$_H$	2,3,10	Job completed without error. Message: The DP slave is deactivated due to a DP mode change (e.g. CP mode selector set to STOP).
1	0	824A$_H$	2,3,10	Job completed without error. Message: The DP slave is deactivated due to a DP_CTRL job in the CPU program.
0	1	8090$_H$	0-10	Logical base address of the module is invalid
0	1	80B0$_H$	0-10	The module does not recognize the data record or is changing from RUN --> STOP.
0	1	80B1$_H$	0-10	Specified data record length incorrect
0	1	80C0$_H$	0-10	Data record cannot be read
0	1	80C1$_H$	0-10	The specified data record is being processed
0	1	80C2$_H$	0-10	Too many jobs pending
0	1	80C3$_H$	0-8	Resources (memory) occupied
0	1	80C4$_H$	0-10	Communication error
0	1	80D2$_H$	0-10	Logical base address wrong
0	1	8183$_H$	0-10	DP master not configured.
0	1	8184$_H$	0-8	System error or bad parameter type.
0	1	8311$_H$	>=2	DTYPE parameter outside range of values.
0	1	8313$_H$	2,3,7,8,10	STATION parameter outside range of values.
0	1	8321$_H$	>=2	The DP slave is not providing any valid data.
0	1	8326$_H$	7,8	The DP slave has more than 242 bytes of data available. The PROFIBUS CP supports a maximum of 242 bytes.
0	1	8335$_H$	7,8	The PROFIBUS CP is in PROFIBUS status: "Station not in ring".
0	1	8341$_H$	2,3,7,8,10	The specified slave was not configured
0	1	8342$_H$	7,8	The DP slave with the PROFIBUS address specified in the STATION parameter is not obtainable.
0	1	8349$_H$	7,8	The DP master is in the OFFLINE mode.
0	1	8F22$_H$	0-10	Area length error reading a parameter (e.g. DB too short)

NDR	ERROR	STATUS	Possible with DTYPE	Meaning
0	1	8F23$_H$	0-10	Area length error writing a parameter (e.g. DB too short)
0	1	8F24$_H$	0-10	Area error reading a parameter
0	1	8F25$_H$	0-10	Area error writing a parameter
0	1	8F28$_H$	0-10	Alignment error reading a parameter
0	1	8F29$_H$	0-10	Alignment error writing a parameter
0	1	8F30$_H$	0-10	Parameter is in the writeprotected 1st current data block
0	1	8F31$_H$	0-10	Parameter is in the writeprotected 2nd current data block
0	1	8F32$_H$	0-10	The DB number in the parameter is too high
0	1	8F33$_H$	0-10	DB number error
0	1	8F3A$_H$	0-10	Area not loaded (DB)
0	1	8F42$_H$	0-10	Timeout reading a parameter from the I/O area
0	1	8F43$_H$	0-10	Timeout writing a parameter to the I/O area
0	1	8F44$_H$	0-10	Address of the parameter to be read locked in the access track
0	1	8F45$_H$	0-10	Address of the parameter to be written is disabled in the access track
0	1	8F7F$_H$	0-10	Internal error, e.g. illegal ANY reference

4.2.5 FC4 DP_CTRL

4.2.5.1 Meaning and call - DP_CTRL

Meaning of the block

FC DP_CTRL transfers control jobs to the PROFIBUS CP. You specify a job field (CONTROL parameter) to specify the control job in greater detail.

The following types of job are possible:

- Global control acyclic/cyclic;
- Delete older diagnostic data;
- Set current DP mode;
- Set DP mode for PLC/CP STOP;
- Read input/output data cyclically;
- Set the operating mode of the DP slave.

There are restrictions relating to the job types listed here (please refer to the information in the manual for the module).

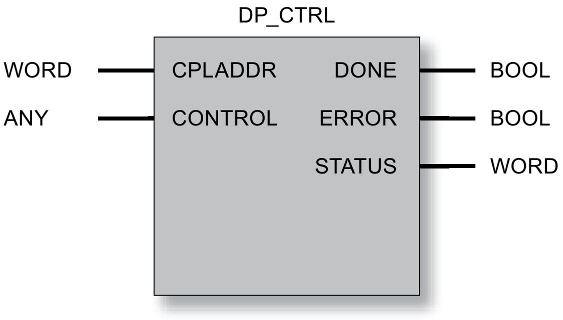
Note

FC DP_CTRL is only of practical use in the DP master mode.

Connector

As long as this block is running, it must not be supplied with new job data.

Call interface



Example in STL representation

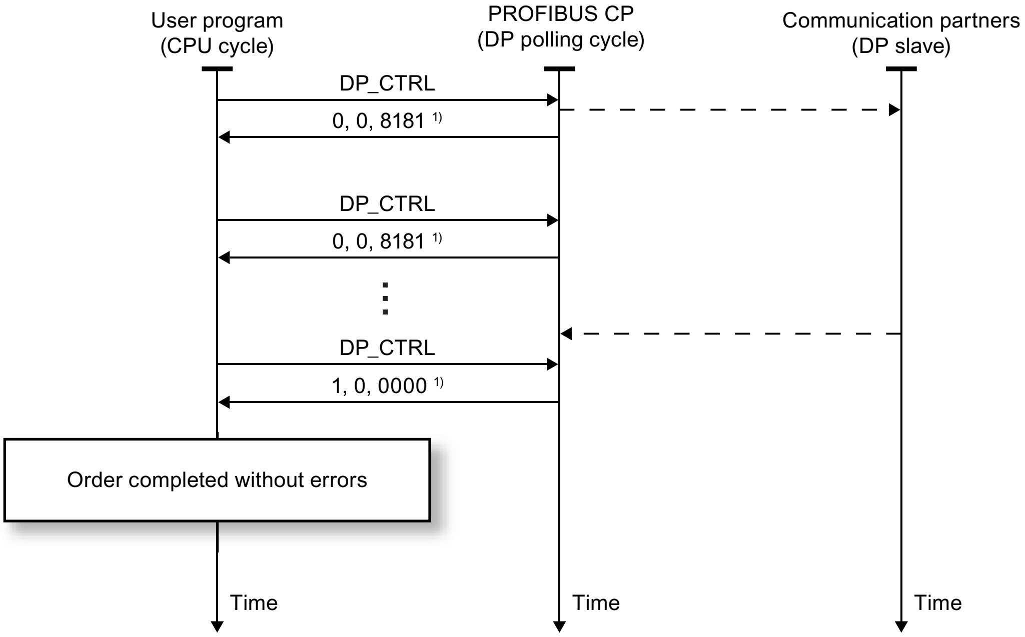
STL	**Explanation**
call fc 4(	//DP_CTRL block call
CPLADDR:=W#16#0120,	
CONTROL:=P#db14.dbx0.0 byte 30,	// The buffer for the control job // occupies the first 30 bytes in DB 14.
DONE:=M 70.0,	
ERROR:=M 70.1,	
STATUS:=MW 72);	

4.2.5.2 How DP_CTRL works

Sequence / handling on the call interface

The DP_CTRL function call is processed within the cyclic execution of the user program as shown below:

The job is triggered with the first call. Diagnostic data is only returned in the acknowledgment of one of the subsequent calls.



Legend:
1) Parameter transfer DONE, ERROR, STATUS

After calling DP_CTRL, you obtain one of the following condition code patterns as the reaction:

- DONE=0, ERROR=0, STATUS=8181
  As long as the code combination DONE=0, ERROR=0 and STATUS=8181 is set, the job parameters must not be modified.
- DONE=1
  The parameter value DONE=1 indicates that the job was executed. Additional information is possible in the STATUS parameter.
- DONE=0, ERROR=1
  An error has occurred. The error message is located in STATUS.

