RSC-D250PROFIBUS • April, 2024
Published for the Basler Electric Power Systems Group

Application Note

Example of configuring the Basler Electric DECS-250 as a PROFIBUS Slave module with a Siemens SIMATIC S7 PLC

This document provides an example of the process for configuring a DECS-250 Digital Excitation Control System as a PROFIBUS slave to a Siemens Automation SIMATIC S7 PLC. This document should be used only as an example. The examples provided here may need to be adjusted and your results may vary if different equipment or software is used.

The SIMATIC S7 and SIMATIC DP master interface are products and intellectual property of Siemens Automation and are not related to, or endorsed by Basler Electric. All questions or concerns regarding these products should be directed to Siemens Automation. Basler Electric Company shall not be held liable for any issues related, directly or indirectly, to the use of this document or any instructions contained therein.

Product and Network Information

The DECS-250 instruction manual is available from the Basler Electric website at www.basler.com.

A GSD file for the DECS-250 is available from Basler Electric's website at https://www.basler.com/Downloads/ or via email at info@basler.com.

Technical information about the PROFIBUS protocol is available from the PROFIBUS International website, www.profibus.com.

Information about Siemens Automation PLCs and the PROFIBUS master system is available from the Siemens Automation website, www.automation.siemens.com.

Equipment and Software Used in this Example

Description	Name / Type	Version
Siemens Automation SIMATIC S7 PLC	IM 151-7 CPU	6ES7 151-7AA20-0AB0 / V2.6
DP Master Interface	SIMATIC DP, DP master interface for IM 151-7 CPU OR IM 151-8 PN/DP (ET200S), module with PROFIBUS DP interface (9 pin sub-D, female) as DP master	6ES7 138-4HA00-0AB0
Siemens PLC programming cable	Not Applicable	Not Applicable
Automatic Voltage Regulator	DECS-250	Style number xxxxxxP
GSD file for DECS-250 *	BASL0D1F.GSD	1.00
Profibus standard cable	Not Applicable	Not Applicable
Power supply, 24Vdc	Not Applicable	Not Applicable

* To obtain a DECS-250 GSD file, visit https://www.basler.com/Downloads/ or contact Basler Electric via email at info@basler.com.

Solution Overview

Figure 1 illustrates the system configuration that was used in the examples of this document.

