SIEMENS

Function Block Description FB283 SINAMICS S120 <--> S7-CPUs

Function description **Edition: 07/2008**

Safety Instructions

This Manual contains information which you should carefully observe to ensure your own personal safety and the prevention of damage to the system. This information is highlighted by a warning triangle and presented in one of the following ways depending on the degree of risk involved:

DANGER

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

WARNING

indicates that death, severe personal injury or substantial property damage **can** result if proper precautions are not taken.

CAUTION

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

NOTE

contains important information about the product, its operation or a part of the document to which special attention is drawn.

Qualified person

The unit may only be started up and operated by a **qualified person or persons**. Qualified persons as referred to in the safety guidelines in this manual are those who are authorized to start up, earth and label units, systems and circuits in accordance with relevant safety standards.

Rules for proper use

Please note the following:

WARNING

The unit may be used only for the applications described in the catalog or the technical description, and only in combination with the equipment, components and devices of other manufacturers as far as this is recommended or permitted by Siemens.

The successful and safe operation of this equipment is dependent on proper transportation, storage, erection and installation and on careful operation and maintenance.

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 his own purposes may violate the rights of the registered holder.

Siemens AG
Bereich Automatisierungs- und Antriebstechnik (A&D)
Geschäftsgebiet Motion Control Systems (MC)
Postfach 3180, D-91050 Erlangen

SINAMICS S120

Description

Function Block FB283
SINAMICS S120 ↔ S7-CPU

Valid for
Function Block FB283, >= Version 1.3

List of contents

1. General Information

This block supports data transfer between a SINAMICS S120 and an S7-CPU via PROFIBUS and PROFINET. This block can be used in each S7-CPU 300/400 that is it least DP-V1-capable. The data interface is individually defined in a data block for each drive by structuring with pre-configured UDTs.

The principle communication of the SINAMICS S120 and a PROFIBUS master is shown in Fig. 1. The process data (PZD) is consistently transferred over the complete length of the PZD area. The control and feedback signals are precisely defined in the User Interface Assignment. The request data (parameters, indices and values) are transferred using non-cyclic (acyclical) services, e.g. PROFIBUS DP-V1.
The techniques applies in the same way for data exchange with a PROFINET IO controller.

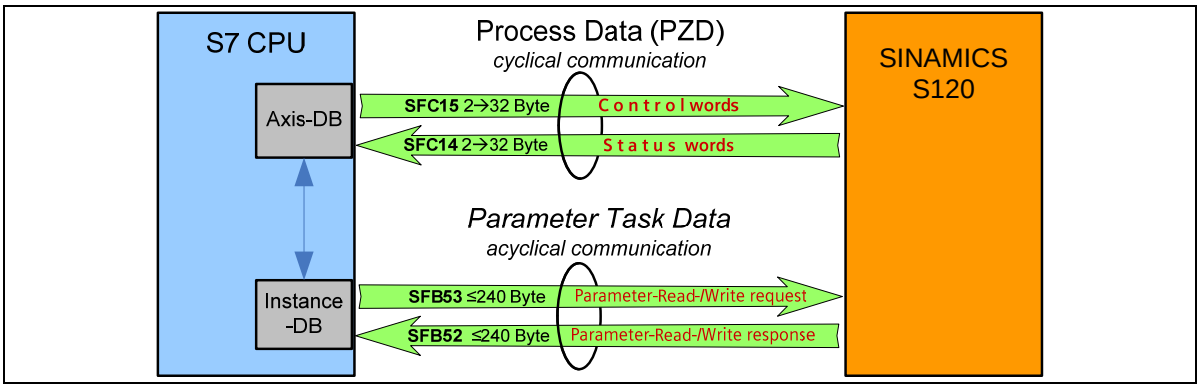


Fig. 1 Example: Consistent data transfer

This block can be used in the following OBs:
Cyclical Task: OB1
Timer Alarm Task: e.g. OB32

NOTE

The language settings for this block can be changed to the listet languages within the SIMATIC Manager:

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

Request data interface

The communication sequence of the acyclical parameter interface (also refer to PROFIdrive Profile Drive Technology → Profibus User Organization, Order No. 3.172) is shown in Fig. 2. The requests are transferred in FB283 using the system function blocks SFB52 and 53.

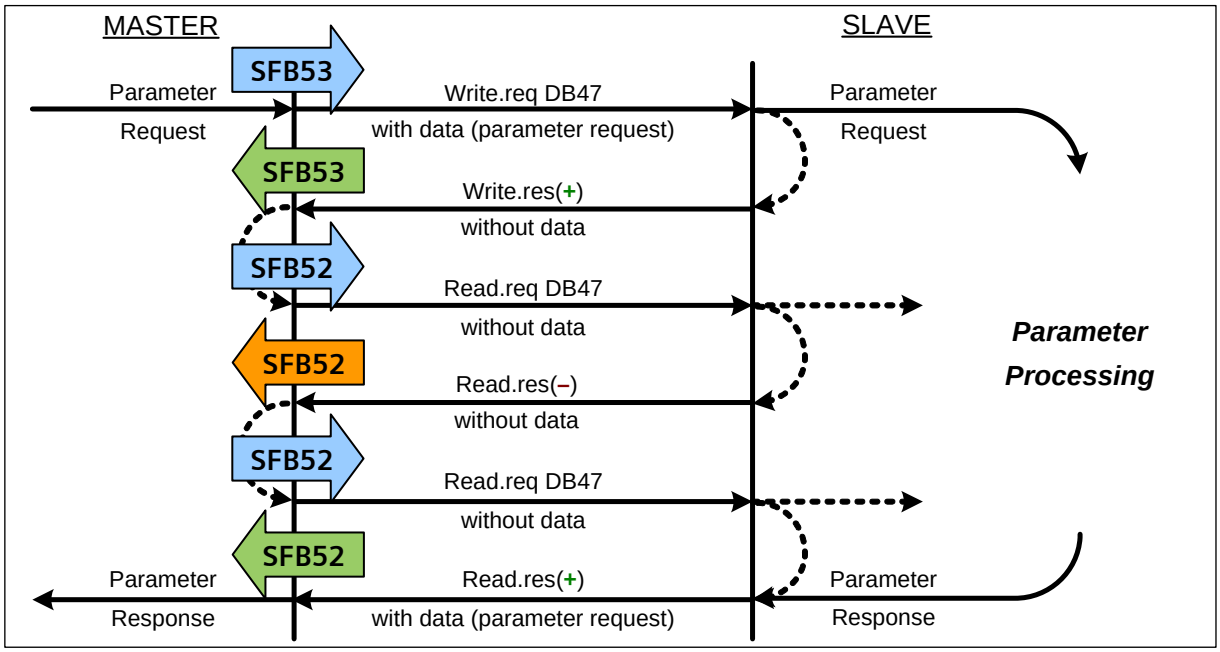


Fig. 2 Coordination of the parameter task jobs using acyclical communication

2. User Interface Structure

The principle structure of the user interface is shown in Fig. 3. One axis-specific data block must be generated using pre-configured UDTs for each axis. The process data area (**either** UDT positioning operation **or** UDT speed operation) is absolutely mandatory; SINAMICS S120 is controlled and its feedback signals are mapped using this process data area. Optional data, such as e.g. traversing blocks and fault signals, can be individually integrated into the axis-specific data block depending on the particular requirements. FB283 must be cyclically called once for each axis.

