

Block Description FB283 SINAMICS <--> S7-CPU

Description of Functions

Edition 07/2011

Safety notices

This manual contains information that you should observe to ensure your own personal safety as well as to avoid material damage. Safety information is highlighted by a safety alert symbol and graded as follows according to the degree of risk involved:



DANGER

indicates that death, serious injury, or substantial property damage **will** result if proper precautions are not taken.



WARNING

indicates that death, serious injury, or substantial property damage **may** result if proper precautions are not taken.



CAUTION

indicates that minor personal injury or property damage may result if proper precautions are not taken.

NOTE

is an important information about the product, the way to handle the product or the respective part of the documentation and we wish to especially bring this to your notice.

Qualified personnel

Commissioning and operation of a device may only be performed by **qualified personnel**. In the context of the safety information related to this manual, the term qualified personnel refers to persons who are authorized to commission, ground and label devices, systems and circuits in accordance with the standards of safety engineering.

Intended use

Please observe the following:



WARNING

This equipment is only permitted to be used for the applications described in the catalog and in the technical description, and only in conjunction with non-Siemens equipment and components recommended by Siemens.

Correct transport, storage, installation and assembly, as well as careful operation and maintenance, are required to ensure that the product operates safely and without faults.

Trademarks

SIMATIC®, SIMATIC HMI®, SIMATIC NET®, SIROTEC®, SINUMERIK® and USS® are registered trademarks of Siemens AG. Other names in this publication might be trademarks whose use by a third party for its own purposes may violate the rights of the registered holder.

Copyright © Siemens AG 2000 - 2008 All rights reserved

Distribution or reproduction of this documentation, use and distribution of its contents are prohibited unless specifically approved. Offenders will be liable for damages. All rights reserved, especially for granting patents or for GM registration.

Disclaimer of liability

We have checked that the contents of this publication agree with the hardware and software described here. However, since deviations cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this document is regularly checked and the necessary corrections included in subsequent editions. We welcome any suggestions for improvement.

Siemens AG
Automation and Drives Group (A&D)
Motion Control Systems Division (MC)
Postfach 3180, D-91050 Erlangen, Germany

We reserve the right to make technical changes.

Siemens AG

SINAMICS

Description

Function block FB 283 SINAMICS ↔ S7 CPU

Valid for
Function block FB 283, as of Version 2.1

General	1
Structure of the user interface	2
Function block call	3
User interface assignment	4
Traversing task processing	5
Configuration	6
Error handling	7
Testing aids	8

Table of contents

1.	General	1
2.	Structure of the user interface	3
3.	Function block call	5
3.1.	General	5
3.2.	Parameterization input „AXIS_NO“	8
3.2.1.	SINAMICS G120	8
3.2.2.	SINAMICS G130 / G150	8
3.2.3.	SINAMICS S.....	8
3.3.	Configuration guide	9
4.	User interface assignment	12
4.1.	List (axis DB).....	12
4.2.	Description	14
5.	Traversing task processing	15
6.	Configuration.....	18
6.1.	Standard message frame 1.....	18
6.2.	Siemens message frame 111 – block positioning and MDI.....	20
7.	Error during task processing	22
8.	Testing aids.....	24
8.1.	Standard message frame 1.....	24
8.2.	SIEMENS message frame 111	24

1. General

This block supports the data exchange between a SINAMICS and an S7 CPU via PROFIBUS and PROFINET. As of the present version V2.1, this block supports devices of the SINAMICS S and G family. The block can be used in every S7-300/400 CPU that supports at least DP-V1. Thanks to preconfigured UDTs, the data interface is defined individually for each drive in a data block.

Fig. 1 shows the basic communication of the SINAMICS with a PROFIBUS master. The process data (PZD) is transferred consistently over the entire length of the PZD area. The control and checkback signals are defined in more detail in the User interface assignment. The task data (parameters, indices and values) is transferred with the aid of acyclic services, e.g. PROFIBUS DP-V1. The process is the same as for the data exchange with a PROFINET IO controller.

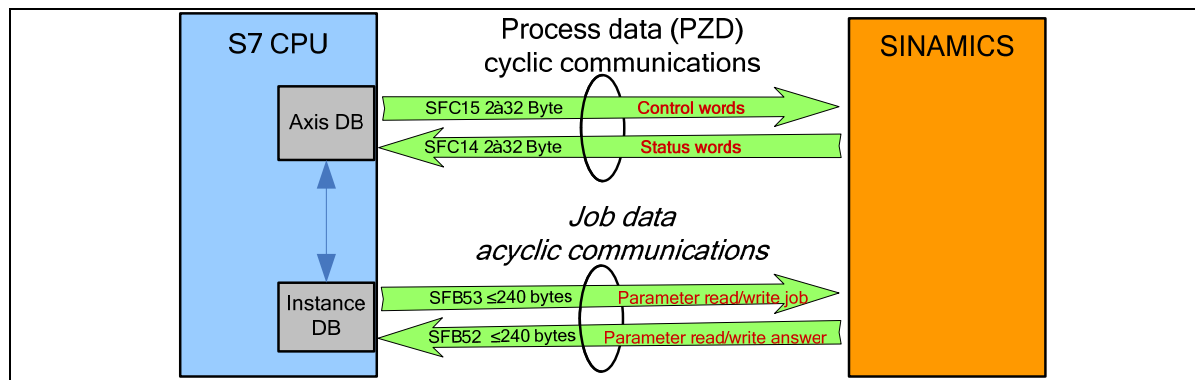


Fig. 1 Example: Consistent data transfer

The block can be inserted alternatively in the following OBs:

Cyclic task: OB1

Watchdog interrupt OB: E.g. OB32

NOTE

A language change for the block can be performed to the following languages in the SIMATIC Manager:

- German (Germany)
- Spanish (International Sort)
- Italian (Italy)
- French (France)
- English (United States)

Task data interface

Fig. 2 shows the communication sequence of the acyclic parameter interface (see also PROFIdrive Profiles, Drive Technology → PROFIBUS user organization, order number 3.172). The tasks are conveyed in FB283 with the aid of the system function blocks SFB52 and 53.

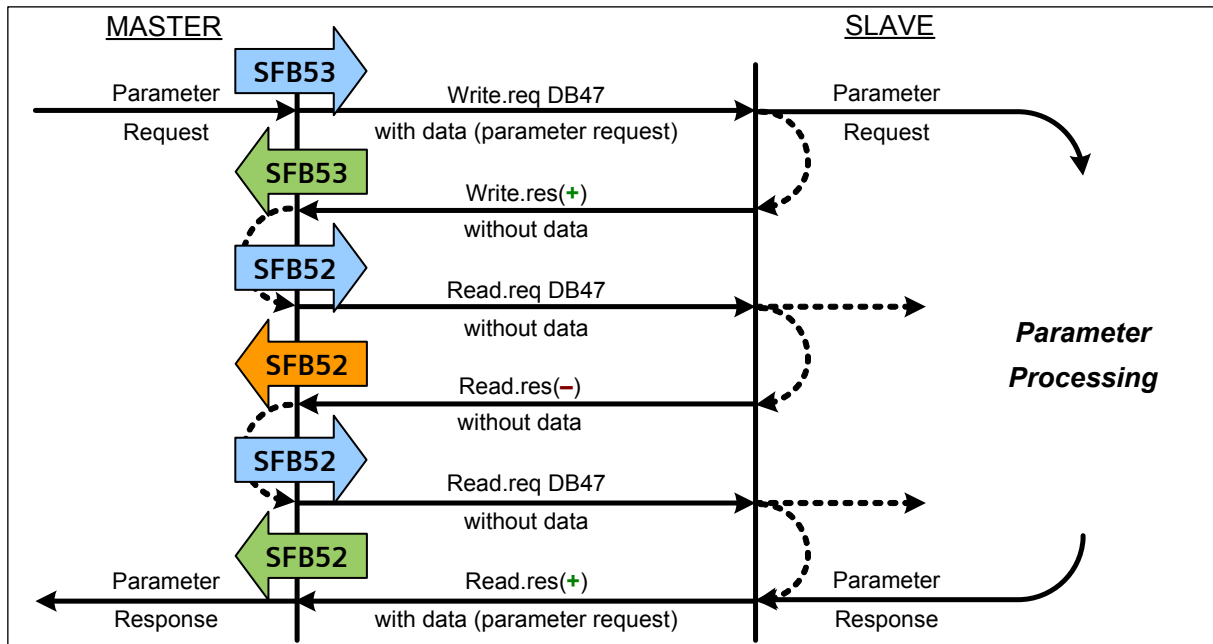


Fig. 2 Communication sequence of the parameter tasks via acyclic communication services