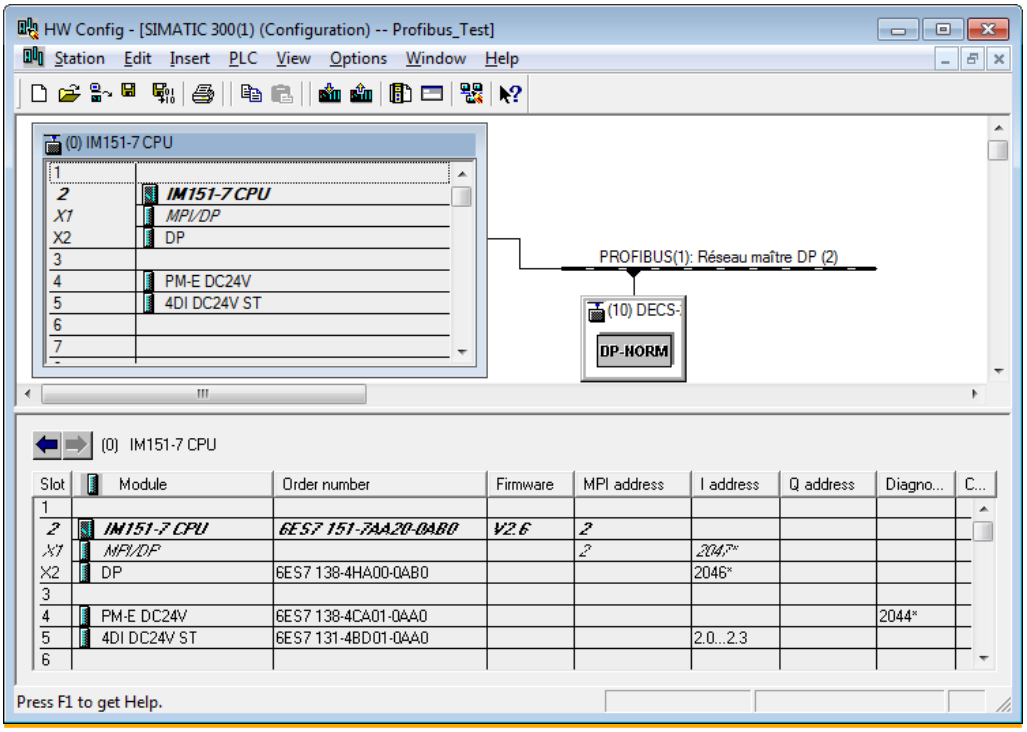

Figure 1. Hardware Connection Overview

Hardware Configuration

The DECS-250 PROFIBUS address was set using the BESTCOMS*Plus*® application.

PLC Configuration

PLC system hardware configuration was performed with the SIMATIC STEP 7 software interface. The PLC was inserted as shown in Figure 2.

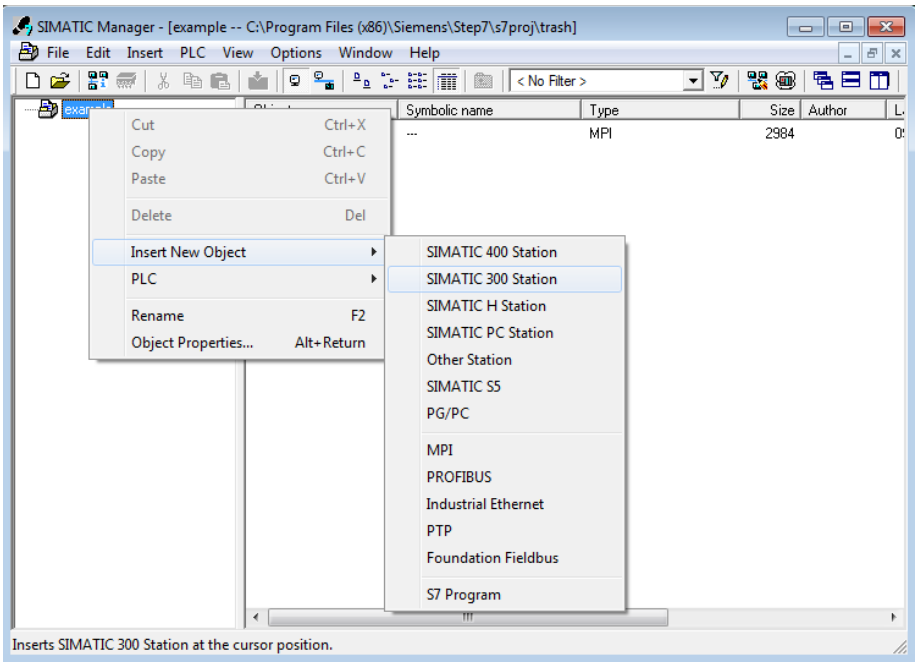

Figure 2. Inserting a New PLC

The hardware configuration was opened by double-clicking on the new SIMATIC 300 station and then on Hardware object name. The left side of Figure 3 shows the software elements to be selected and the right side of Figure 3 shows the network when completed.