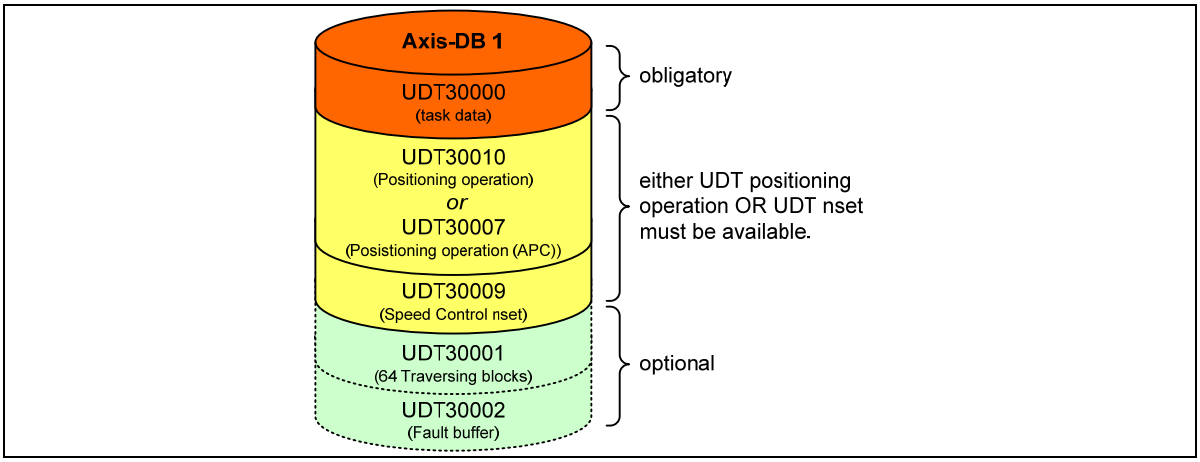


Fig. 3 Principle structure of the user interface

In the first S7-CPU cycle or after setting the input "RESTART" FB283 checks which data areas the user defined in the axis DB (Fig. 4). This information is stored in the internal data of the FB283 (→instance DB). If the structure of an axis DB is modified, then the S7-CPU must be restarted (new start or cold start) or the input "RESTART" must be set.

Caution

Block FB283 must be called during the first PLC cycle. Otherwise, the FB283 will not be initialized after a PLC restart and the structure of the axis DBs will not be recognized and saved in the instance DB. SFC6 is used to implement the UDTs Autoscan.

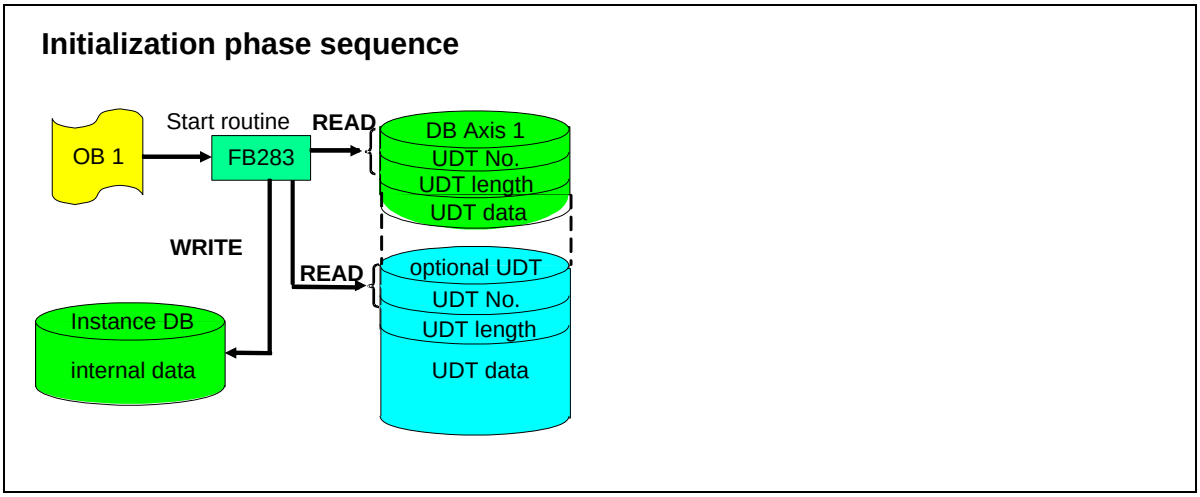


Fig. 4 Checking the data areas using FB283

The structure of the obligatory UDTs is shown in Fig. 5.
The precise assignment of the UDTs is provided in the User Interface Assignment.

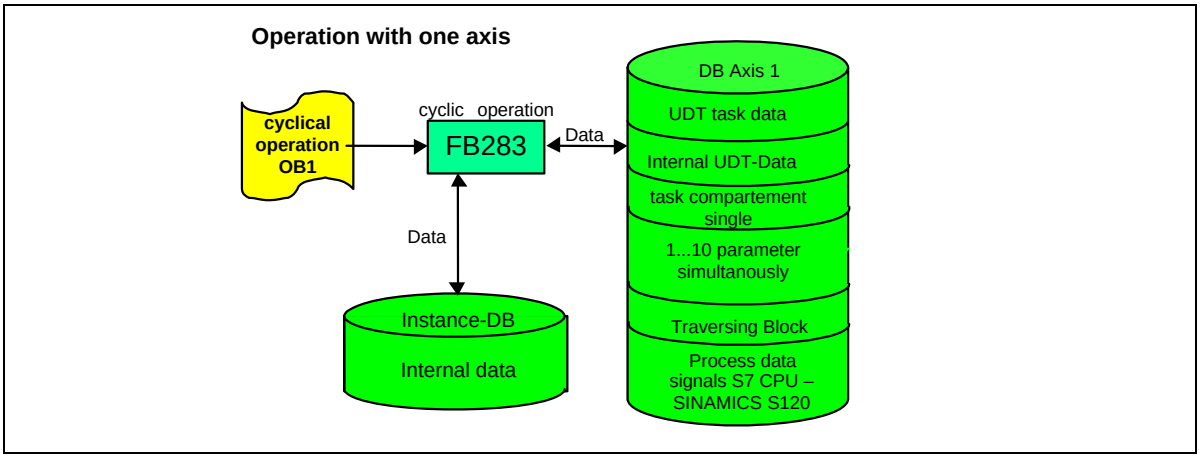


Fig. 5 Structure of the obligatory UDTs

For operation with n axes, only n+1 DBs are required if the internal data is saved in a multi-instance DB, refer to Fig. 6.