4.2.5.3 Explanation of the formal parameters - DP_CTRL

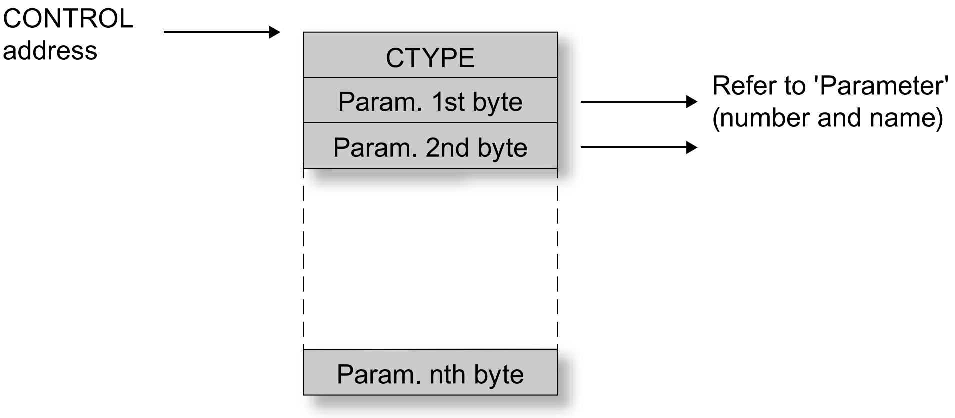
Explanation of the formal parameters

The following table explains all the formal parameters for the DP_CTRL function:

Parameter	Declaration	Data type	Possible values	Description
CPLADDR	INPUT	WORD		Module start address When you configure the CP, the module start address is displayed in the configuration table. Specify this address here.
CONTROL	INPUT	ANY (only the following are permitted as VARTYPE: BYTE, WORD and DWORD)	The length must be set from 1 to 240	Specifies the address and length of the CONTROL job field Address of the data area. References the following alternatives: • PI area • Memory bit area • Data block area The length must be at least as long as the number of parameters.
DONE	OUTPUT	BOOL	0: - 1: Job executed without error.	Indicates whether the job was sent and completed without errors. For the meaning in conjunction with the ERROR and STATUS parameters, refer to the following table.
ERROR	OUTPUT	BOOL	0: - 1: Error	Error code For the meaning in conjunction with the DONE and STATUS parameters, refer to the follow table.
STATUS	OUTPUT	WORD	See following table 'Return Codes'	Status code The following table shows the condition codes formed by the DONE and ERROR parameters.

Structure of the CONTROL job field

The control job has the following structure:



Example of the job field

With a job field as shown below, a cyclic global control job SYNC and Unfreeze is sent for group 4 and group 5 without the autoclear option.

DB 14

Byte 0	01H	CTYPE
Byte 1	24H	Command Mode
Byte 2	18H	Group Select
Byte 3	00H	Autoclear

The length in the ANY pointer must be at least 4 (in the example, 30 has been selected).

4.2.5.4 Job types - DP_CTRL

Job types

Permitted or feasible specifications for the job are shown in the following overview based on the specification for CTYPE and the information in the job field.

CTYPE	**Corresponds to job**	**Parameter in job field**		**Description**
		Name	Quantity	
0	Trigger global control	1. byte: command mode 2nd byte: group select (See section following this table.)	2	A single global control job is sent to the DP slaves selected with group select. The command mode parameter specifies the following global control jobs: • SYNC • UNSYNC • FREEZE • UNFREEZE • CLEAR - is not supported (please read the information in the manual as well) It is possible to specify more than one job in the command mode parameter.

		Parameter in job field		
1	Trigger cyclic global control	1. byte: command mode 2nd byte: group select 3rd byte: autoclear (See section following this table.)	3	The sending of cyclic global control jobs to the DP slaves selected with group select is triggered on the PROFIBUS CP. The autoclear parameter is only evaluated with the SYNC global control job. If at least one DP slave in the selected group is not in the data transfer phase and autoclear=1 is set, the CLEAR mode is activated, in other words, the output data of the DP slaves is set to "0". The following global jobs can be activated in the command mode parameter: • SYNC • FREEZE • CLEAR (CLEAR-Bit = 1) - is not supported (please read the information in the manual as well) or deactivated: • UNSYNC • UNFREEZE • UNCLEAR (CLEAR bit = 0) It is possible to specify more than one job in the command mode parameter. An active cyclic global control job can only be terminated by a further global control job (cyclic or acyclic). To terminate the job set in the command mode, the job must be canceled. For example, the SYNC job is canceled by an UNSYNC job.
3	Delete older DP single diagnostic data	1. Byte: Slave address 1 to 126 127 = all slaves	1	The older diagnostic data stored on the PROFIBUS CP is deleted for one or all DP slaves.

5	Set DP mode for CPU STOP	1. Byte:
		RUN = 00_H
		CLEAR = 01_H
		STOP = 02_H
		OFFLINE = 03_H
			1	This job specifies which DP mode the PROFIBUS CP changes to if the CPU changes to STOP:
				• RUN
				• CLEAR
				• STOP (now the OFFLINE mode)
				• OFFLINE
				As default, the PROFIBUS CP changes to the DP status CLEAR if the CPU changes to STOP.
				This mode remains set during a CP mode change from RUN --> STOP --> RUN.
				Please read the information in the manual as well.

		Parameter in job field		

6	Set DP mode for CP STOP	1. Byte:
		STOP =02_H
		OFFLINE=03_H
			1	This job specifies which DP mode the PROFIBUS CP changes to if the CP changes to STOP.
				• STOP (now the OFFLINE mode)
				• OFFLINE
				As default, the PROFIBUS CP changes to the DP status OFFLINE if the CP changes to STOP.
				This mode remains set during a CP mode change from RUN --> STOP --> RUN.
				Please read the information in the manual as well.

7	Read input data cyclically (DP master class 2)	1. byte: slave address 1 to 125	1	This job is not supported. Please read the information in the manual as well.
8	Read output data cyclically (DP master class 2)	1. byte: slave address 1 to 125	1	This job is not supported. Please read the information in the manual as well.
9	Terminate cyclic processing of the DP slave by the DP master (class 1, class 2)	1. byte: slave address 1 to 125	1	This job terminates the cyclic reading of the input data or output data of the addressed DP slave or the data transfer (DP master class 1). The DP slave is then no longer processed by the PROFIBUS CP acting as DP master (class 2). **This deactivates the DP slave.**
10	Start cyclic processing as DP master (class 1)	1. byte: slave address 1 to 125	1	The PROFIBUS CP acting as the DP master (class 1) then assigns parameters to the addressed DP slave and starts cyclic data transfer (writing outputs/reading inputs). **This activates the DP slave.**

4.2.5.5 Command mode and group select - DP_CTRL

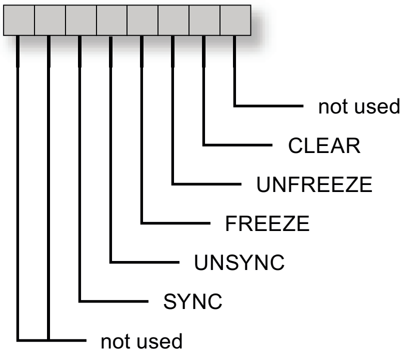
Structure of command mode

In the command mode parameter, you specify the modes for input and output data for the cyclic and acyclic global control jobs.

The meaning is as follows:

1 = activated
0 = not activated

Bit number: 7 6 5 4 3 2 1 0



Structure of group select

In the group select parameter, you specify the group to be addressed by the control job specified in the command mode parameter. The group select parameter occupies the second byte in the control job. Each bit defines a possible DP slave group.

The meaning is as follows:

1 = assigned
0 = not assigned

Bit number:	7	6	5	4	3	2	1	0
Group;	8	7	6	5	4	3	2	1

4.2.5.6 Condition codes of the DP_CTRL block

Condition codes

The following table shows the return codes formed by the DONE, ERROR and STATUS parameters that must be evaluated by the user program.

Note

For entries coded with 8FxxH in STATUS, refer to the information in the STEP 7 Standard and System Functions reference manual. The chapter describing error evaluation with the RET_VAL output parameter contains detailed information.

To find out which SFCs are used and are relevant for error evaluation, display the properties dialog of the FC described here in the "Calls" tab.