2. Structure of the user interface

Fig. 3 shows the basic structure of the user interface. An axis-specific data block with preconfigured UDTs must be generated for each axis. The process data area (**either** UDT positioning mode **or** UDT speed-controlled mode) via which the SINAMICS is controlled or which mirrors its checkback signals, is essential. Optional data, such as traversing blocks and fault messages, can be integrated individually in the axis-specific data block according to requirements. The FB283 must be called cyclically once for each axis.

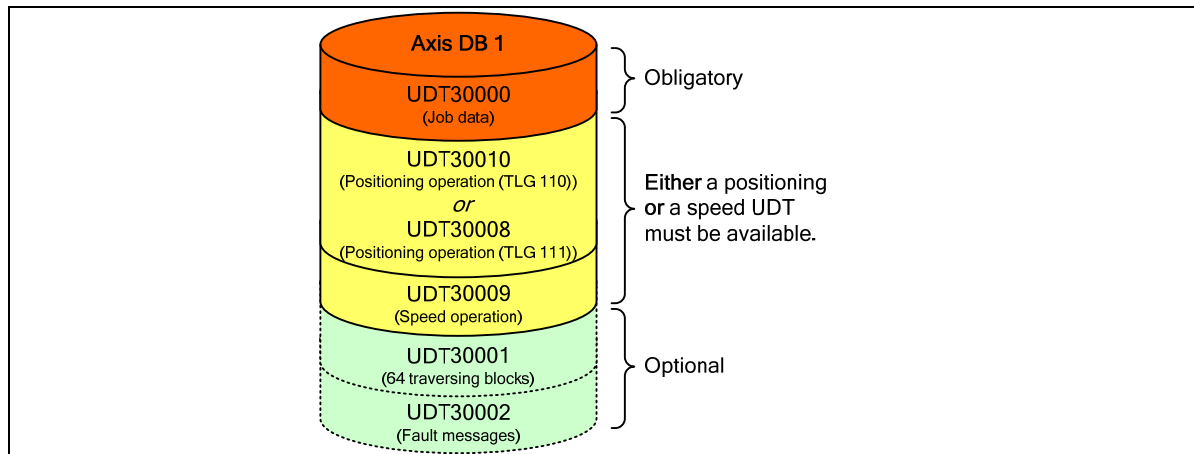


Fig. 3 Basic structure of the user interface

In the first S7 CPU cycle or after the "RESTART" input has been set, the FB283 checks which data areas have been defined by the user in the axis DB (Fig. 4). This is stored in the internal data of the FB283 (→ instance DB). If the structure of an axis DB is changed, the S7 CPU must be restarted (restart or cold restart) or the "RESTART" input set.

Notice

The FB283 block must be called during the first PLC cycle. Otherwise the initialization of the FB283 after the PLC restart is not performed and the structure of the axis DB is not identified and saved to the instance DB. The autoscan of the UDTs has been implemented with SFC6.

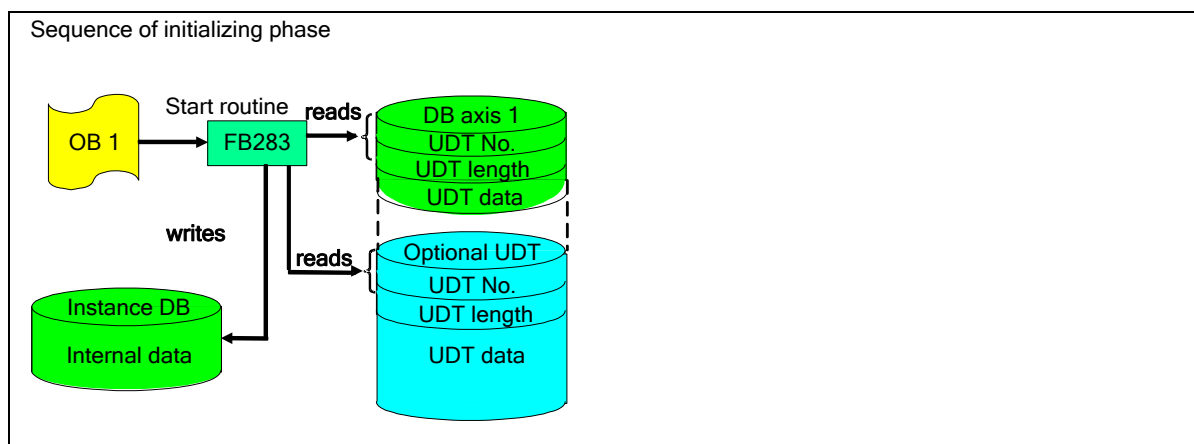


Fig. 4 Checking of the data areas by the FB283

The structure of the obligatory UDT is shown in Fig. 5.
The exact assignment of the UDT can be found in User interface assignment.

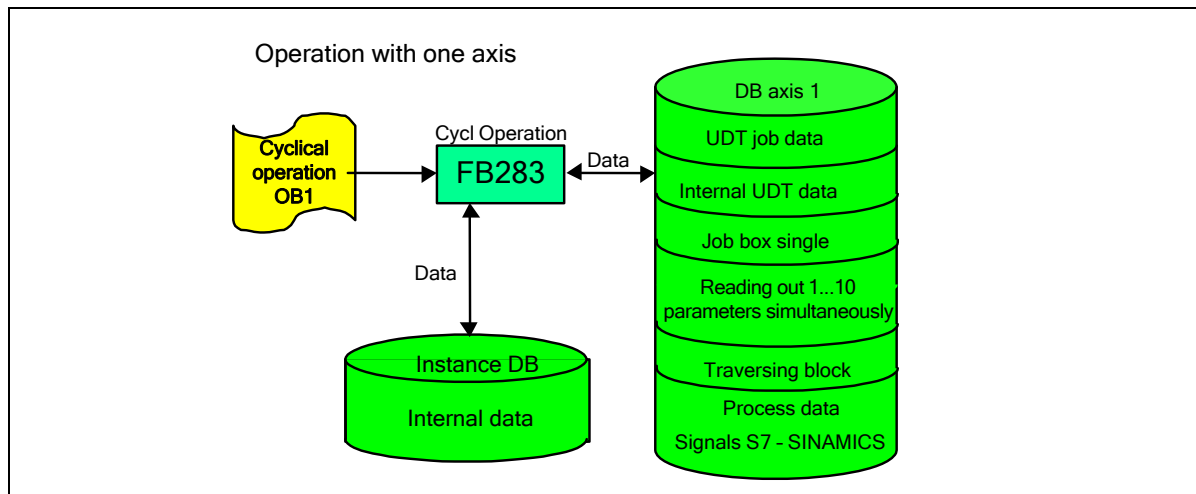


Fig. 5 Structure of the obligatory UDT

When operating with n axes, only $n+1$ DBs are required when the internal data is stored in a multi-instance DB, see Fig. 6.