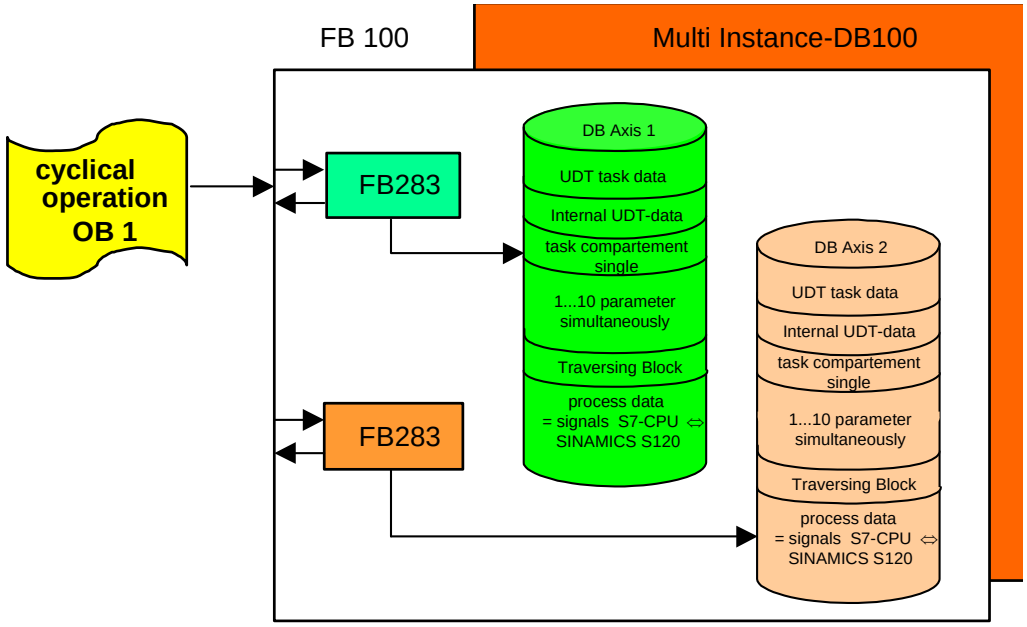


Fig. 6 Operation with multiple axis

3. Calling the Function Block

3.1. General information

Function block FB283 must be cyclically called for each axis. Communication is realized via the peripheral address set in the hardware configuration.
In the configuration example (Chapter 6) the FB must be transferred when first calling address 256 for the axis. The number of the particular axis data block (axis DB No. e.g. 71) must be transferred to the FB.

In order to transfer process data two “any” pointers must be transferred for each axis. The process data should preferably be located in one DB (any pointer, for example: p#db71.dbx 166.0 byte 20).

If process data is to be saved in a DB the following UDTs can be used:

- UDT 30010 for positioning operation with traversing blocks and MDI via telegram 110
- *UDT 30008 for positioning operation (reserved for future applications)*
- UDT 30009 for speed controlled operation

This is recommended as the individual bits of the process data are symbolized in the UDT.

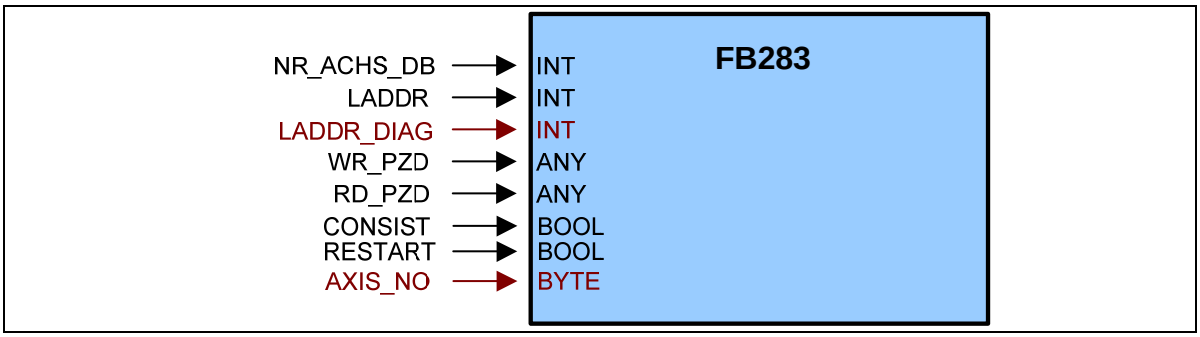


Fig. 7 Input parameters of the FB283

This block can be used in any S7-CPU 300/400 capable of DP-V1 or PROFINET IO.