Table 4-10 DP_CTRL condition codes

DONE	ERROR	STATUS	Possible with CTYPE	Meaning
0	0	8181$_H$	0..10	Job active. DP master not started due to: • CP STOP or • "no parameter assignment" (occurs here instead of the code 0,1,8183$_H$).
1	0	0000$_H$	0..10	**Job completed without error.**
1	0	8214$_H$	0,1	**Job completed without error.** Message: Cyclic global control job is sent as acyclic global control job
1	0	8215$_H$	0,1	**Job completed without error.** The slaves addressed in the selected group are all deactivated.
1	0	8219$_H$	0,1	**Job completed without error.** An attempt was made to send an already active cyclic global control again. The global control continues unchanged.
1	0	8228$_H$	0,1	**Job completed without error.** Message: The DP slaves addressed in the selected groups do not have any input modules.
1	0	8229$_H$	0,1	**Job completed without error.** Message: The DP slaves addressed in the selected groups do not have any output modules.
1	0	8231$_H$	4,5,6	**Job completed without error.** Message: The DP status is already "RUN"
1	0	8232$_H$	4,5,6	**Job completed without error.** Message: The DP status is already "CLEAR"
1	0	8233$_H$	4,5,6	**Job completed without error.** Message: The DP status is already "STOP"
1	0	8234$_H$	4,5,6	**Job completed without error.** Message: The DP status is already "OFFLINE"

DONE	ERROR	STATUS	Possible with CTYPE	Meaning
1	0	8235$_H$	4	**Job completed without error.** Message: The DP status is already "RUN" with activated AUTOCLEAR
1	0	8236$_H$	4	**Job completed without error.** Message: The DP status is already "RUN" with deactivated AUTOCLEAR
1	0	8241$_H$	7-10	**Job completed without error.** Message: The specified DP slave was not configured.
1	0	8243$_H$	7-10	**Job completed without error.** Message: The DP slave is already deactivated since the module list of the DP slave in the CP database only contains empty modules.
1	0	8245$_H$	7-10	**Job completed without error.** Message: The DP slave is already in the "read input data cyclically" mode
1	0	8246$_H$	7-10	**Job completed without error.** Message: The DP slave is already in the "read output data cyclically" mode
1	0	8248$_H$	7-10	**Job completed without error.** Message: The module list of the DP slave in the CP database contains input, output, or input/output modules.
1	0	8249$_H$	7-10	**Job completed without error.** Message: This slave is deactivated due to a change in the DP mode.
1	0	824A$_H$	7-10	**Job completed without error.** Message: The DP slave is already deactivated due to a DP_CTRL job in the CPU program
0	1	8090$_H$	0..10	No module with this address exists.
0	1	8091$_H$	0..10	Logical base address not at a double word boundary.
0	1	80B0$_H$	0..10	The module does not recognize the data record.
0	1	80B1$_H$	0..10	The specified data record length is incorrect.
0	1	80C0$_H$	0..10	The data record cannot be read.
0	1	80C1$_H$	0..10	The specified data record is currently being processed.
0	1	80C2$_H$	0..10	There are too many jobs pending.
0	1	80C3$_H$		Resources occupied (memory).
0	1	8183$_H$	0..10	The DP master is not configured.
0	1	8184$_H$		System error or bad parameter type.
0	1	8311$_H$	0..10	CTYPE parameter outside the range of values
0	1	8312$_H$	0..10	The length of the area in the CONTROL parameter is too short.
0	1	8313$_H$	3,7,8,9, 10	The slave address parameter is outside the range of values.
0	1	8315$_H$	0,1	All DP slaves of the group specified in the global control are deactivated (always occurs with an empty group).
0	1	8317$_H$	8	The length of the configured output data is greater than the configured receive area of the DP slave. Activating the slave mode "Read output data" is not possible.
0	1	8318$_H$	0,1,4,5,6	The parameter 1st byte of the job data field is outside the range of values. With GLOBAL CONTROL, CLEAR was used with SYNC or a GLOBAL CONTROL with CLEAR set was sent to group 0.
0	1	831A$_H$	0,1	At least one DP slave cannot handle FREEZE.
0	1	831B$_H$	0,1	At least one DP slave cannot handle SYNC.
0	1	8333$_H$	0,1	This job is not permitted in the DP "STOP" mode.

DONE	ERROR	STATUS	Possible with CTYPE	Meaning
0	1	8334$_H$	0, 1	This job is not permitted in the DP "OFFLINE" mode.
0	1	8335$_H$	0, 1	The PROFIBUS CP is in PROFIBUS status: "Station not in ring".
0	1	8339$_H$	0, 1	At least one DP slave in the selected group is not in the data transfer phase.
0	1	833C$_H$	1	Cyclic global control must not be used in the "PLC <-> CP free running" mode. This error does not occur on the CP 3425 because this mode is not possible with this CP (PBUS data records are always used for data transfer).
0	1	8341$_H$	7-10	The specified DP slave was not configured.
0	1	8183$_H$	0..10	DP master not configured.
0	1	8184$_H$	-	System error or bad parameter type.
0	1	8F22$_H$	0..10	Area length error reading a parameter. (e.g. DB too short).
0	1	8F23$_H$	0..10	Area length error writing a parameter.
0	1	8F24$_H$	0..10	Area error reading a parameter.
0	1	8F25$_H$	0..10	Area error writing a parameter.
0	1	8F28$_H$	0..10	Alignment error reading a parameter.
0	1	8F29$_H$	0..10	Alignment error writing a parameter.
0	1	8F30$_H$	0..10	The parameter is in the writeprotected first current data block.
0	1	8F31$_H$	0..10	The parameter is in the writeprotected second current data block.
0	1	8F32$_H$	0..10	Parameter contains a DB number that is too high.
0	1	8F33$_H$	0..10	DB number error.
0	1	8F3A$_H$	0..10	Area not loaded (DB).
0	1	8F42$_H$	0..10	Timeout reading a parameter from the I/O area.
0	1	8F43$_H$	0..10	Timeout writing the parameter to the I/O area.
0	1	8F44$_H$	0..10	Access to a parameter to be read during block execution is prevented.
0	1	8F45$_H$	0..10	Access to a parameter to be written during block execution is prevented.
0	1	8F7F$_H$	0..10	Internal error, e.g. illegal ANY reference.
0	1	80C4$_H$	0..10	Communication error (occurs temporarily, it is usually best to repeat the job in the user program).
0	1	80D2$_H$	0..10	Logical base address incorrect.

4.3 Configuration limits / resources required by the FCs and FBs (PROFIBUS)

Required resources

NOTICE
Please note the version information of the blocks. Blocks with other versions have different resource requirements.

Table 4-11 Information for FCs / FBs with an S7400

NAME	Version	FC/FB no.	Load memory bytes	Work memory bytes	MC7 bytes	Local data bytes
AG_SEND	1.1	FC5	732	576	540	20
AG_RECV	1.1	FC6	656	522	486	20
AG_LSEND	3.0	FC50	1044	846	810	52
AG_LRECV	3.0	FC60	1190	992	956	58

Table 4-12 Information for FCs / FBs with an S7300

NAME	Version	FC/FB no.	Load memory bytes	Work memory bytes	MC7 bytes	Local data bytes
DP_SEND	3.0	FC1	1066	886	850	42
DP_RECV	3.0	FC2	1144	950	914	46
DP_DIAG	3.0	FC3	1956	1638	1602	58
DP_CTRL	3.0	FC4	1532	1292	1256	52
AG_SEND	4.2	FC5	1976	1664	1628	50
AG_RECV	4.7	FC6	1440	1206	1170	40

FBs for PROFIBUS FMS 5

5.1 Overview of FBs and their use

Overview

The following function blocks are available for an S7 station involved in FMS communication.

The list shows the block numbers as they are when supplied. You can change these block numbers.

Function block		Can be used in the function of the PROFIBUS CP as:		Meaning / function
Type	Block number	FMS client	FMS server	
IDENTIFY	FB2	X	X	For querying device properties
READ	FB3	X	-	For reading data
REPORT	FB4	-	X	For transferring data unconfirmed
STATUS	FB5	X	X	For a status query
WRITE	FB6	X	-	For writing data

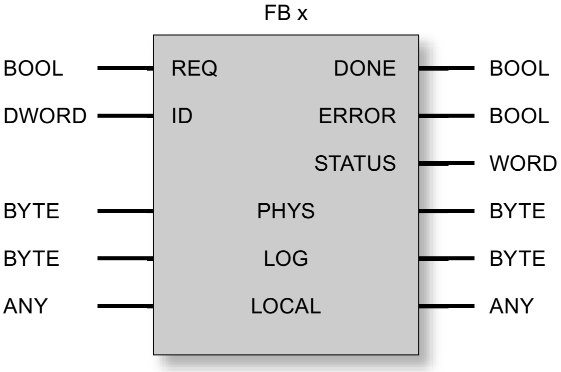
Difference between S7300 and S7400

Different FBs are supplied for the S7300 and S7400. Make sure you access the appropriate block library (SIMATIC_NET_CP) depending on whether you are creating a user program for an S7300 or an S7400.