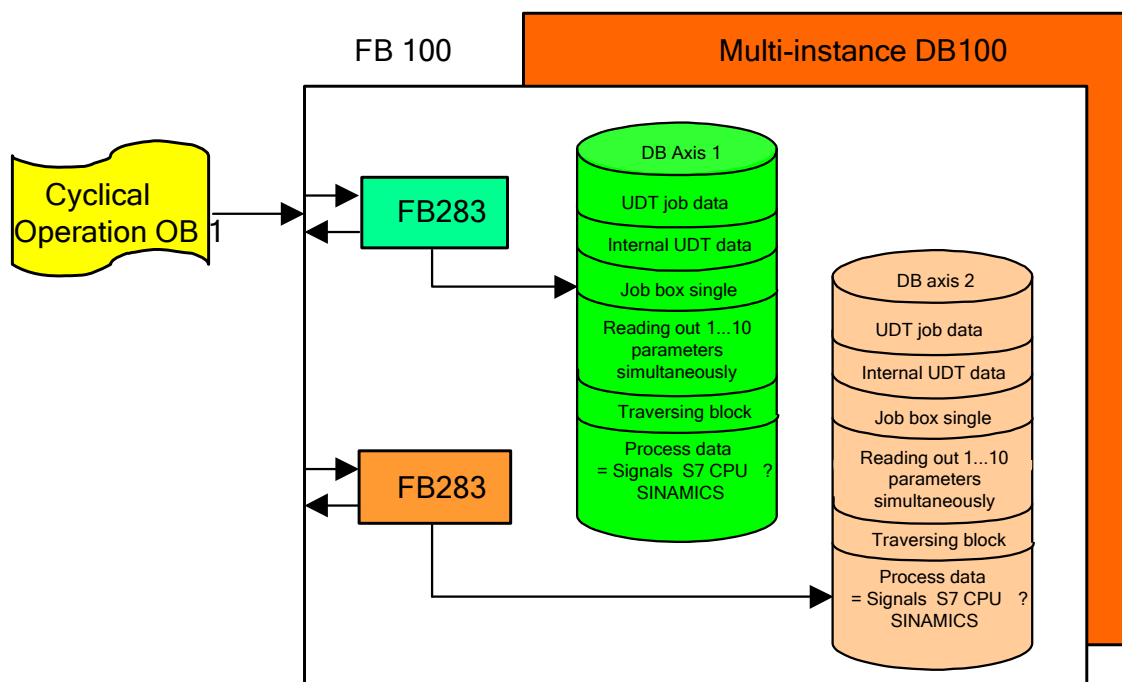


Fig. 6 Operation with n axes as multi-instance solution

NOTICE

When using multi-instances, make sure that the multi-instance for the FB283 does not exceed the address 4095 in the instance DB. If the limit is exceeded, the following error is entered in the diagnostics buffer of the S7 CPU: Area length error when writing, and the OB "Programming error OB (OB121)" is called.

3. Function block call

3.1. General

The function block FB283 must be called cyclically for each axis. Communication is performed via the I/O address set in the hardware configuration.

In the configuration example (Chapter 6), address 256 must be transferred to the FB for the axis at the first call. The number of the respective axis data block (axis DB no., e.g. 71) must be transferred to the FB.

Two ANY pointers must be transferred for each axis in order to transfer the process data. The process data should preferably be in a DB (ANY pointer example: p#db71.dbx 166.0 byte 20).

If the process data is to be stored in a DB, one of the following UDTs must be used:

- UDT 30010 for positioning mode with traversing blocks and MDI via message frame 110
- UDT 30008 for positioning mode with traversing blocks and MDI via message frame 111
- UDT 30009 for speed-controlled operation

This is recommended because the individual bits or the process data are symbolized in the UDT.

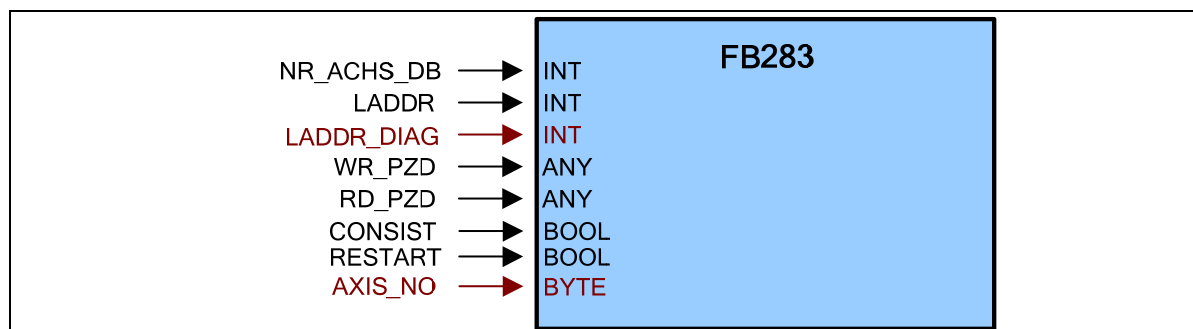


Fig. 7 FB283 input parameters

The block can be used in every S7-300/400 CPU that supports DP-V1 or PROFINET IO.

Sample call in STL (Siemens message frame 110, PZD-12/7)

```
CALL FB283, DB283
NR_ACHS_DB      := 71,
LADDR           := 256,
LADDR_DIAG      := 8188,
WR_PZD          := P#DB71.DBX172.0 BYTE 24,
RD_PZD          := P#DB71.DBX212.0 BYTE 14,
CONSIST         := TRUE,
AXIS_NO         := B#16#2;
```

Sample call in STL (Siemens message frame 111, PZD-12/12)

```
CALL FB283, DB283
NR_ACHS_DB      := 72,
LADDR           := 256,
LADDR_DIAG      := 8188,
WR_PZD          := P#DB72.DBX172.0 BYTE 24,
RD_PZD          := P#DB72.DBX212.0 BYTE 24,
CONSIST         := TRUE,
AXIS_NO         := B#16#2;
```

Note

If FB283 is called for several axes, it is important that the individual calls be mutually interlocked (the acyclic communication channel can always only be used by one application). To implement the interlocking, the "Done" bit (addr. 14.2) from the axis DB should be evaluated. Only when the "Done" bit is set can the call be started for the next axis.

Conditions for the settings of the data consistency

The data consistency over the entire length is specified in the parameter **CONSIST**.

With an inconsistent data transfer over the word, the parameter must be set to **False**. With a consistent data transfer over the the entire length, the parameter must be set to **True**. The setting must match the specification in the hardware configuration.

Description of the formal parameters of FB 283:

Signal	Type	Data type	Value range	Comment
NR_ACHS_DB	I	Int	CPU-dependent	Number of the data block for the axis DB
LADDR	I	Int		Start of the I/O address (cycl. com. → PZD)
LADDR_DIAG	I	Int		Diagnostics address (acycl. com.)
WR_PZD	I	Any	P#Mm.n byte x.. P#DBno.dbxm.n byte x	<p>Target range for process data, master → slave (control words / setpoints)</p> <p>Generally the axis DB is used here, i.e. in the pointer, the same DB no. must be specified as formal parameter "NR_ACHS_DB"</p> <p>The length of the pointer depends on the message frame. Standard message frame 1: 4 bytes Siemens message frame 111: 24 bytes</p>
RD_PZD	I	Any	P#Mm.n byte x.. P#DBno.dbxm.n byte x	<p>Target range for process data, master ← slave (status words / actual values)</p> <p>Generally the axis DB is used here, i.e. in the pointer, the same DB no. must be specified as formal parameter "NR_ACHS_DB"</p> <p>The length of the pointer depends on the message frame. Standard message frame 1: 4 bytes Siemens message frame 111: 24 bytes</p>
CONSIST	I	Bool		<p>True: The PZD area is "Constant over the entire length". Transfer of process data to the areas specified at WR_PZD/RD_PZD is performed with SFC 14/15.</p> <p>False: The PZD area is consistent over the unit. Process data is transferred via load/transfer commands.</p> <p>Please refer to your hardware configuration for the required setting.</p>
RESTART	I	Bool		Starts first initialization
AXIS_NO	I	Byte		<p>The axis no. or drive object ID (DO_ID) of the respective axis must be specified here. The DO_ID generally starts at 2 and can be taken from the configuration overview in STARTER.</p> <p>(Only relevant for parameter access)</p>

3.2. Parameterization input „AXIS_NO“

Dependent on the used drive the input „AXIS_NO“ of the function block must be parameterized as follow.