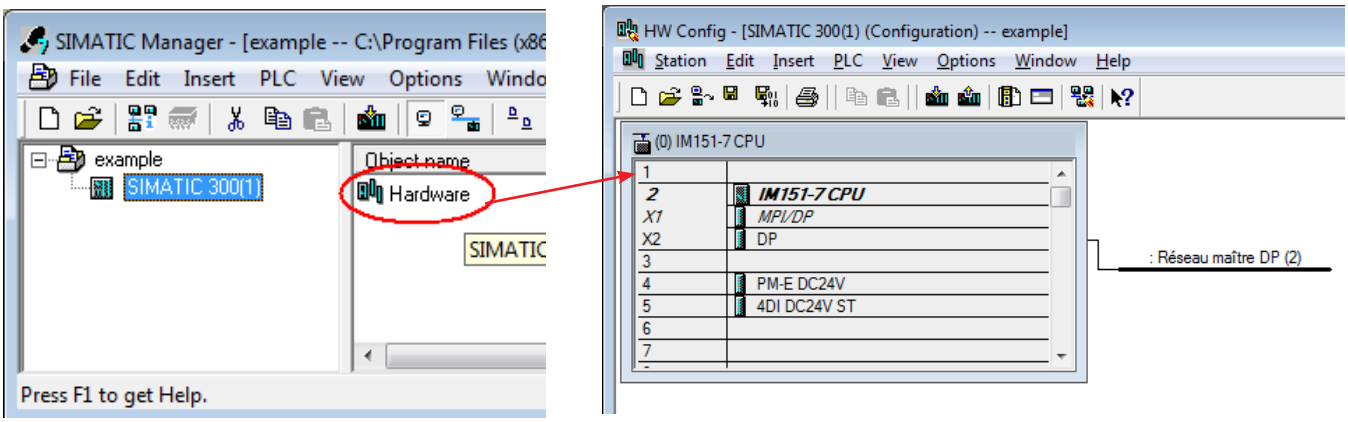

Figure 3. Opening the Hardware Configuration

A rail, the power module and the PLC was added as shown in Figure 4. The next step was to double-click on the DP, PROFIBUS master module to configure the PROFIBUS network.

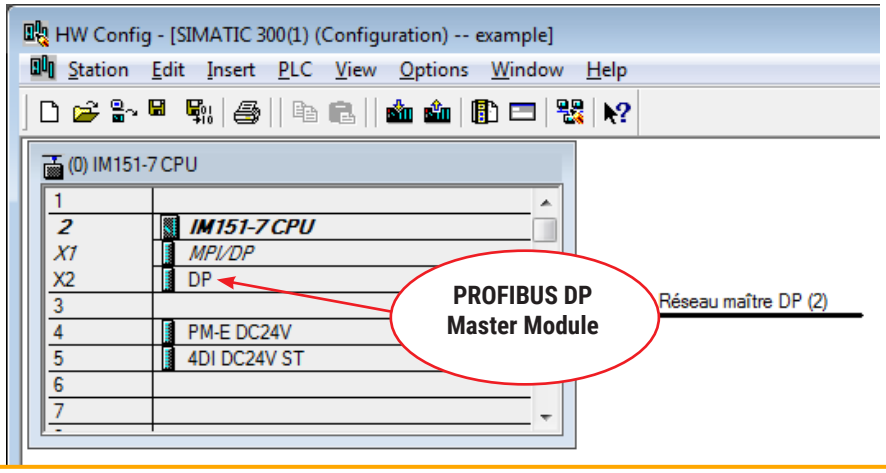

Figure 4. Adding the Hardware to the Configuration

Then, as shown in Figure 5, the Properties button was clicked.