5.2 FMS block parameters

FB call interfaces

The following sections describe the call interface for each FB as shown below:



Depending on the FB type, the FB has different parameters of the type INPUT, OUTPUT or INOUT.

The following tables explain the meaning, data type, range of values and memory area of **all** block parameters.

INPUT parameters

INPUT parameters	Meaning	Data type	Value range/memory area	Used in FB				
REQ	Edge signal for executing the block	BOOL	0=FALSE; 1=TRUE 0->1: "Start"/ I,Q,M,D,L	2	3	4	5	6
ID	This identifier identifies the FMS connection. S7300: The ID specifies both the LAN connection and the P bus address. S7400: The ID specifies both the LAN connection and the K bus connection. Make sure you use the ID from connection configuration or match it to this ID.	DWORD (with FB 1: WORD)	0001 0001 .. FFFF FFFF / I,Q,M,D,L	2	3	4	5	6

INPUT parameters	Meaning	Data type	Value range/memory area	Used in FB				
VAR_1	The parameter addresses the remote communications variable to be read or written. Depending on the configuration on the FMS server, a name or index can be specified.	ANY	String: Max. length = 254 bytes e.g. '<102>' (index access) "SLAVE2" (named access) D	2	3	4	-	6
SD_1	Address of a local data area from which the variables will be transferred.	ANY	This type corresponds to a reference to a DB, I/O process image or bit memory area. Example: SD_1 := P#DB17.DBX0.0 BYTE 16 In this example, the first 16 bytes of DB17 are transferred. I,Q,M,D,L,C,T,DBx	-	-	4	-	6
RD_1	Address of a local data area to which the variables will be transferred.	ANY	This type corresponds to a reference to a DB, I/O process image or bit memory area. Example: SD_1 := P#DB17.DBX0.0 BYTE 16 In this example, the first 16 bytes of DB17 are transferred. I,Q,M,D,L,DBx Note on array of bytes for S7-300: If there is an odd number of bytes to be read, the length of the receive area must be configured up to the next higher even number of bytes. Example: For an array[1..13] of bytes, reserve a receive buffer size of 14 bytes.	-	3	-	-	-

OUTPUT parameters

OUTPUT parameters	Meaning	Data type	Range of values/ memory area	Used in FB
DONE	Indicates that the job is completed.	BOOL	0=FALSE 1=TRUE: Job completed; I,Q,M,D,L	- - 4 - 6
NDR	Indicates reception of data.	BOOL	0=FALSE 1=TRUE: New data were accepted; I,Q,M,D,L	2 3 - 5 -
ERROR	Indicates whether or not an error occurred.	BOOL	0=FALSE 1=TRUE: Error occurred; I,Q,M,D,L	2 3 4 5 6
STATUS	Provides detailed information about warnings or errors after the job has been completed.	WORD	You will find detailed decoding information in the sections following. I,Q,M,D,L	2 3 4 5 6

INPUT/OUTPUT parameters

INOUT parameters	Meaning	Data type	Range of values/ memory area	Used in FB
PHYS	Indicates the physical status of the partner device (VFD).	BYTE	0...3 I,Q,M,D,L	- - - 5 -
LOG	Indicates the logical status of the partner (VFD).	BYTE	0...3 I,Q,M,D,L	- - - 5 -
LOCAL	"local detail" parameter of the partner	ANY	This detail can be up to 16 bytes long. I,Q,M,D,L	- - - 5 -
VENDOR	Name of the device vendor.	STRING	Length<255 D	2 - - - -
MODEL	Name of the device model.	STRING	Length<255 D	2 - - - -
REVISION	Version (revision) of the device.	STRING	Length<255 D	2 - - - -

Memory area

The abbreviated forms for the memory areas in the table correspond to the following:

Short form	Type
I	Input
Q	Output

Short form	Type
M	Bit memory
L	Temporary local data
D	Data block area
C	Counter
T	Timer
DBX	Data block

FB output parameters during the CP startup (S7400)

When the FB is called (REQ:0->1, EN_R=1) while the PROFIBUS CP is starting up (for example due to a power cycle or activating a switch) the following output parameters are possible:

- DONE = 0
- NDR = 0
- ERROR = 1
- STATUS = 0001 (connection has not been established yet) or
  STATUS = 0601 (Get-OV still running)

5.3 FB2 IDENTIFY

5.3.1 Meaning and call - IDENTIFY

Meaning of the block

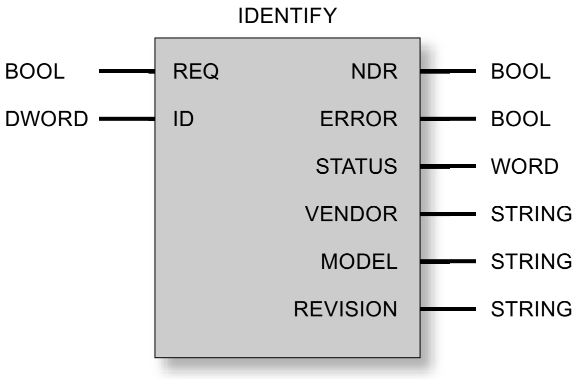
With the IDENTIFY function block, you can fetch the following information about the partner device (with S7 stations about the CPU):

- Name of the device vendor.
- Name of the device model.
- Version (revision) of the device.

Depending on the information you receive, you could, for example:

- Set the local program function to match the performance and response of the partner
- Set communication parameters

Call interface



Example in STL representation

STL	Explanation
`call FB 2, DB 22 (`	`//IDENTIFY block call with instance DB`
`REQ := M 1.0,`	`//Edge signal for executing the FB`
`ID := DW#16#10001,`	`//Matched with configuration of the`
	`//FMS connection`
`NDR := M 1.1,`	`//Indicates when "new data accepted"`
`ERROR := M 1.2,`	`//Indicates error in execution`
`STATUS := MW 20,`	`//Detailed error decoding`
`VENDOR := "SLAVE2".VENDOR_ABBILD,`	`//Data area for vendor name`
`MODEL := "SLAVE2".MODEL_ABBILD,`	`//Data area for model`
`REVISION := "SLAVE2".REV_ABBILD );`	`//Data area for revision`

Additional information

"SLAVE2"
is the symbolic name of a data block. This name is defined in the corresponding symbols table.
VENDOR_IMAGE, MODEL_IMAGE and REVISION_IMAGE
are variables of the data type STRING. These are defined in the "SLAVE2" data block.