3.2.1. SINAMICS G120

	Module ID / AXIS_NO
Single axis	1

3.2.2. SINAMICS G130 / G150

	Module ID / AXIS_NO
Control Unit (CU320)	1
Drive (Vector)	2
Module TM31 (-A60)	3
Module TM31 (-A61)	4

3.2.3. SINAMICS S

Refer to CU-Parameter p101 (i.e. p978) for the individual Module ID of the drives objects.

3.3. Configuration guide

Initial situation: A new S7 project has been created. The hardware has been configured.

There are two procedures:

- Opening and changing project FB283_Bsp_V21 (see A)
- A functioning project exists, the interface must be inserted (see B)

A) Changing project FB283_Bsp_V21

The example contains a data block created by means of UDTs for each operating mode

- Speed-controlled mode DB70 (with associated call via FC70)
 - Positioning mode block + MDI (MF110) DB71 (with associated call via FC71)
 - Positioning mode block + MDI (MF111) DB72 (with associated call via FC72)
 - Positioning mode block + MDI (APC) DB73 ((with associated call via FC73)
- APC example: "SINAMICS S120 EPOS (appl. no.: A4027118-N00142-A0434) V2.0"

Opening of the block container and editing of the data blocks 70 to 73. Removal of any UDTs that are not required. If several drive objects are required, additional axis data blocks can be inserted by copying.

B) Inserting an interface in the project

1. Inserting the FB283 and the required UDTs in the existing S7 project.

UDT30000 = Basic UDT

UDT30001 = Edit 64 traversing blocks

UDT30002 = Read out faults

UDT30007 = Positioning mode with traversing blocks and MDI in accordance with the APC
example "SINAMICS S120 EPOS (appl. no.: A4027118-N00142-A0434) V2.0"

UDT30008 = Positioning mode with traversing blocks and MDI in accordance with message
frame 111

UDT30009 = Speed-controlled mode

UDT30010 = Positioning mode with traversing blocks and MDI in accordance with message
frame 110

2. Create a data block that contains the required UDTs for each axis.

UDT30000 must always be contained in the axis data block and called first.

See examples of positioning and speed-controlled mode.

Examples of the data block declaration:

Positioning mode with message frame 110

Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+0.0	Basis	"UDT_Basis"		Basis UDT30000
+162.0	pos	"UDT_TVb+MDI_TLC110"		Positioning UDT30010
+252.0	tvb	"UDT_64TraversingBlocks"		Traversing Blocks UDT30001
+1928.0	fault	"UDT_FaultBuffer"		Fault Buffer UDT30002
=2708.0		END_STRUCT		

Positioning mode according to APC application example

Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+0.0	Basis	"UDT_Basis"		Basic UDT30000
+162.0	pos	"UDT_TVb+MDI_APC"		Positioning UDT30007
+252.0	tvb	"UDT_64TraversingBlocks"		Traversing Blocks UDT30001
+1928.0	fault	"UDT_FaultBuffer"		Fault Buffer UDT30002
=2708.0		END_STRUCT		

Speed-controlled mode

Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+0.0	Basis	"UDT_Basis"		Basic UDT30000
+162.0	Speed_Control	"UDT_SpeedControl"		Speed Control UDT30009
+252.0	Fault	"UDT_FaultBuffer"		Fault Buffer UDT30002
=1032.0		END_STRUCT		

You now have two options:

- A) Manage the internal data with a multi-instance data block.
- B) Assign a separate data block to each axis.

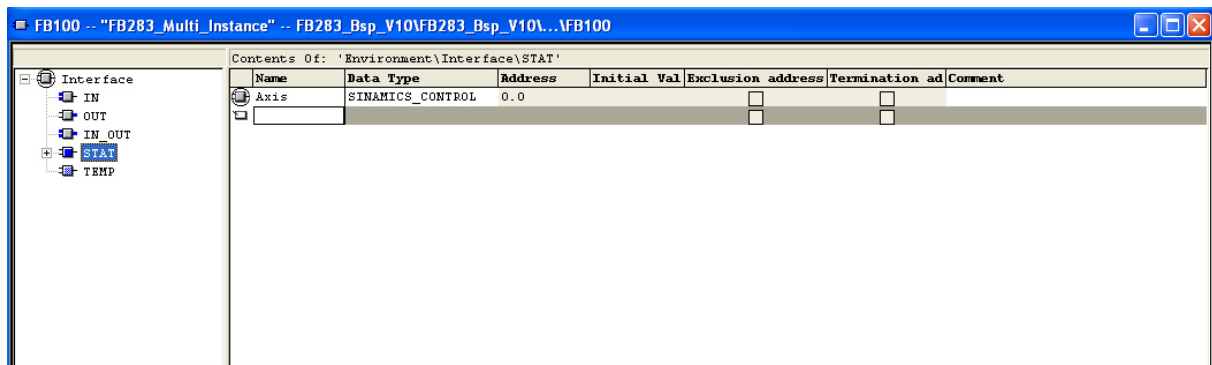
A) Manage the internal data with a multi-instance data block

Create a function block (call function block).

Declare a call variable for each axis in the call function block.

Declaration = stat, **Name** = freely selectable, **Type** = FB283

Example:



- The interface FBs (FB283) are called in this function block or call function block (FB100).
- The FB283 must be called once for each axis.
- Set the call for the call function block at the location.
Example: Call FB100, DB100 → whereby DB100 is the new multi-instance DB of the interface
- Set up error OBs if required (OB 81, 82, 86, 87, 121, 122).

B) Assign a separate data block to each axis

Call the FB283 in the program or in a new FC or FB.

Call command: Call FB283, DB XY → DB XY is the instance DB (one for each axis)

Open the new data block and assign it to FB283 as instance data block.

Properties - Instance data block for FB 283

General - Part 1 | General - Part 2 | Calls | Attributes

Name: DB283

Symbolic Name: InstanceDB_to_FB283

Symbol Comment: Instanz-DB zum FB283

Created in Language: DB

Project Path: FB283_Bsp_V10\FB283_Bsp_V10\Bausteine\DB283

Storage location of project: D:\Program Files\Siemens\Step7\s7proj\FB283_Bs

	Code	Interface
Date created:	02/09/2007 05:19:44 PM	
Last modified:	01/24/2007 02:18:17 PM	01/09/2007 04:21:32 PM

Comment:

OK Cancel Help

If changes are made to the data blocks with regard to the UDT arrangement, perform a restart. Otherwise the changes are not registered by the autoscan.

Notice

The headers of the UDTs and the UDTs themselves must not be changed. If the UDTs are changed, errors can occur in the write and read length.