Example of a call in STL (Siemens telegram 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;
```

Conditions for the data consistency settings

The data consistency over the complete length is defined in parameter **CONSIST**.

For an inconsistent data transfer via the word, the parameter must be set to **false**. If consistent data transfer over the complete length exists, then the parameter should be set to **true**. The setting must match the data in the hardware configuration.

Explanation of the formal parameters of FB 283:

Signal	Type	Char.	Value range	Comment
NR_ACHS_DB	I	Int	CPU-dependent	Number of the data block for axis DB
LADDR	I	Int		Start of the I/O address (cyclic com. → PZD)
LADDR_DIAG	I	Int		Diagnostics address (acycl. com.)
WR_PZD	I	Any	P#Mm.n Byte x.. P#DBnr.dbxm.n Byte x	Target area for process data, master –> slave (control words/setpoints) Generally, the axis DB is used here; i.e. the same DP No. must be specified in the pointer as at the formal parameter "NR_ACHS_DB" The pointer length depends on the telegram. Standard telegram 1: 4 bytes Siemens telegram 110 24 bytes
	I	Any	P#Mm.n Byte x.. P#DBnr.dbxm.n Byte x	Target area for process data, master <– slave (status words/actual values) Generally, the axis DB is used here; i.e. the same DP No. must be specified in the pointer as at the formal parameter "NR_ACHS_DB" The pointer length depends on the telegram. Standard telegram 1: 4 bytes Siemens telegram 110 14 bytes
CONSIST	I	Bool		**True:** The PZD area is "constant over the complete length" transfer of the process data in the area specified under WR_PZD/RD_PZD is realized using SFC 14/15. **False:** The PZD area is consistent over the unit. Process data is transferred using load/transfer commands. Take the required setting from your hardware configuration.
RESTART	I	Bool		Starts the first initialization
AXIS_NO	I	Byte		The axis No. and DriveObject_ID (DO_ID) of the axis to be addressed should be specified here. Generally, the DO_ID starts at 2 and can be taken from the configuration overview in Starter. (only relevant for parameter access operations)

3.2. Configuration instructions

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

There are two ways to proceed:
- Open and change the project FB283_Bsp_V13 (refer to A)
- There is a functioning project, the interface must be inserted (refer to B)

A) Changing the project FB283_Bsp_V13

The example already contains the data block for the modes generated using UDTs
- Speed-controlled operation DB70 (with the associated call via FC70)
- Positioning operation, block + MDI (TLG110) DB71 (with the associated call via FC71)
- *Positioning operation (reserved) DB72 (with the associated call via FC72)*
- Positioning operation, block + MDI (APC) DB73 (with the associated call via FC73)
  APC example: "SINAMICS S120 EPOS" (Appl. No.: A4027118-N00142-A0434 V2.0

Open the block container and process data blocks 70 to 73. Remove any UDTs that are not required. If several drive objects are required then additional axis data blocks can be inserted by copying.

B) Inserting an interface into the project

1. Insert the FB283 and the required UDTs into the existing S7 project.
   UDT30000 = basis UDT
   UDT30001 = processing 64 traversing blocks
   UDT30002 = read-out faults
   UDT30007 = positioning operation with traversing blocks and MDI acc. to the APC example
   „SINAMICS S120 EPOS (Appl. No.: A4027118-N00142-A0434) V2.0“
   UDT30008 = positioning operation (reserved)
   UDT30009 = speed-controlled operation
   UDT30010 = positioning operation with traversing blocks and MDI acc. to telegram 110

2. For each axis, create a data block that contains the required UDTs.