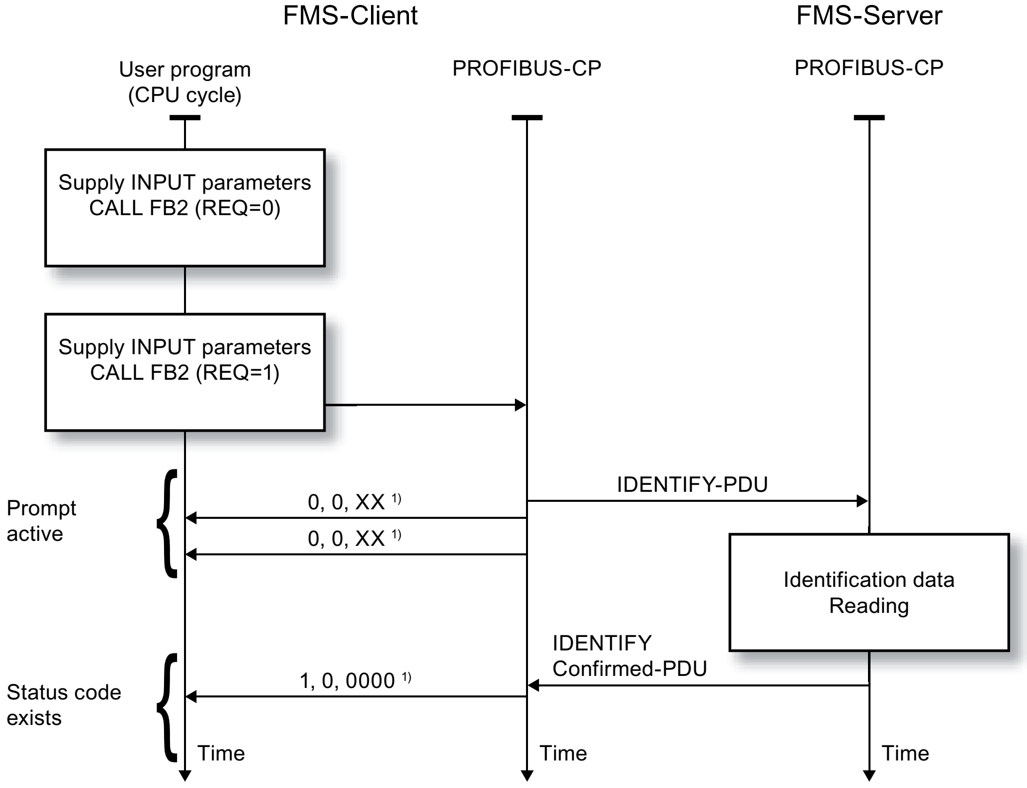
5.3.2 How IDENTIFY works

Operating principle

The following flow chart illustrates the normal sequence of an IDENTIFY job.

The job is activated by a (positive-going) edge change at the parameter REQ.

Each IDENTIFY job of the user program is acknowledged by the PROFIBUS CP with a value in the output parameters NDR, ERROR and STATUS.



1) Parameter transfer NDR, ERROR, STATUS

5.4 FB3 READ

5.4.1 Meaning and call - READ

Meaning

The READ function block reads data from a data area of the communication partner specified by a name or index depending on the assignment of parameters for the job. The data that is read is saved locally in a data block, an area in the process image of the inputs/outputs or in a bit memory area.

Requirement: Configuration of communications variables

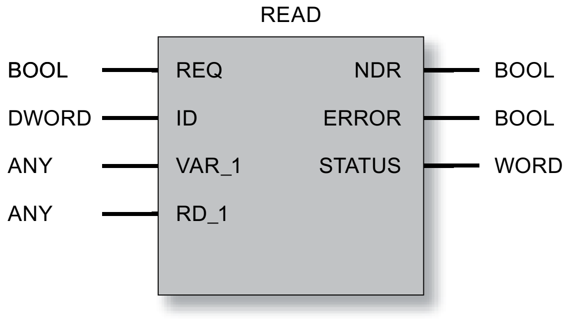
The structure of the variables on the communications partner (FMS server) is fixed. When the FMS connection is established, the structure description is read out from the communications partner. This is then available on the PROFIBUS CP to convert the data to the FMS representation.

The structure description is only read when the connection is established if the communications variable was selected during configuration of the FMS connection.

Access rights

Remember that access rights can be set for the data transfer. Data transmission is then only possible if the FMS client has been assigned suitable rights.

FB call interface



Example in STL representation

STL	**Explanation**
`call FB 3, DB 29 (`	`//READ block call with instance DB`
`REQ := M 1.0,`	`//Signal edge change to execute the FB`
`ID := DW#16#10001,`	`//Compared with configuration of`
	`//FMS connection`
`VAR_1 := "SLAVE2".INDEX,`	`//Addresses K variable that will be read`

STL	Explanation
`RD_1 := "PROZESS".Motor1,`	`//Addresses data area as destination`
`NDR := M 1.1,`	`//Confirmation of execution`
`ERROR := M 1.2,`	`//Indicates incorrect execution`
`STATUS := MW 20 );`	`//Detailed error decoding`

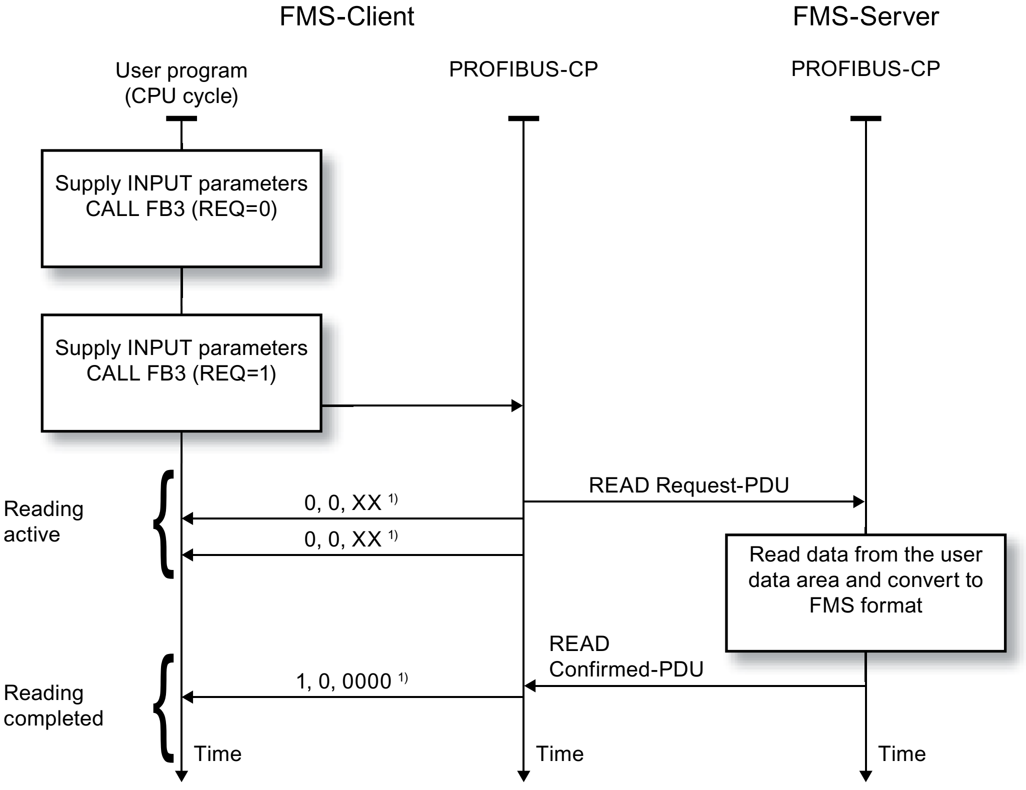
5.4.2 How READ works

Operating principle

The following sequence chart shows the normal sequence of data reception triggered with READ in the user program.

The job is activated by a (positive-going) edge change at the parameter REQ.

Every READ job in the user program is acknowledged by the PROFIBUS CP with values in the output parameters NDR, ERROR and STATUS.



1) Parameter transfer NDR, ERROR, STATUS

Guarantee of data transfer

The diagram shows that the reading out of the data is confirmed with the code NDR=1, ERROR=0 and STATUS=0000.

Positive confirmation of the read job does not necessarily mean that the read job was registered by the partner application.

5.5 FB4 REPORT

5.5.1 Meaning and call - REPORT

Meaning of the block

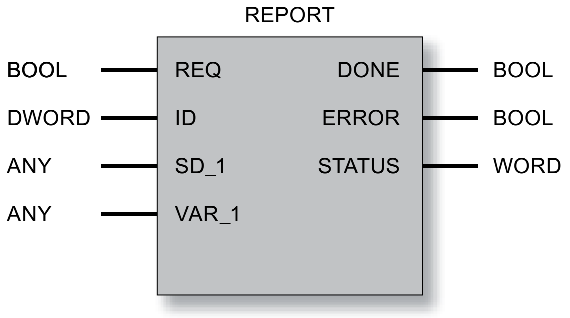
The REPORT function block allows unconfirmed transmission of variables by an FMS server. This job type is used particularly for transmission on broadcast/multicast FMS connections.

The structure of the variables to be reported must be configured locally on the FMS server.

S7 station as communications partner

To allow the reported variables to be accepted by the communications partner, the variables must be entered during configuration of the communications partner (FMS client).