4. User interface assignment

4.1. List (axis DB)

Internal data								
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBW 0	UDT number							
DBW 2	UDT length							
DBW 4	Control number							
DBW 6	Reserve							
DBW 8	Reserve							
DBW 10	Reserve							

Fault	
DBW 12	ErrorNumberDrive

Single tasks PLC ↔ SINAMICS								
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB 14	Error				Busy	Done	WR	RD
DBW 16	tasksi							
DBW 18	ind							
DBD 20	Data							
DBW 24	ErrorNumbr							
DBW 26	Reserve							
DBW 28	Reserve							
DBW 30	Reserve							
DBW 32	Reserve							

chain_task								
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBW 54	chain_PNU_1							
DBW 56	chain_IND_1							
DBD 58	chain_Data_1							
DBW 62	chain_PNU_2							
DBW 64	chain_IND_2							
DBD 66	chain_Data_2							
DBW 70	chain_PNU_3							
DBW 72	chain_IND_3							
DBD 74	chain_Data_3							
DBW 78	chain_PNU_4							
DBW 80	chain_IND_4							
DBD 82	chain_Data_4							
DBW 86	chain_PNU_5							
DBW 88	chain_IND_5							
DBD 90	chain_Data_5							
DBW 94	chain_PNU_6							
DBW 96	chain_IND_6							
DBD 98	chain_Data_6							
DBW 102	chain_PNU_7							
DBW 104	chain_IND_7							
DBD 106	chain_Data_7							
DBW 110	chain_PNU_8							
DBW 112	chain_IND_8							
DBD 114	chain_Data_8							
DBW 118	chain_PNU_9							
DBW 120	chain_IND_9							
DBD 122	chain_Data_9							
DBW 126	chain_PNU_10							
DBW 128	chain_IND_10							
DBD 130	chain_Data_10							

Complete traversing block								
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB 134	Mode	Command parameter	Command	Deceleration override	Acceleration override	Velocity	Position	Block number
DBW 136	Block number							
DBD 138	Position							
DBD 142	Velocity							
DBD 146	Acceleration override							
DBD 150	Deceleration override							
DBW 154	Command							
DBD 156	Command parameters							
DBW 160	Mode							

4.2. Description

Address	Symbolic name	Description
DBW 0	UDT number	Number of the next UDT starting from here. Initial value: 30000
DBW 2	UDT length	Length of the next UDT starting from here. Initial value: 162
DBW 4	Control number	Initial value: 30000
DBW 12	ErrorNumbrSINAMICS	Error number of the SINAMICS is stored here
DBX 14.0	RD	Reading the data
DBX 14.1	WR	Writing the data
DBX 14.2	Done	Task completed checkback signal
DBX 14.3	Busy	Task is being processed
DBX 14.7	Error	Task aborted with error
DBW 16	tasks_i	Task number. Can be a parameter number or a special task number
DBW 18	ind	Subparameter number. See SINAMICS Description of Functions, Chap. Parameter lists
DBD 20	Data	Value that is to be written or value read out after a read task
DBW 24	ErrorNumbr	Error number that has occurred while executing the task. See Chap. 8 - Error code table
DBW 26	SFB_ERR	Error status of SFB52/53. See online help for these SFBs
DBW 54, 62 ..., 126	chain_PNU_1-10	Parameter number for reading out up to ten arbitrary parameters
DBW 56, 64 ..., 128	chain_IND_1-10	Subparameter number associated with the respective parameter number
DBD 58, 66, ..., 130	chain_Data_1-10	Transferred parameter value from the drive
DBX 134.0	Blocknum	Preselection of which parameters are to be transferred (134.1 to 134.7)
DBX 134.1	pos	Position preselection
DBX 134.2	vel	Velocity preselection
DBX 134.3	accelover	Acceleration preselection
DBX 134.4	decelover	Deceleration preselection
DBX 134.5	cmd	Command preselection
DBX 134.6	cmdpara	Command parameter preselection
DBX 134.7	mod	Mode preselection
DBW 136	Block number	Block number value
DBD 138	Position	Position value
DBD 142	Veloc	Velocity value
DBD 146	Accel_over	Acceleration override value
DBD 150	Decel_over	Deceleration override value
DBW 154	Command	Command value
DBD 156	Command parameters	Command parameter value
DBW 160	Mode	Mode value

5. Traversing task processing

Various tasks can be started with the interface block:

- Read/write individual parameters
- Read out fault buffer
- Read/write individual traversing blocks
- Read/write blocks of traversing blocks
- Preassign traversing blocks 0...63
- Read/write up to ten arbitrary parameters

Note

The task may only be triggered when all the task data is present.

Single task interface

Tasks that only have to be executed once can be triggered with the single task interface. The Busy status bit is set while the task is being executed. When the task has been completed, the Busy bit is reset and the Done bit is set.

The Error bit is set when one or more data items are entered incorrectly:

- Parameter number (tasks_i) is incorrect
- Subparameter number (IND) is incorrect
- Date is incorrect

An error number is displayed in the error box. For a description of the error codes, see, for example, SINAMICS S120 Description of Functions, Chapters 5, 7 and 8.

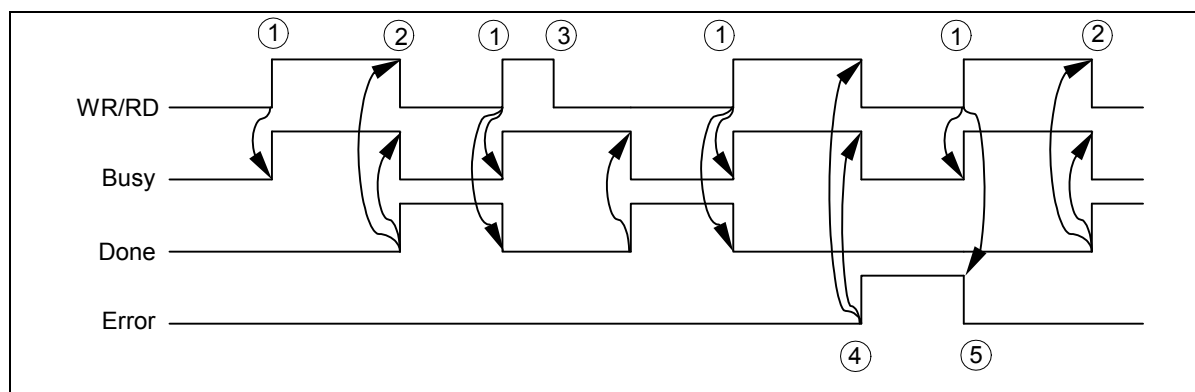


Fig. 8 Structure of the obligatory UDT

- ① Write or read task started by user
- ② WR/RD request from SINAMICS automatically reset
- ③ The task is performed even when the request is withdrawn
- ④ Request is reset when a transfer error occurs
- ⑤ New task resets the error message

Task processing

Two types of tasks can be started with the SINAMICS single task interface:

- Reading and writing all parameters from SINAMICS (see SINAMICS parameter list)
- Special tasks (e.g. reading/writing blocks of traversing blocks)

Read individual parameters

Preconditions:	single.tasksi	=	parameter number
	single.Ind	=	subparameter number
Trigger:	single.RD	=	with 1
End:	single.Done	=	TRUE
Output:	single.Data	=	read-out parameter value

Write individual parameters

Preconditions:	single.tasksi	=	parameter number
	single.Ind	=	subparameter number
	single.Data	=	value to be written
Trigger:	single.WR	=	with 1
End:	single.Done	=	TRUE

Read out fault buffer

Preconditions:	single.tasksi	=	30002
	single.Ind	=	is ignored
Trigger:	single.RD	=	start read-out with 1
End:	single.Done	=	TRUE
Values:	Faults and number of faults are stored in the associated data block in the UDT30002.		

Read individual traversing blocks

Preconditions:	single.tasksi	=	30000
	single.Ind	=	select the traversing block number (0 to 63)
	Bit string 134.0-7	=	preselection of which parameters are to be transferred
Trigger:	single.RD	=	read with 1
End:	single.Done	=	TRUE

Write individual traversing blocks

Preconditions:	single.tasksi	=	30000
	single.Ind	=	select the traversing block number (0 to 63)
	Bit string 134.0-7	=	preselection of which parameters are to be transferred
Trigger:	single.WR	=	write with 1
End:	single.Done	=	TRUE

Read/write blocks of traversing blocks

Preconditions:	single.tasksi	=	30001
	single.Ind	=	first traversing block number
	single.Data	=	last traversing block number
Trigger:	single.RD	=	read with 1
	single.WR	=	write with 1
End:	single.Done	=	TRUE
Values:	Read	=	values are stored in the associated data block in the UDT30001
	Write	=	values are transferred from the UDT30001 from the data block to the SINAMICS

Notice

All data is only stored in the working memory in the SINAMICS.