   UDT 30000 must always be contained in the axis data block and must be called at the first position. Refer to examples for positioning and speed controlled operation.

Examples for declaring a data block:

Positioning operation with telegram 110

Address	Name	Type	Establish connection to configured CPU	alue	Comment
0.0		STRUCT			
+0.0	Basis	"UDT_Basis"			Basis UDT30000
+162.0	pos	"UDT_TVB+MDI_TLG110"			Positioning UDT30010
+252.0	tvb	"UDT_64TraversingBlocks			Traversing Blocks UDT30001
+1928.0	fault	"UDT_FaultBuffer"			Fault Buffer UDT30002
=2708.0		END_STRUCT			

Positioning operation according to the APC application example

Address	Name	Type	Initial value	Comment
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 operation

Address	Name	Type	Initial value	Comment
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		

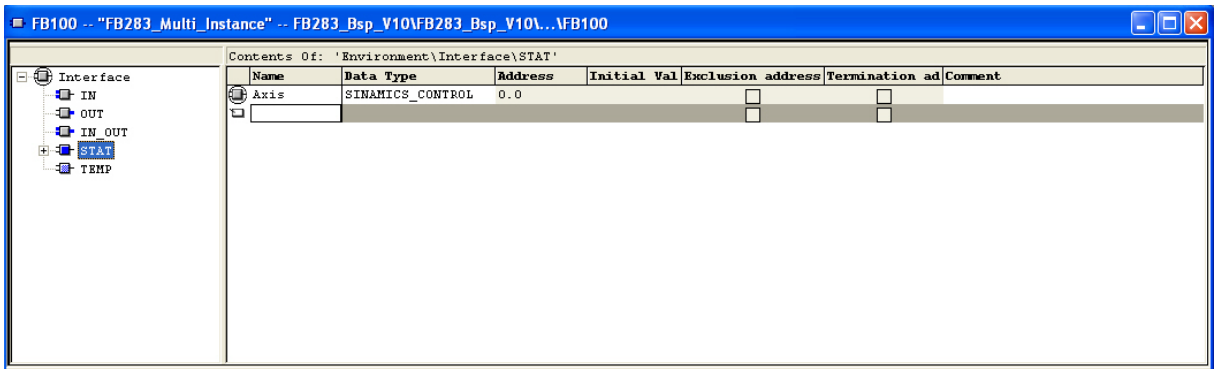
You have 2 possibilities:

A) Manage internal data using a multi-instance data block.
B) Assign each axis a dedicated data block.

A) Managing internal data using a multi-instance data block

Create a function block (call block).
For each axis declare a call variable in the call block.
Declaration = stat, **name** = freely selectable, **type** = FB283

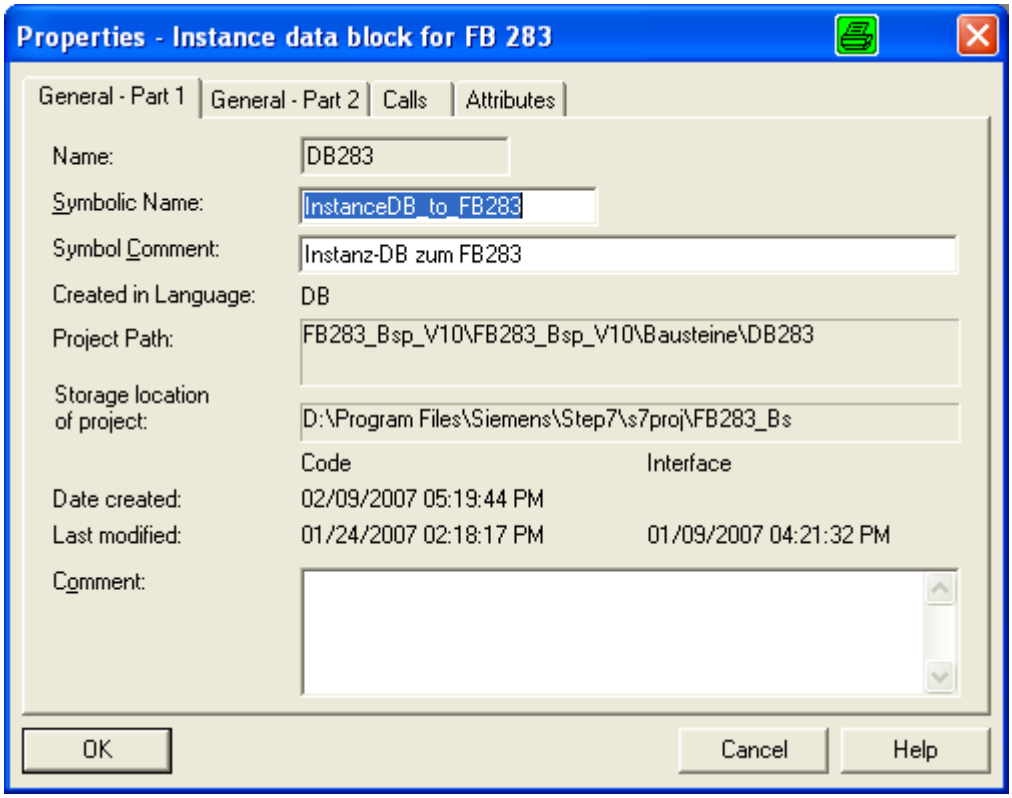
Example:



- The interface FBs (FB283) are called in this function block or in the call block (FB100).
- FB283 must be called once for each axis.
- Where it is used, set the call for the call block.
  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 each axis its own dedicated data block

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 (must be unique for each axis)



If changes are made to the data blocks regarding the UDT arrangement, then a new start must be carried-out. Otherwise, Autoscan will not register the changes.

Caution
The headers of the UDTs and the UDTs themselves may, under no circumstances be changed. If UDTs are changed then errors can occur regarding 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	Check number							
DBW 6	Reserve							
DBW 8	Reserve							
DBW 10	Reserve							

Error	
DBW 12	ErrorNumberDrive

Requests-single PLC ⇔ SINAMICS S120								
	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	Com-mand	Decelera-tion override	Accelera-tion 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 parameter							
DBW 160	Mode							

4.2. Description

Address	Symbol. name	Description
DBW 0	UDT Nummer	Number of the UDTs following from here. Starting value: 30000
DBW 2	UDT Länge	Length of the UDTs in bytes following from here. Starting value: 162
DBW 4	Kontrollnummer	Starting value: 30000
DBW 12	ErrorNumbrSINAMICS	Error number of the SINAMICS S120 is saved here.
DBX 14.0	RD	Reads the data
DBX 14.1	WR	Writes the data
DBX 14.2	Done	Feedback signal, request completed
DBX14.3	Busy	Request is being processed.
DBX 14.7	Error	Request interrupted by an error.
DBW 16	tasksi	Request number. Can be a parameter number or special request number.
DBW 18	ind	Sub-parameter number. Refer to the Function Description SINAMICS S120, Chapter Parameter lists.
DBD 20	Data	Value that is to be written or after a read request, value that has been read out.
DBW 24	ErrorNumbr	Error number that has occurred when executing the request. Refer to Chapter 8 - error code table
DBW 26	SFB_ERR	Fault state of SFB52/53; see Online-Help fort his SFBs
DBW 54, 62 ..., 126	chain_PNU_1-10	Parameter number to read-out up to 10 parameters of any type
DBW 56, 64 ..., 128	chain_IND_1-10	Sub-parameter number associated with the particular parameter number.
DBD 58, 66, ..., 130	chain_Data_1-10	Parameter value transferred from the S120