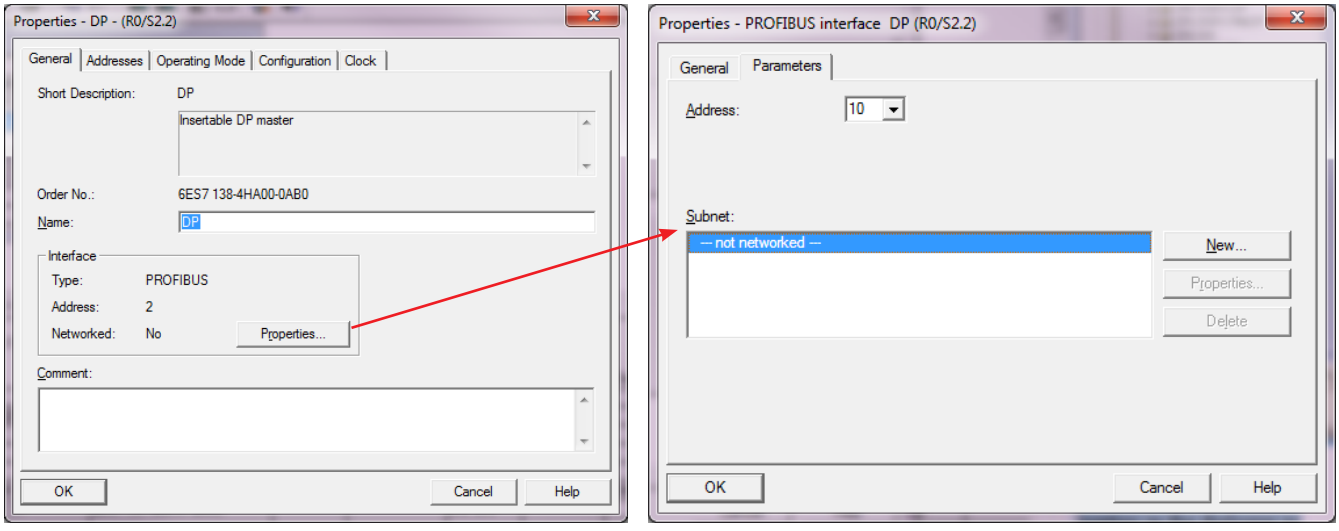

Figure 5. Adding the Hardware to the Configuration

On the PROFIBUS interface properties window (Figure 6), the New button was clicked to define a new network.

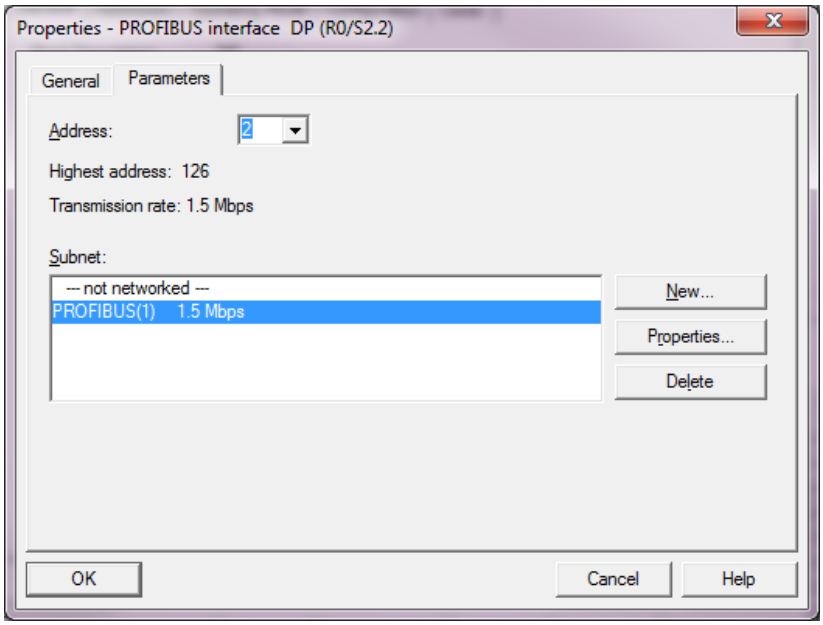

Figure 6. Configuring the Properties of the Profibus Network

The desired settings were made and saved with the OK button. The resulting PLC hardware configuration is shown in Figure 7.