The data must therefore be backed up to the EEPROM via parameter P971. Otherwise the data is deleted after the next power on.

Perform backup: Set parameter P971 = 1

Preassign traversing blocks 0...63

Preconditions:	single.tasksi	=	30011
	single.Ind	=	is ignored
	single.Data	=	is ignored
Trigger:	single.WR	=	write with 1
End:	single.Done	=	TRUE
Values:	Write	=	block numbers in the drive are preassigned with 0 to 63

Read/write up to ten arbitrary parameters

Preconditions:	single.tasksi	=	30010
	single.Ind	=	specifies the first chain_PNU_xy to be processed
	single.Data	=	specifies the last chain_PNU_xy to be processed
	chain_PNU_xy	=	parameter numbers
	chain_Ind_xy	=	subparameter numbers
Trigger:	single.RD	=	read with 1
	single.WR	=	write with 1
End:	single.Done	=	TRUE
Values:	chain_Data_xy		values are in the respective mailboxes of the parameter numbers

The block can be inserted alternatively in the following OBs:

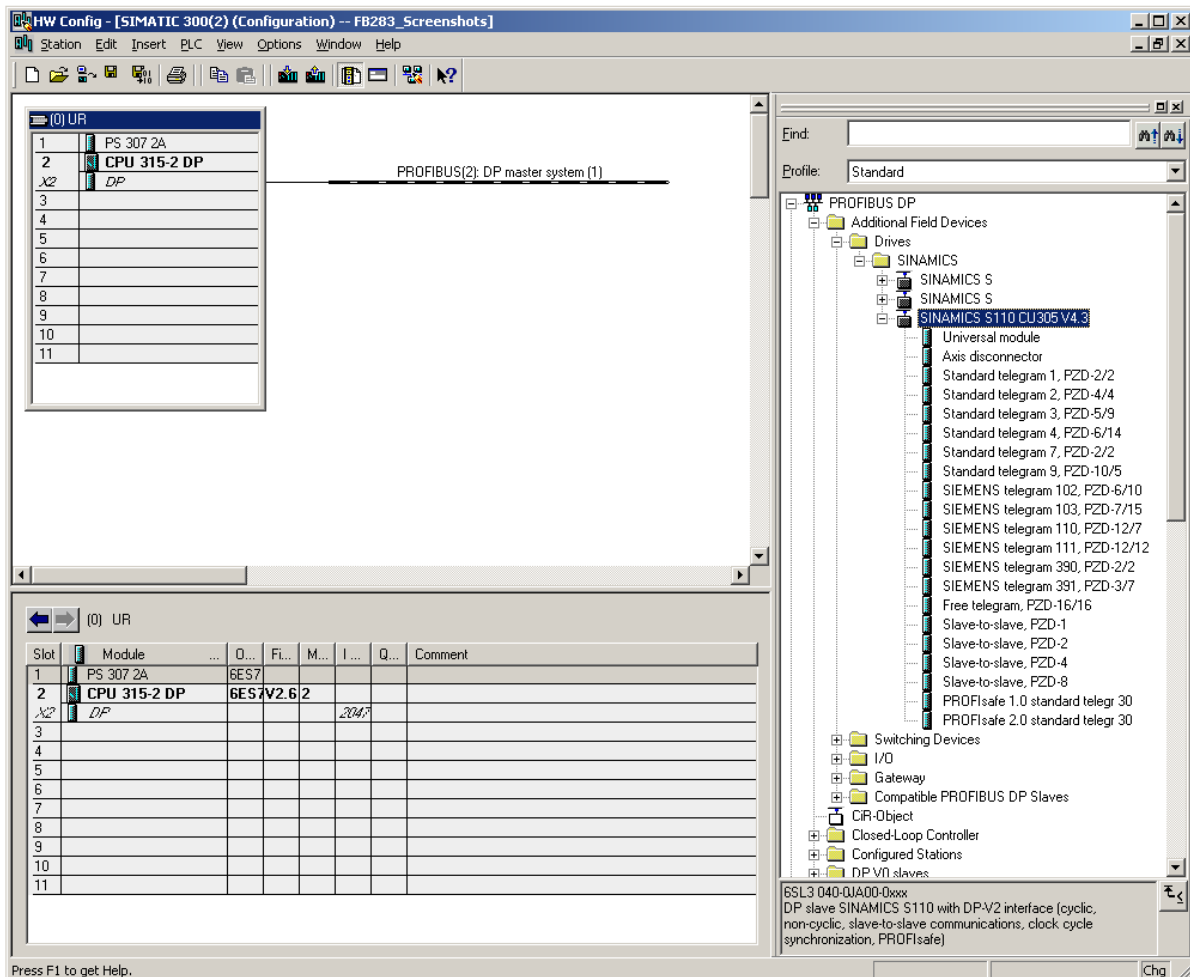
Cyclic task: OB1

Watchdog interrupt OB: E.g. OB32

6. Configuration

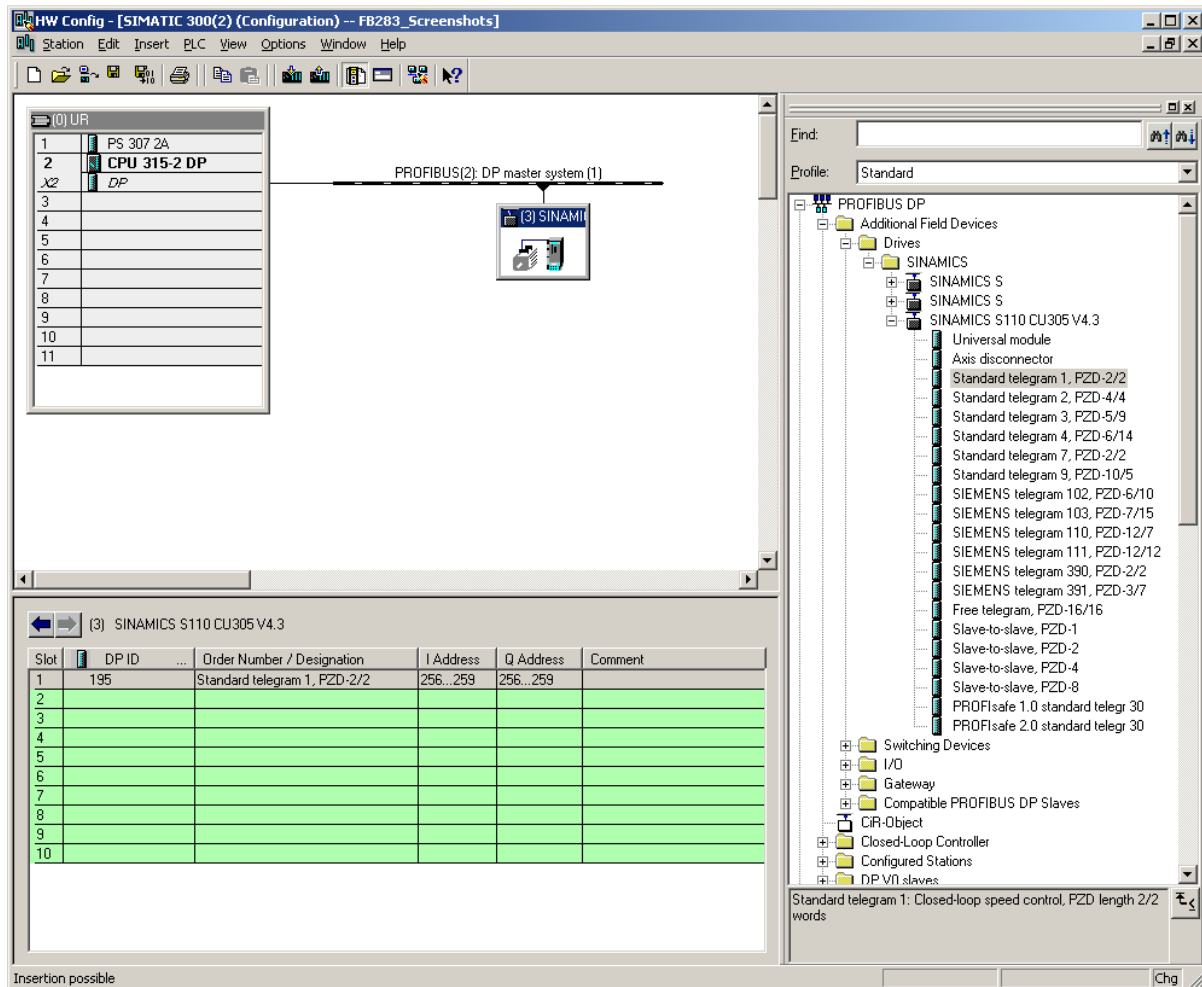
6.1. Standard message frame 1

After configuring the S7 (for example a 315-2 DP), the hardware configuration appears as follows:



Depending on your hardware, insert the SINAMICS S110 module on the PROFIBUS DP, for example. Then select the desired message frame for each axis.