Call interface



Example in STL representation

STL	Explanation
`call FB 4, DB 28 (`	`//REPORT block call with instance DB`
`REQ := M 1.0,`	`//Signal edge change to execute the FB`
`ID := DW#16#10001,`	`//Compared with configuration of FMS connection`
`VAR_1 := "SLAVE2".INDEX,`	`//Names the C variable to be reported`
`SD_1 := "PROZESS".Motor1,`	`//Addresses the source data area`
`DONE := M 1.1,`	`//Confirmation of execution`
`ERROR := M 1.2,`	`//Indicates incorrect execution`
`STATUS := MW 20 );`	`//Detailed error decoding`

NOTICE

The parameter SD_1 is used to address the data area from which the variable values are read and reported. According to the FMS conventions, you also need to specify the variable index on the FC interface. The consistency of this information is not, however, checked when the call is executed.

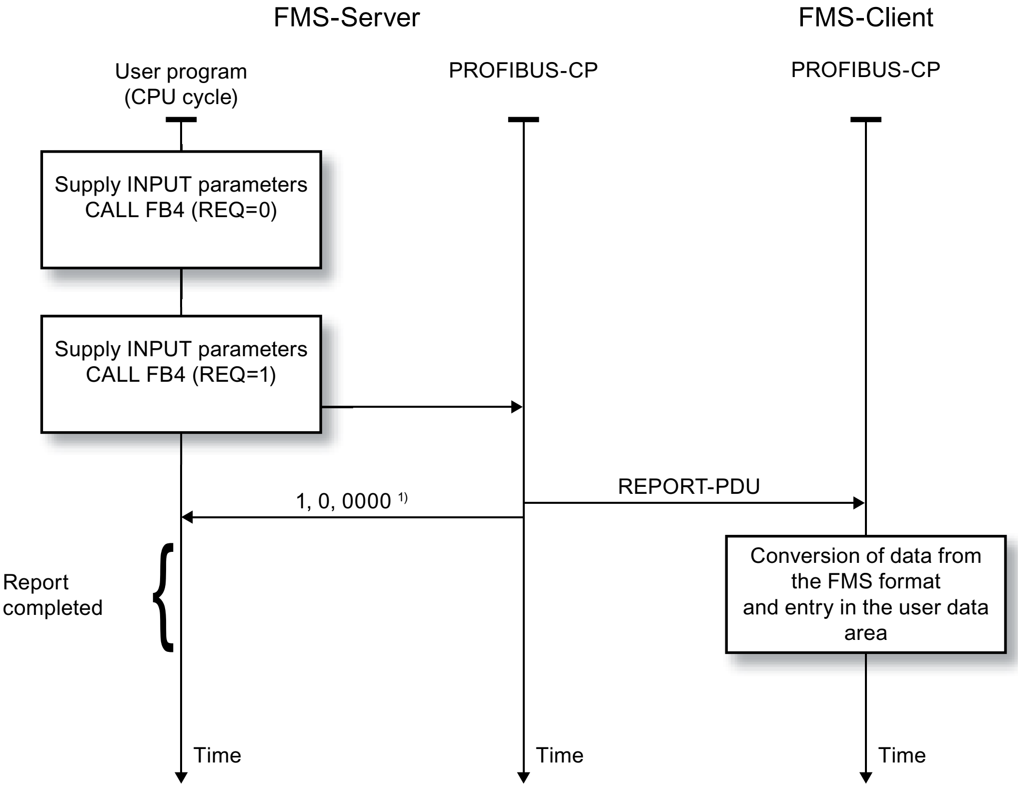
5.5.2 How REPORT works

Operating principle

The following flow chart shows the normal sequence of a data transfer transferred by REPORT in the user program.

The job is activated by a (positive-going) edge change at the parameter REQ.

Every REPORT job in the user program is acknowledged by the PROFIBUS CP with values in the output parameters DONE, ERROR and STATUS.



1) Parameter transfer DONE, ERROR, STATUS

5.6 FB5 STATUS

5.6.1 Meaning and call - STATUS

Meaning of the block

The STATUS function block allows status information to be requested from the communications partner on the specified FMS connection.

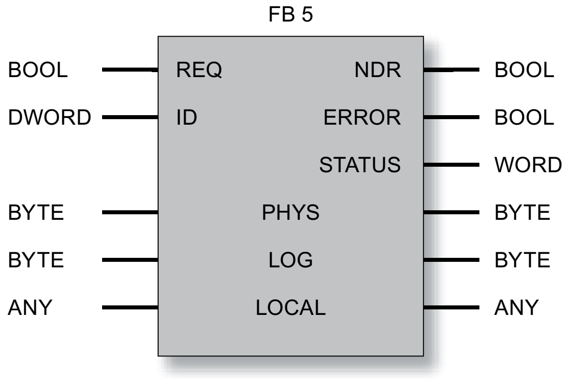
The following information is available:

- The logical status of the VFD;
  for example information whether communication is possible.
- The physical status of the VFD
  information about the status of the device
- Device-specific information
  normally provides vendorspecific information

The following table provides information about the codes that a device can supply as a result of the status request:

Device	Message version	Log	Phys	Local detail
S7 with PROFIBUS CP	1	00H: Ready for communication CP in RUN, CPU in RUN	10H: Ready for communication, CPU in RUN	No entry
	2	02H: Limited services, CP in RUN, CPU in STOP	13H: Maintenance required, CPU in STOP	No entry
Third-party device	The following are possible:	00H: Ready for communication 02H: Number of services limited	10H: Operational 11H Partly operational 12H Not operational 13H Maintenance required	- vendor specific -

Call interface



Example in STL representation

STL	Explanation
`call FB 5, DB 21 (`	`//STATUS block call with instance DB`
`REQ := M 1.0,`	`//Signal edge change to execute the FB`
`ID := DW#16#10001,`	`//Compared with configuration of FMS connection`
`NDR := M 1.1,`	`//Indicates when new data is accepted`
`ERROR := M 1.2,`	`//Indicates incorrect execution`
`STATUS := MW 20,`	`//Detailed error decoding`
`PHYS := MB 22,`	`//Data area for physical status`
`LOG := MB 23,`	`//Data area for logical status`
`LOCAL := P#DB18.DBX0.0 WORD8 );`	`//Data area for "local detail"`

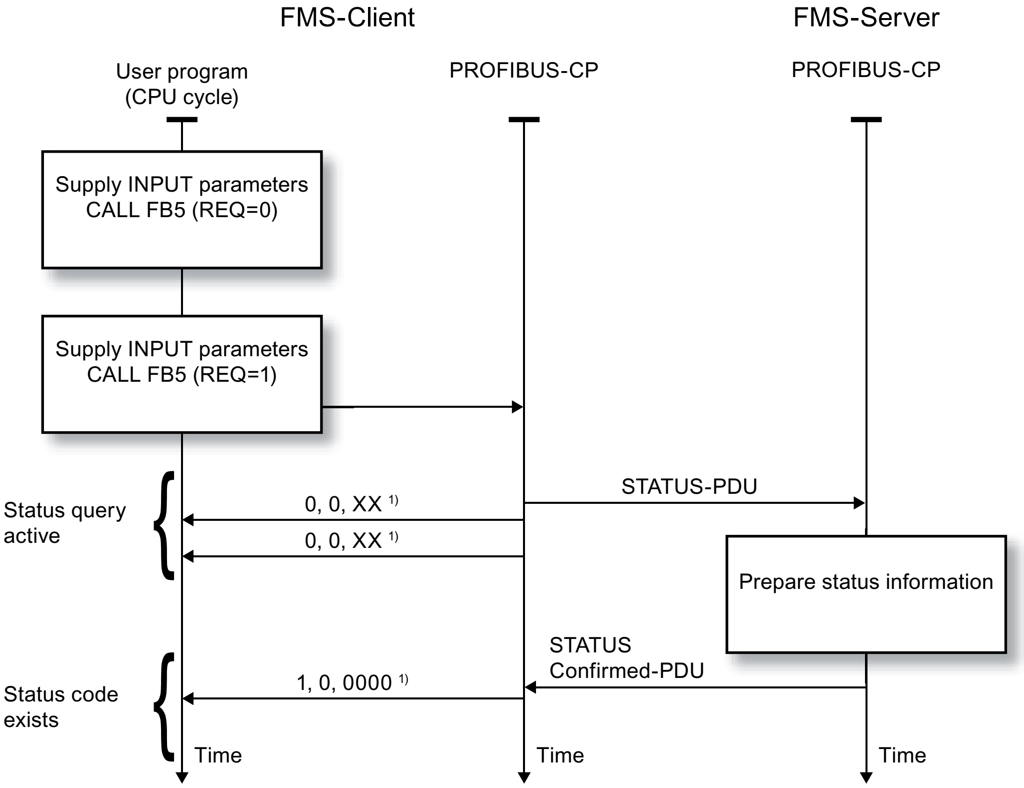
5.6.2 How STATUS works

Operating principle

The following flow chart shows the normal sequence of a STATUS job.