DBX 134.0	Satznum	Pre-selection which parameters should be transferred (134.1 to 134.7)
DBX 134.1	pos	Pre-selection, position
DBX 134.2	gesch	Pre-selection, velocity
DBX 134.3	beschover	Pre-selection, acceleration
DBX 134.4	verzover	Pre-selection, deceleration
DBX 134.5	bef	Pre-selection, command
DBX 134.6	befpara	Pre-selection, command parameter
DBX 134.7	mod	Pre-selection, mode
DBW 136	Satznummer	Value, block number
DBD 138	Position	Value, position
DBD 142	Geschw	Value, velocity
DBD 146	Beschl_over	Value, acceleration override
DBD 150	Verzoeg_over	Value, deceleration override
DBW 154	Befehl	Value, command
DBD 156	Befehlsparameter	Value, command parameter
DBW 160	Modus	Value, mode

5. Traversing Task Processing

Various requests can be started using the interface block:

- Individually read/write parameters
- Read-out the fault memory
- Read/write individual traversing blocks
- Read/write traversing blocks
- Pre-assign traversing blocks 0...63
- Read/write up to 10 parameters of any type

Note

The request may only be initiated if all request data are available.

Request interface, single

Using the request interface single, requests can be initiated that are only executed once. The status bit Busy is set while the request is running. If the request has been completed Busy is reset and the Done bit is set.

The error bit is set if one or several pieces of data are incorrectly entered:

- Parameter number (tasksi) is incorrect
- Sub-parameter number (IND) is incorrect
- Date ((data)) is incorrect

An error number is specified in the error compartment. Refer to the Function Description of SINAMICS S120, Chapters 5, 7 and 8 for a description of the error code.

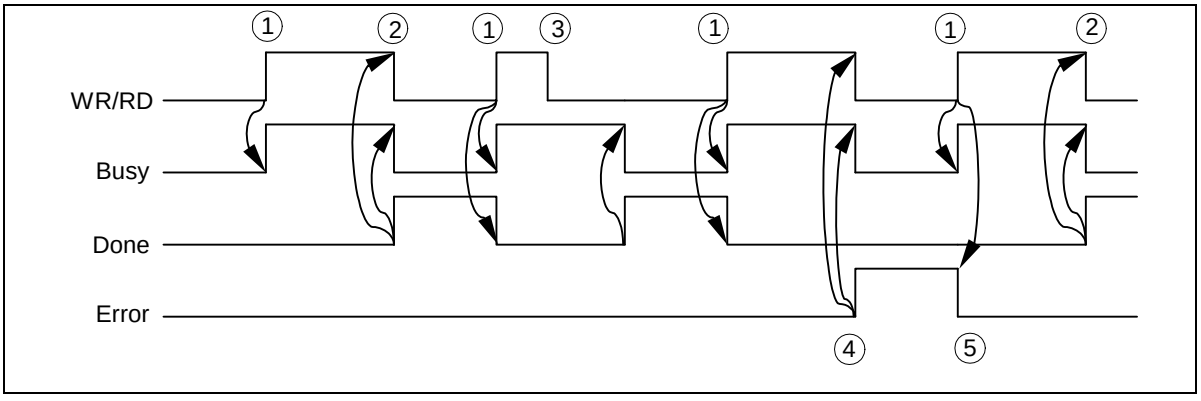


Fig. 8 Timing diagram for parameter task execution

① Write or read request, started by the user
② WR/RD request automatically reset from SINAMICS S120
③ Also when the request is withdrawn, the request is completed
④ The request is reset for a data transfer error
⑤ New request resets the error signal

Request administration

Using the S120 request interface single two types of requests can be started:

- Reading/writing all parameters from SINAMICS S120
  (refer to the SINAMICS S120 parameter list)
- Special requests (e.g. Reading or writing traversing block)

Individually read parameters

Pre-conditions:	single.tasksi	=	parameter number
	single.Ind	=	sub-parameter number
Initiate:	single.RD	=	with 1
End:	single.Done	=	TRUE
Output:	single.Data	=	parameter value that has been read-out

Individually write to parameters

Pre-conditions:	single.tasksi	=	parameter number
	single.Ind	=	sub-parameter number
	single.Data	=	value to be written
Initiate:	single.WR	=	with 1
End:	single.Done	=	TRUE

Reading-out the error memory

Pre-conditions:	single.tasksi	=	30002
	single.Ind	=	is ignored
Initiate:	single.RD	=	start to read-out with 1
End:	single.Done	=	TRUE
Values:	Error situations and the number of errors are saved in the associated data block in UDT30002.		

Reading individual traversing blocks

Pre-conditions:	single.tasksi	=	30000
	single.Ind	=	selects traversing block number (0 to 63)
	Bitleiste 134.0-7	=	pre-selects which parameters are to be transferred
Initiate:	single.RD	=	reads with 1
End:	single.Done	=	TRUE

Writing individual traversing blocks

Pre-conditions:	single.tasksi	=	30000
	single.Ind	=	selects traversing block number (0 to 63)
	Bitleiste 134.0-7	=	pre-selects which parameters are to be transferred
Initiate:	single.WR	=	writes with 1
End:	single.Done	=	TRUE

Reading/writing traversing blocks

Pre-conditions:	single.tasksi	=	30001
	single.Ind	=	first traversing block number
	single.Data	=	last traversing block number
Initiate:	single.RD	=	reads with 1