If a single-axis module with standard message frame 1 is selected, your hardware configuration appears as follows:



The I/O addresses 256 to 259 are assigned to drive 1.

Caution

The start address areas for inputs and outputs may not be separated.

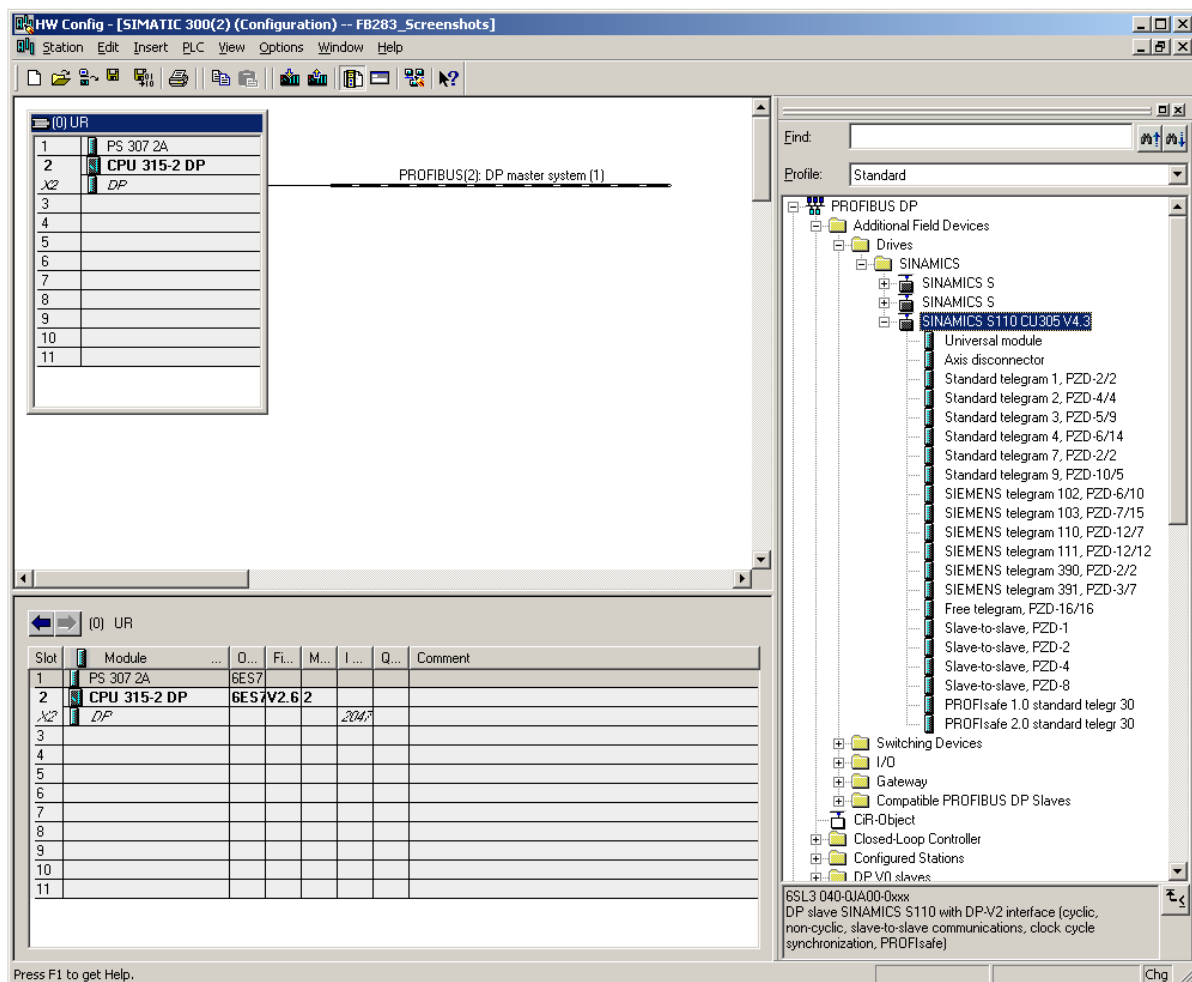
The PZD of the drives must always be directly connected.

The call of FC70 from the sample project can be used for a configuration with "standard message frame 1".

6.2. Siemens message frame 111 – block positioning and MDI

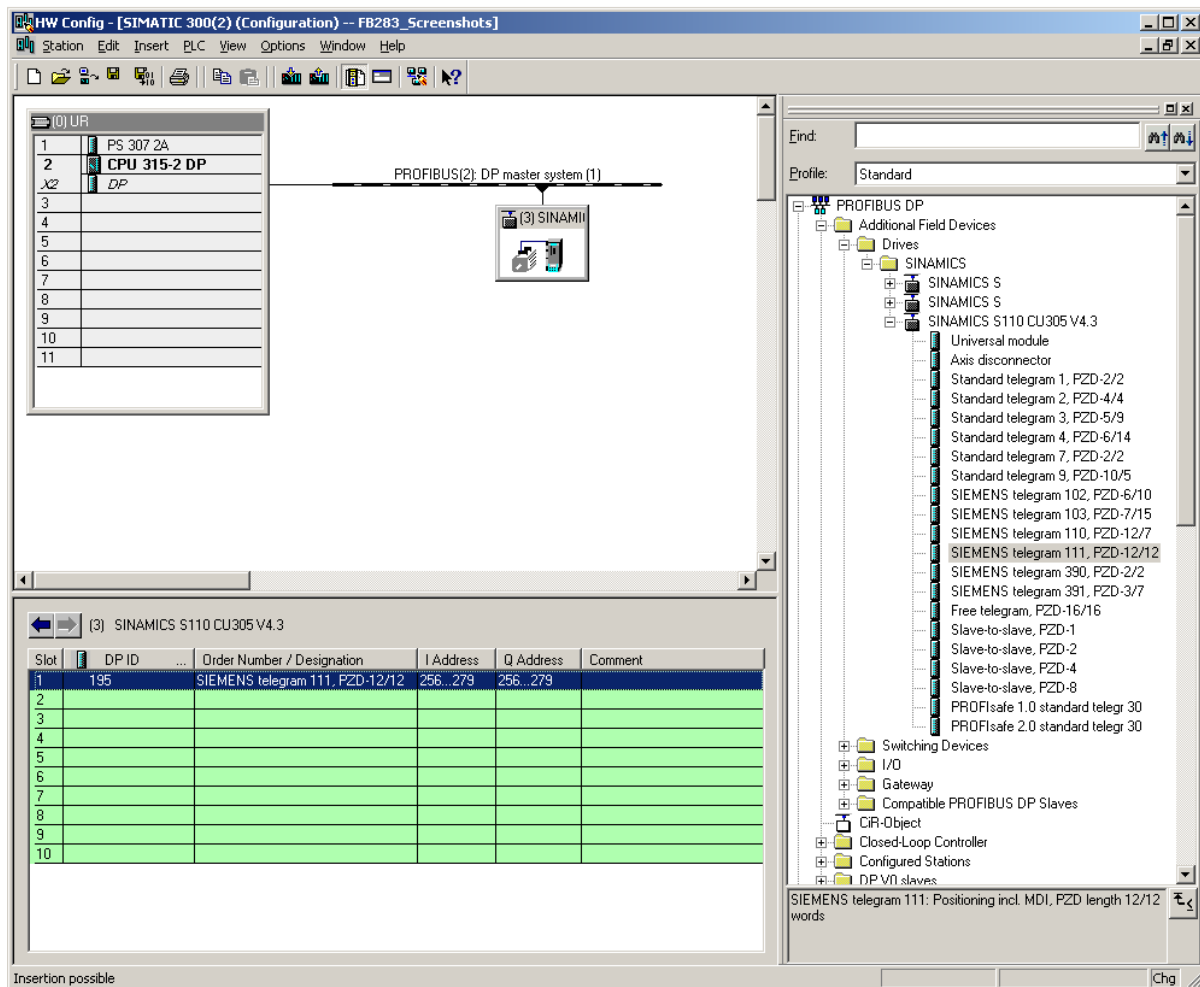
Please refer to the description of functions of your SINAMICS device for the structure of the Siemens message frame 111 and handling of the MDI mode.