The job is activated by a (positive-going) edge change at the parameter REQ.

Every STATUS job in the user program is confirmed by the PROFIBUS CP with values in the output parameters NDR, ERROR and STATUS.



1) Parameter transfer NDR, ERROR, STATUS

5.7 FB6 WRITE

5.7.1 Meaning and call - WRITE

Meaning

The WRITE FB transfers data from a specified local data area to a data area on the communication partner. The local data area can be a data block, an area in the process input or output image or a bit memory area. (See also parameter SD_1, FMS block parameter (Page 168))

The data area of the communication partner is specified using a variable name or a variable index.

Requirement: Configuration of communications variables

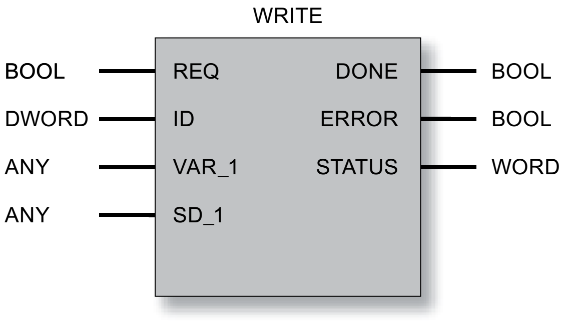
The structure of the variables on the communications partner (FMS server) is fixed. When the FMS connection is established, the structure description is read out from the communications partner. This is then available on the PROFIBUS CP to convert the data to the FMS representation.

The structure description is only read when the connection is established if the communications variable was selected during configuration of the FMS connection.

Access rights

Remember that access rights can be set for the data transfer. Data transmission is then only possible if the FMS client has been assigned suitable rights.

Call interface



Example in STL representation

STL	Explanation
call FB 6, DB 28 (	//WRITE block call with instance DB
REQ := M 1.0,	//Signal edge change to execute the FB

```
STL                                   Explanation
ID := DW#16#10001,                    //Compared with configuration of FMS connection
VAR_1 := "SLAVE2".INDEX,              //Names the C variable to be written
SD_1 := "PROZESS".Motor1,             //Addresses the source data area
DONE := M 1.1,                        //Confirmation of execution
ERROR := M 1.2,                       //Indicates incorrect execution
STATUS := MW 20 );                    //Detailed error decoding
```

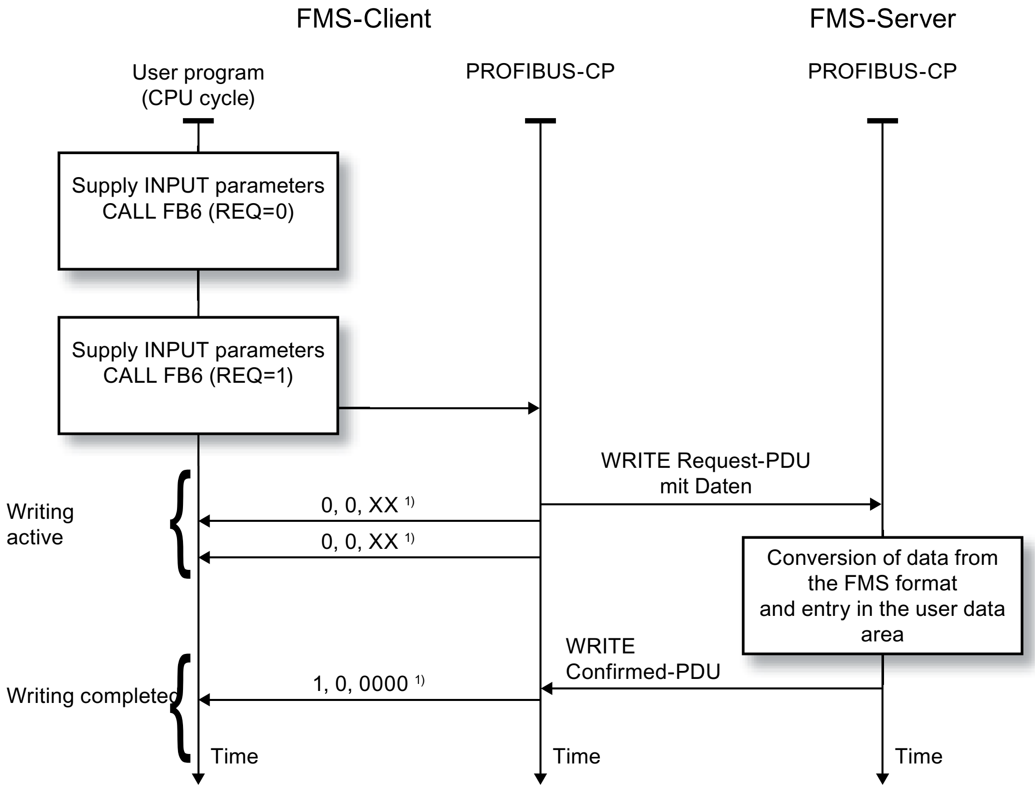
5.7.2 How WRITE works

Operating principle

The following flow chart shows the normal sequence of a data transfer triggered with WRITE in the user program.

The job is activated by a (positive-going) edge change at the parameter REQ.

Every WRITE job in the user program is confirmed by the PROFIBUS CP with values in the output parameters DONE, ERROR and STATUS.


FMS-Client
FMS-Server
User program (CPU cycle)
PROFIBUS-CP
PROFIBUS-CP
Supply INPUT parameters CALL FB6 (REQ=0)
Supply INPUT parameters CALL FB6 (REQ=1)
WRITE Request-PDU mit Daten
Writing active
0, 0, XX 1)
0, 0, XX 1)
Conversion of data from the FMS format and entry in the user data area
WRITE Confirmed-PDU
Writing completed
1, 0, 0000 1)
Time
Time
Time

1) Parameter transfer DONE, ERROR, STATUS

Guarantee of data transfer

The diagram also shows that with the confirmation DONE=1, ERROR=0 and STATUS=0000, data transfer to the communications partner and entry in the remote data area is functioning correctly.

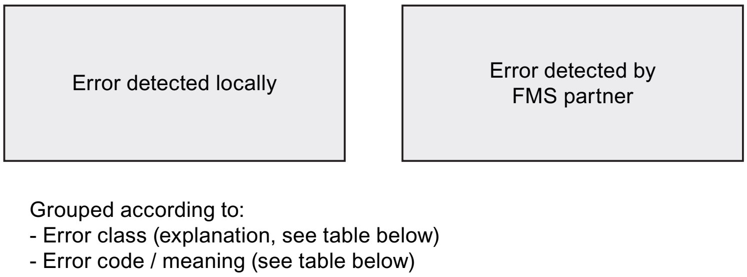
A positive confirmation of the job does not necessarily mean that the data has already been received and processed by the partner application.