	single.WR	=	writes with 1
End:	single.Done	=	TRUE
Values:	lesen	=	values are saved in the associated data block in UDT30001
	schreiben	=	values are transferred from the UDT30001 from the data block to the S120

Caution

All data is only saved in SINAMICS S120 in the working memory.

This is the reason that data must be backed-up in the EEPROM using parameter P971. Otherwise, data will be deleted when powering-up again.

Carry-out data back-up: Set parameter P971 to 1

Pre-assigning traversing blocks 0...63

Pre-conditions:	single.tasksi	=	30011
	single.Ind	=	is ignored
	single.Data	=	is ignored
Initiate:	single.WR	=	writes with 1
End:	single.Done	=	TRUE
Values:	schreiben	=	block numbers in the drive are pre-assigned with 0 to 63

Reading/writing up to 10 parameters of any type

Pre-conditions:	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	=	sub-parameter numbers
Initiate:	single.RD	=	reads with 1
	single.WR	=	writes with 1
End:	single.Done	=	TRUE
Values:	chain_Data_xy values are located in the particular data compartments of the parameter numbers		

This block can be used in the follwing OBs:

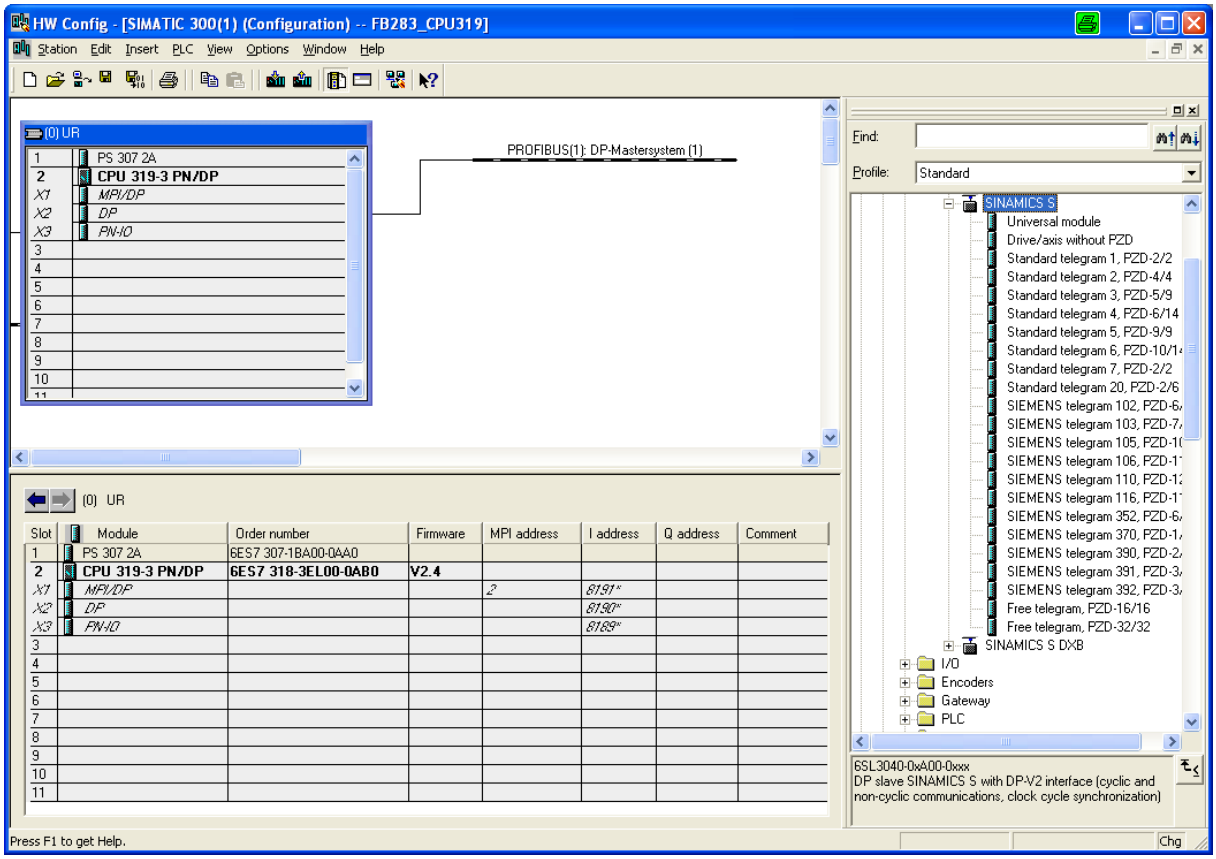
Cyclical Task: OB1

Timer Alarm-OB: z.B. OB32

6. Configuring

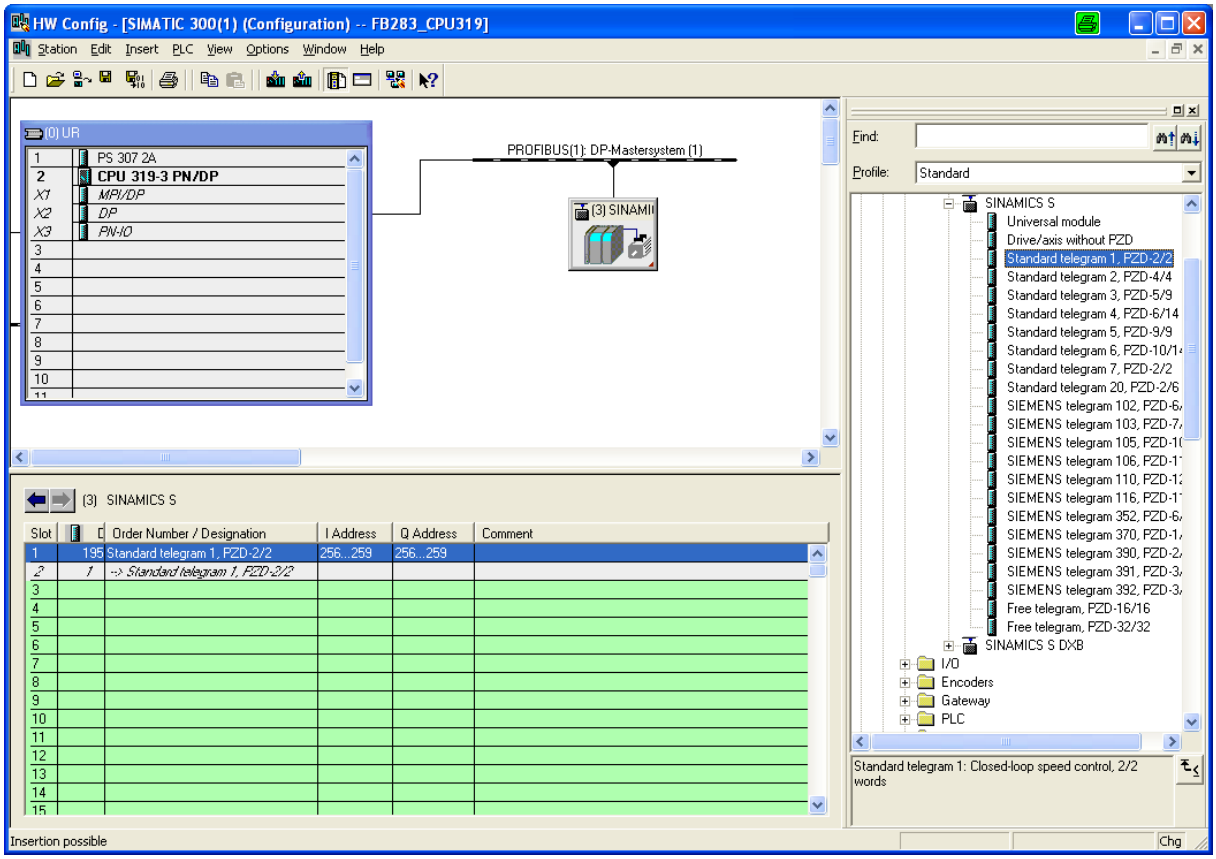
6.1. Standard telegram 1

After configuring the S7 (for example, a 319-3DP/PN) the following screen is displayed in the hardware configuration:



Attach the SINAMICS S120 module to PROFIBUS-DP.
Then select the number of axes and the standard telegram required.

When selecting a single-axis module with standard telegram 1 your hardware configuration looks like this.



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

Caution

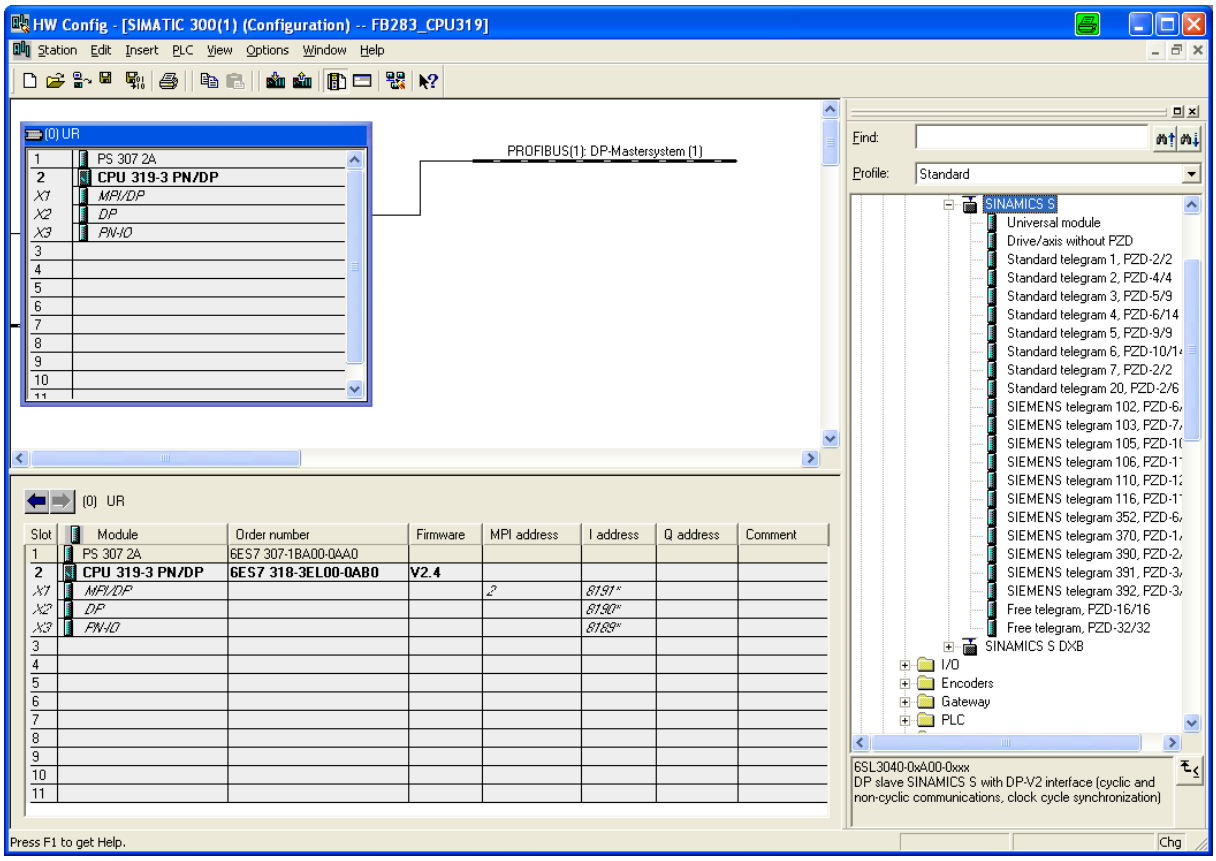
It is not permissible to separate the starting address areas for inputs and outputs.

The PZD of the drives must always be consecutively assigned (i.e. one after the other without any gaps)

6.2. Siemens telegram 110 – block positioning and MDI

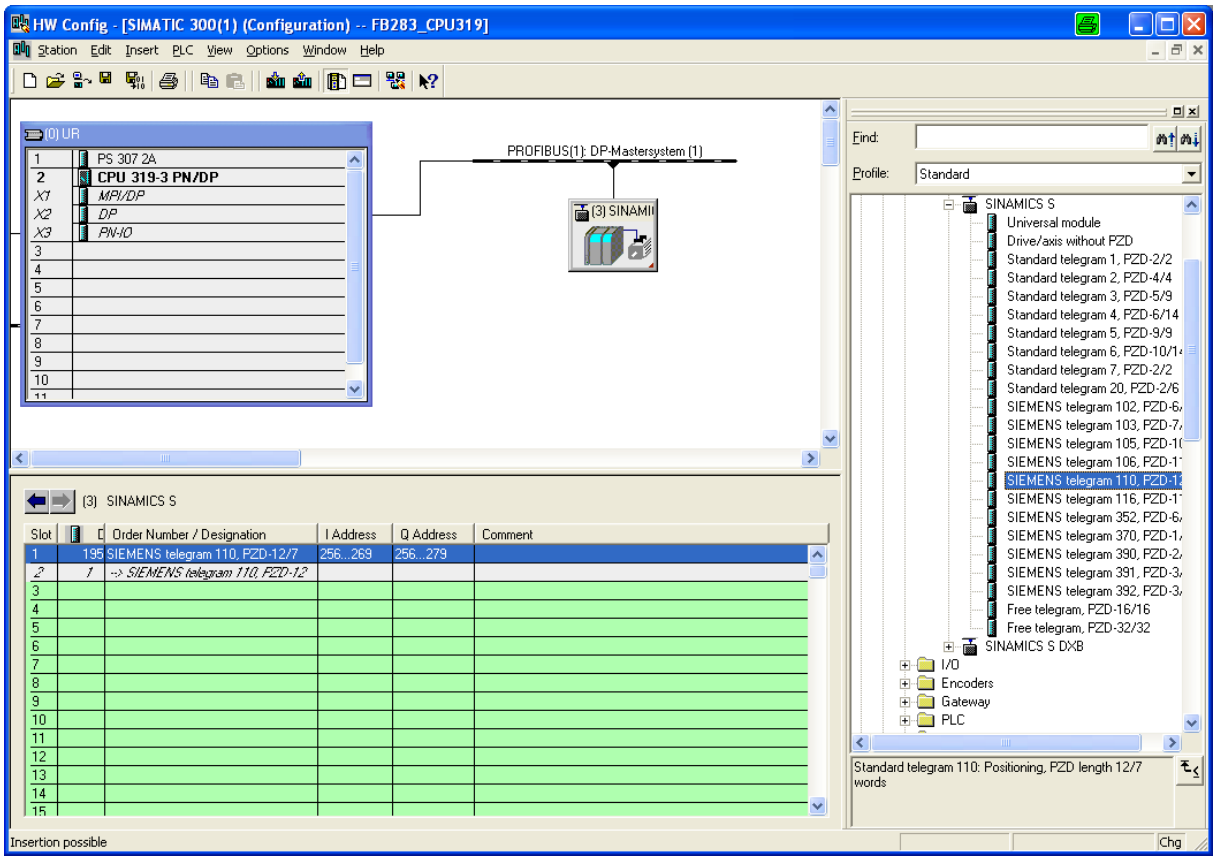
Please refer to the SINAMICS S120 Function Description for the structure of Siemens telegram 110 and handling MDI operation.

After configuring the S7 (for example, a 319-3DP/PN) the following screen is displayed in the hardware configuration:



Attach the SINAMICS S120 module to PROFIBUS-DP.
Then select the number of axes and the Siemens telegram required.

When selecting a single-axis module with Siemens telegram 110 your hardware configuration looks like this.



I/O addresses 256 to 269 and 256 to 279 are assigned to drive 1.

Caution

FB283 can only function when the starting addresses for the I/O areas are identical.

When selecting multi-axis configurations, it is possible that different starting addresses are assigned automatically for the I/O areas for the second and following axes. This is because of Siemens telegram 110 the I/O areas have different lengths.

This is the reason that starting addresses must be manually corrected to ensure that the same I/O starting addresses are used (identical starting addresses).

7. Error Handling

If an error is indicated (**single.Error** = true) when executing a request, then the error can be evaluated using parameter **single.ErrorNumbr**.

ErrorNumber	Significance
10XY	Network 2: SFC6 RET_VAL <> 0
11XY	Network 2: Obligatory Udt30000 not present in the axis DB or is not at the start of the axis DB *Remedy* ***1.) Restart the CPU*** *2.) Establish the axis DP corresponding to the documentation,* *e.g. UDT30000, UDT30001, ..., and at the end own variables.*
20XY	Network 3: SFC15 RET_VAL <> 0, error when writing to process data
21XY	Network 3: SFC21 RET_VAL <> 0, error when deleting status words