After configuring the S7 (for example a 315-2 DP), the hardware configuration appears as follows:



Depending on your hardware, insert the SINAMICS S110 module on the PROFIBUS DP, for example. Then select the desired message frame for each axis.

If a single-axis module with Siemens message frame 111 is selected, your hardware configuration appears as follows:



The I/O addresses 256 to 279 are assigned to drive 1.

Notice

The FB283 can only work with the same start addresses for the I/O area.

If multi-axis configurations are selected, different start addresses for the I/O area may be automatically assigned for the second and the following axes, as these may have different lengths, for example, with Siemens message frame 110.

The start addresses must therefore be manually corrected to the same I/O start addresses.

The call of FC72 from the sample project can be used for a configuration with "SIEMENS message frame 111".

7. Error during task processing

If an error (**single.Error** = true) is output during the task processing, an evaluation can be performed via parameter **single.ErrorNumbr**.

ErrorNumbr	Meaning
10XY	Network 2: SFC6 RET_VAL <> 0
11XY	Network 2: Obligatory UDT30000 is not present in the axis DB or is not at the start of the axis DB <u>Remedy</u> 1.) Restart the CPU 2.) Construct the axis DB according to the documentation, <i>e.g. UDT30000, UDT30001, ..., and your own variables at the end.</i>
20XY	Network 3: SFC15 RET_VAL <> 0, error while writing the process data
21XY	Network 3: SFC21 RET_VAL <> 0, error while reading the status words
22XY	Network 3: SFC20 RET_VAL <> 0, error while copying from #rdz to #hilf_zsw
30XY	Network 4: Task type number has an invalid value
40XY	Network 5: single.taski = 0
41XY	Network 5: single.taski has an invalid value
42XY	Network 5: Task 30011 "Preassign traversing blocks 0...63" cannot be started with a pos. edge from the bit single.RD
43XY	Network 5: Task 30002 "Read out fault buffer" cannot be started with a pos. edge from the bit single.WR
50XY	Network 6: Task step number has an invalid value
51XY	Network 6: Incorrect return value (ParameterValues.Format) from the drive
60XY	Network 7: Task step number has an invalid value
61XY	Network 7: Internal program error
70XY	Network 8: Task step number has an invalid value
71XY	Network 8: single.Ind < 0 (permitted values: 0...63)
72XY	Network 8: single.Ind > 63 (permitted values: 0...63)
80XY	Network 9: Task step number has an invalid value
81XY	Network 9: Optional UDT30001 is not present in the axis DB, i.e. task "Read/write block of traversing blocks" cannot be executed. <u>Remedy</u> 1.) Restart the CPU 2.) Construct the axis DB according to the documentation, <i>e.g. UDT30000, UDT30001, ..., and your own variables at the end.</i>
82XY	Network 9: single.Ind > 63 (permitted values: 0...63)
83XY	Network 9: single.Ind < 0 (permitted values: 0...63)
84XY	Network 9: single.Data > 63 (permitted values: 0...63)
85XY	Network 9: single.Data < single.Ind (note: single.Ind must be <= single.Data)
86XY	Network 17: single.Ind > 10 (permitted values: 1...10)
87XY	Network 17: single.Ind < 1 (permitted values: 1...10)
88XY	Network 17: single.Data > 10 (permitted values: 1...10)
89XY	Network 17: single.Data < 1 (permitted values: 1...10)
8AXY	Network 17: single.Ind <= single.Data (note: single.Ind must be <= single.Data)
90XY	Network 10: Task step number has an invalid value

91XY	Network 10: Optional UDT30002 is not present in the axis DB, i.e. task "Read out fault buffer" cannot be executed. <u>Remedy</u> 1.) Restart the CPU 2.) Construct the axis DB according to the documentation, e.g. UDT30000, UDT30001, ..., and your own variables at the end.
92XY	Network 10: Internal program error during the task generation - branch destination list
93XY	Network 10: Internal program error during the task evaluation - branch destination list
94XY	Network 10: Internal program error during the task evaluation - error code
95XY	Network 10: Internal program error during the task evaluation - error number
96XY	Network 10: Internal program error during the task evaluation - error value
A0XY	Network 12: Task step number has an invalid value
B0XY	Network 12: Task step number has an invalid value
B1XY	Network 12: A value could not be read
B2XY	Network 12: Internal program error during the task evaluation
C0XY	Network 13: Task step number has an invalid value
C1XY	Network 13: A value could not be written
D0XY	Network 14: Error during the acyclic write access by means of SFB53 (for SFB errors, see below)
D1XY	Network 14: Error during the acyclic read access by means of SFB52
D2XY	Network 14: Internal program error
E0XY	Network 15: Error during the acyclic access by means of SFB
E1XY	Network 14/15: Request error task completed with errors (response ID = 0x81/82); additional tasks aborted. For more detailed error numbers, see instance DB DBW322 (Response.Error_Nr).
F0XY	Network 17/18: Task step number has an invalid value
F1XY	Network 17: Incorrect return value (ParameterValues.Format) from the drive

....X..	Task type
.....Y	Task step

8. Testing aids

8.1. Standard message frame 1

The following variable tables can be used for the control/demonstration of a speed-controlled SINAMICS from a SIMATIC CPU.

Variable table	Description
VAT70_FaultBuffer	Read out fault buffer
VAT70_Para_1_10	Read/write one to ten parameters
VAT70_Parameter	Read/write a single parameter
VAT70_SpeedControl	Traverse drive with speed control

The correlated FC70 in OB1 must always be called.

8.2. SIEMENS message frame 111

The following variable tables can be used for the control/demonstration of the various positioning functions of a SINAMICS from a SIMATIC CPU.

Variable table	Description
VAT72_FaultBuffer	Read out fault buffer
VAT72_MDI	Traverse axis with the MDI function
VAT72_Para_1_10	Read/write one to ten parameters
VAT72_Parameter	Read/write a single parameter
VAT72_TVB	Traverse axis with the traversing blocks function
VAT72_TVBlock	Transfer 1 to 64 traversing blocks
VAT72 TVBsingle	Transfer a single traversing block

The correlated FC72 in OB1 must always be called.

To
SIEMENS AG
A&D MC PM
P.O. Box 3180
D-91050 Erlangen, Germany

Suggestions**Corrections**

For document:

SINAMICS

Function block FB 283
for SINAMICS ↔ S7 CPU

Sender

Name

Address of your Company/Dept.

Street

Postal code:

Location:

Phone:

/

Fax:

/

Description

Edition 07/2011

Should you come across any printing errors when reading this publication, please notify us on this sheet. Suggestions for improvement are also welcome.