5.8 Condition codes and error messages - FMS blocks

Structure of the tables

The following tables explain the condition codes and error codes that must be handled in your user program. The meanings of the parameters DONE/NDR, ERROR and STATUS are explained in FMS block parameters (Page 168)

To provide a better overview, the error codes are listed as follows:



Grouped according to:
- Error class (explanation, see table below)
- Error code / meaning (see table below)

Error-free job execution

If the job was executed free of errors, the parameters on the FB interface have the following values:

DONE/NDR	ERROR	STATUS	Meaning
1	0	0x0000	Job completed without errors
0	0	0x000B	Job active

Error classes

The possible error codes are grouped into the following error classes:

Table 5-1 "Block" error class

Error class	Meaning
Block	Indicates errors or problems involving the following: • FB parameter assignment • Block execution in the CPU and CP
Application	Indicates errors or problems on the interface between the user program and FB.
Definition	Indicates errors that usually involve inconsistencies between the user program and FMS configuration.
Components	Indicates resource problems on the PROFIBUS CP.
Service	Indicates errors or problems in conjunction with the requested FMS service.
Access	Indicates denied access to objects due to the following: • Absence of access rights • Hardware problems • Other inconsistencies

Error class	Meaning
OD (object directory)	Indicates problems accessing the object dictionary of the VFD.
VFD status	Unspecified error on the VFD
otherwise	Other errors

5.8.1 Error detected locally

Table 5-2 "Application" error class

DONE/NDR	ERROR	STATUS	Meaning
0	1	0x0001	Communications problem: For example, communications bus connection not established.
0	1	0x0002	The function cannot be executed: Either negative acknowledgment from CP or error in the sequence, for example communications bus error.
0	1	0x0003	The connection is not configured (invalid ID specified). If the connection is configured, the error message indicates that the permitted parallel job processing limit has been exceeded. Example: SAC=0 is configured and a REPORT job is sent.
0	1	0x0004	The receive data area is too short or the data types do not match.
0	1	0x0005	A reset request has been received from the CP (BRCV).
0	1	0x0006	The corresponding job execution on the CP is in the DISABLED state or a reset request has been received from the CP; the transfer is therefore incomplete.
0	1	0x0007	Corresponding job execution on the CP is in the wrong state. For REPORT: The error is specified in greater detail in the diagnostic buffer.
0	1	0x0008	Job execution on the CP signals an error accessing the user memory.
0	1	0x000A	Access to local user memory not possible (for example, the DB was deleted).
0	1	0x000C	When the underlying BSEND or BRCV SFBs were called, an instance DB that does not belong to SFB12/SFB13 was specified or no instance DB was used, but rather a global DB.
0	1	0x0014	Not enough work or load memory available.

DONE/NDR	ERROR	STATUS	Meaning
0	1	0x0200	Unspecified application reference error.
0	1	0x0201	The configured connection cannot be established at present; for example LAN connection not established.

Table 5-3 "Definition" error class

DONE/NDR	ERROR	STATUS	Meaning
0	1	0x0300	Unspecified definition error.
0	1	0x0301	Object with requested index/name is not defined.
0	1	0x0302	Object attributes are inconsistent.
0	1	0x0303	Name exists already.

Table 5-4 "Resources" error class

DONE/NDR	ERROR	STATUS	Meaning
0	1	0x0400	Unspecified resource error.
0	1	0x0401	No memory available.

Table 5-5 "Service" error class

DONE/NDR	ERROR	STATUS	Meaning
0	1	0x0500	Unspecified service error.
0	1	0x0501	Conflict due to object status.
0	1	0x0502	Configured PDU size exceeded.
0	1	0x0503	Conflict due to object restrictions.
0	1	0x0504	Inconsistent parameters.
0	1	0x0505	Illegal parameters.

Table 5-6 "Access" error class

DONE/NDR	ERROR	STATUS	Meaning
0	1	0x0600	Unspecified access error.
0	1	0x0601	Invalid object or no OD loaded;
0	1	0x0602	Hardware fault
0	1	0x0603	Object access was denied.
0	1	0x0604	Invalid address.
0	1	0x0605	Inconsistent object attributes.
0	1	0x0606	Object access not supported.

DONE/NDR	ERROR	STATUS	Meaning
0	1	0x0607	Object does not exist in OD or GetOD still active.
0	1	0x0608	Type conflict or variable content outside permitted range of values
0	1	0x0609	Access using names not supported.

Table 5-7 "Object dictionary" (OD) error class / VFD Status/Reject error class

DONE/NDR	ERROR	STATUS	Meaning
0	1	0x0700	Unspecified OD error.
0	1	0x0701	Permitted name length exceeded.
0	1	0x0702	Overflow of the object dictionary.
0	1	0x0703	Object dictionary is write protected.
0	1	0x0704	Overflow of the extension length.
0	1	0x0705	Overflow of the object description length.
0	1	0x0706	Processing problem.

DONE/NDR	ERROR	STATUS	Meaning
0	1	0x0100	Unspecified VFD status error.
0	1	0x0108	RCC/SAC/RAC error
0	1	0x0106	Service not supported.
0	1	0x0105	PDU length error
0	1	0x0102	Bad FMS-PDU

Table 5-8 "Other" error class

DONE/NDR	ERROR	STATUS	Meaning
0	1	0x0800	Unspecified error.

5.8.2 Errors detected by FMS partner

Table 5-9 Application error class

DONE/NDR	ERROR	STATUS	Meaning
0	1	0x8200	Unspecified application reference error.
0	1	0x8201	Application (e.g. user program) cannot be reached.

Table 5-10 Definition error class

DONE/NDR	ERROR	STATUS	Meaning
0	1	0x8300	Unspecified definition error.
0	1	0x8301	Object with requested index/name is not defined.
0	1	0x8302	Object attributes are inconsistent.
0	1	0x8303	Name exists already.

Table 5-11 Resources error class

DONE/NDR	ERROR	STATUS	Meaning
0	1	0x8400	Unspecified resource error.
0	1	0x8401	No memory available.

Table 5-12 Service error class

DONE/NDR	ERROR	STATUS	Meaning
0	1	0x8500	Unspecified service error.
0	1	0x8501	Conflict due to object status.
0	1	0x8502	Configured PDU size exceeded.
0	1	0x8503	Conflict due to object restrictions.
0	1	0x8504	Inconsistent parameters.
0	1	0x8505	Illegal parameters.

Table 5-13 Access error class

DONE/NDR	ERROR	STATUS	Meaning
0	1	0x8600	Unspecified access error.
0	1	0x8601	Invalid object.
0	1	0x8602	Hardware error.
0	1	0x8603	Object access was denied.
0	1	0x8604	Invalid address.
0	1	0x8605	Inconsistent object attributes.
0	1	0x8606	Object access is not supported.
0	1	0x8607	Object does not exist.
0	1	0x8608	Type conflict or variable content outside permitted range of values
0	1	0x8609	Access using names is not supported.

Table 5-14 OD (object dictionary) error class

DONE/NDR	ERROR	STATUS	Meaning
0	1	0x8700	Unspecified OD error.
0	1	0x8701	Permitted name length exceeded.
0	1	0x8702	Overflow of the object dictionary.
0	1	0x8703	Object dictionary is write protected.
0	1	0x8704	Overflow of the extension length.
0	1	0x8705	Overflow of the object description length.
0	1	0x8706	Processing problem.

Table 5-15 VFD status error class / "Other" error class

DONE/NDR	ERROR	STATUS	Meaning
0	1	0x8100	Unspecified VFD status error.

DONE/NDR	ERROR	STATUS	Meaning
0	1	0x8000	Unspecified error detected by partner.

5.9 Quantity framework / resource requirements of FBs (PROFIBUS FMS)

Note

Please note the version information of the blocks. Blocks with other versions have different resource requirements.

Table 5-16 Information for FBs with S7-400

NAME	Version	FB No.	Load memory bytes	Work memory bytes	MC7 Bytes	Local data Bytes	Instance DB Block Bytes	Instance DB MC7 Bytes
IDENT	1.3	2	1658	1364	1328	136	464	196
READ	1.5	3	2474	2086	2050	130	606	338
REPORT	1.5	4	2184	1818	1782	156	588	332
STATUS	1.3	5	1656	1390	1354	112	438	190
WRITE	1.5	6	2486	2094	2058	142	632	358

Table 5-17 Information for FBs with S7-300

NAME	Version	FB No.	Block bytes	Work memory bytes	MC7 Bytes	Local data Bytes	Instance DB Block Bytes	Instance DB MC7 Bytes
IDENT	1.6	2	1462	1254	1218	86	306	158
READ	1.5	3	1998	1700	1664	64	218	70
REPORT	1.6	4	2024	1718	1682	76	230	72
STATUS	1.6	5	1430	1244	1208	60	182	46
WRITE	1.6	6	2016	1710	1674	76	230	72

-