22XY	Network 3: SFC20 RET_VAL <> 0, error when copying from #rdz to #hilf_zsw
30XY	Network 4: #AuftrArt has an invalid value
40XY	Network 5: single.taski = 0
41XY	Network 5: single.taski has an invalid value
42XY	Network 5: The request 30011 "Pre-assign traversing blocks 0..63" cannot be started with a pos. edge from bit single.RD
43XY	Network 5: The request 30002 "Read-out fault memory" cannot be started with a pos. edge of bit single.WR
50XY	Network 6: #AuftrSchritt has an invalid value
51XY	Network 6: Incorrect return value (ParameterValues.Format) from SINAMICS S120
60XY	Network 7: #AuftrSchritt has an invalid value
61XY	Network 7: Internal program error
70XY	Network 8: #AuftrSchritt has an invalid value
71XY	Network 8: singl.Ind < 0 (permitted values: 0...63)
72XY	Network 8: singl.Ind >63 (permitted values: 0...63)
80XY	Network 9: #AuftrSchritt has an invalid value
81XY	Network 9: Optional UDT30001 not present in the axis DB, i.e. the request "Read / write traversing block" cannot be executed. *Remedy* ***1.) Restart the CPU*** *2.) Structure the axis DB corresponding to the documentation,* *e.g. UDT30000, UDT30001, ..., and at the end own variables*
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: #AuftrSchritt has an invalid value

91XY	Network 10: Optional UDT30002 not present in the axis DB, i.e. the request "Read-out error memory" cannot be executed. *Remedy* ***1.) Restart the CPU*** *2.) Structure the axis DB corresponding to the documentation,* *e.g.UDT30000, UDT30001, ..., and at the end own variables*
92XY	Network 10: Internal program error when generating a request - jump bar
93XY	Network 10: Internal program error when evaluating a request - jump bar
94XY	Network 10: Internal program error when evaluating a request - error code
95XY	Network 10: Internal program error when evaluating a request - error number
96XY	Network 10: Internal program error when evaluating a request - error value
A0XY	Network 10: #AuftrSchritt has an invalid value
B0XY	Network 12: #AuftrSchritt has an invalid value
B1XY	Network 12: A value was not able to be read
B2XY	Network 12: Internal program error when evaluating a request.
C0XY	Network 13: #AuftrSchritt has an invalid value
C1XY	Network 13: A value was not able to be written
D0XY	Network 14: Error for acyclic write access using SFB53 (SFB errors, refer below)
D1XY	Network 14: Error for acyclic read access using SFB52
D2XY	Network 14: Internal program error
E0XY	Network 15: Error when acyclically accessing using SFB
E1XY	Network 14/15: Request error Response contains error identifier, further requests canceled The detailed error indication is given in the Instance-DB, DBW322 (Response, Error-No.)
F0XY	Network 17/18: #AuftrSchritt has an invalid value
F1XY	Network 17: Incorrect return value (ParameterValues.Format) from SINAMICS S120

....X..	***Request type***
.....Y	***Request step***

8. Test Aids

Variable tables can be used to control / demonstrate the MDI function using a SIMATIC-CPU:
VAT71_MDI / VAT73_MDI_APC

The traversing data blocks can be changed / demonstrated using the following variable tables:
VAT71_TVB / VAT71_TVBblock

The correlating FC71 or FC73 in OB1 must always be called for this purpose.

To
SIEMENS AG
A&D MC PM
Postfach 3180
D-91050 Erlangen

Recommendations

Corrections

For documentation:

SINAMICS S120

Function Block FB283
for SINAMICS S120 ↔ S7-CPU

Sender

Name

Address of your company/department

Street

Postal Code: City:

Telephone: /

Telefax: /

Description

Edition: 07/2008

If, when reading this document you come across printing errors then please inform us using this form.
We would also be grateful for any suggestions or recommendations for improvement.