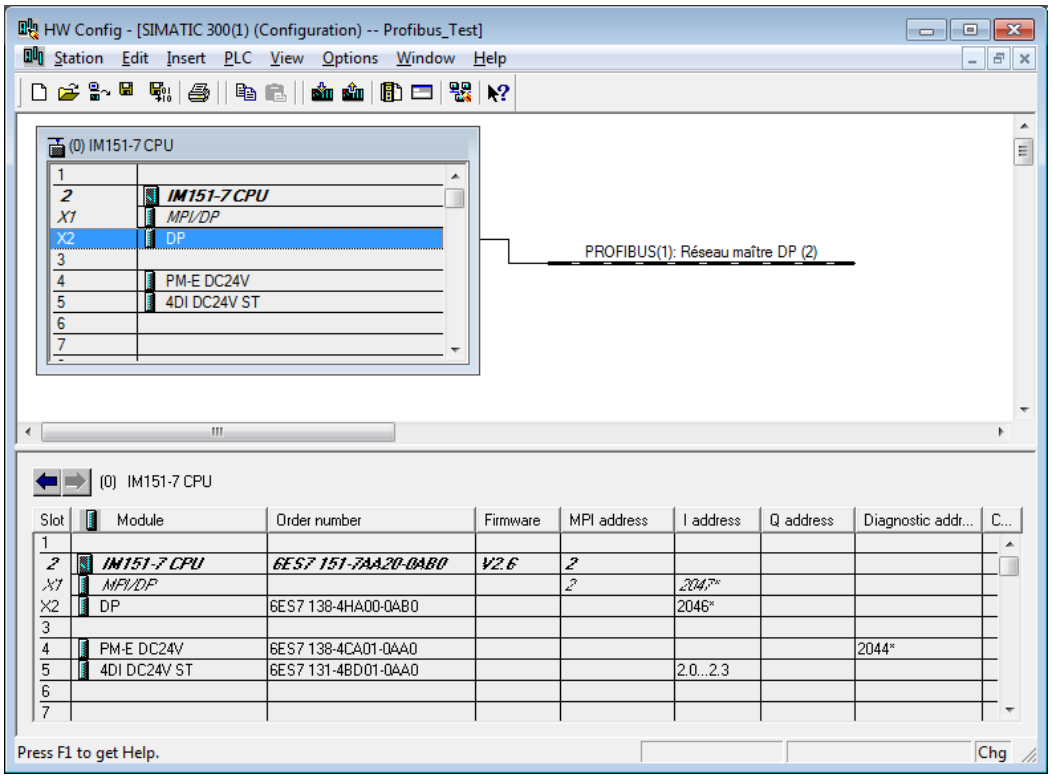

Figure 7. PLC with Empty Bus

Profibus Configuration

PROFIBUS configuration tasks consisted of importing the DECS-250 GSD file, configuring the DECS-250 module, establishing the I/O configuration, and configuring the data organization.

Importing the GSD File

It was necessary to import the DECS-250 GSD file into the configuration tool in order to include the DECS-250 module. Figure 8 illustrates the menu selections. In this case, a DECS-250 Slave module was added as a slave in the network. A DECS-250 GSD file may be requested from Basler Electric via email at info@basler.com. Because the SIMATIC software does not accommodate file names longer than 12 characters, the GSD file may need to be renamed.

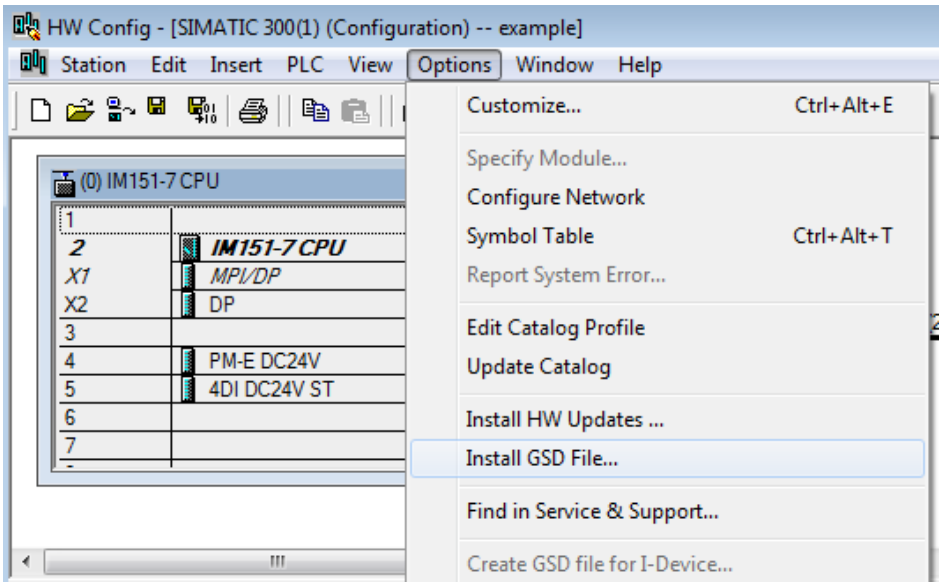

Figure 8. Installing a New GSD File in "HW Config"

Configuring the DECS-250 Module

As shown in Figure 9, the DECS-250 module was shown in the hardware catalogue after the GSD file was imported. This made it possible to include the DECS-250 slave in the network.

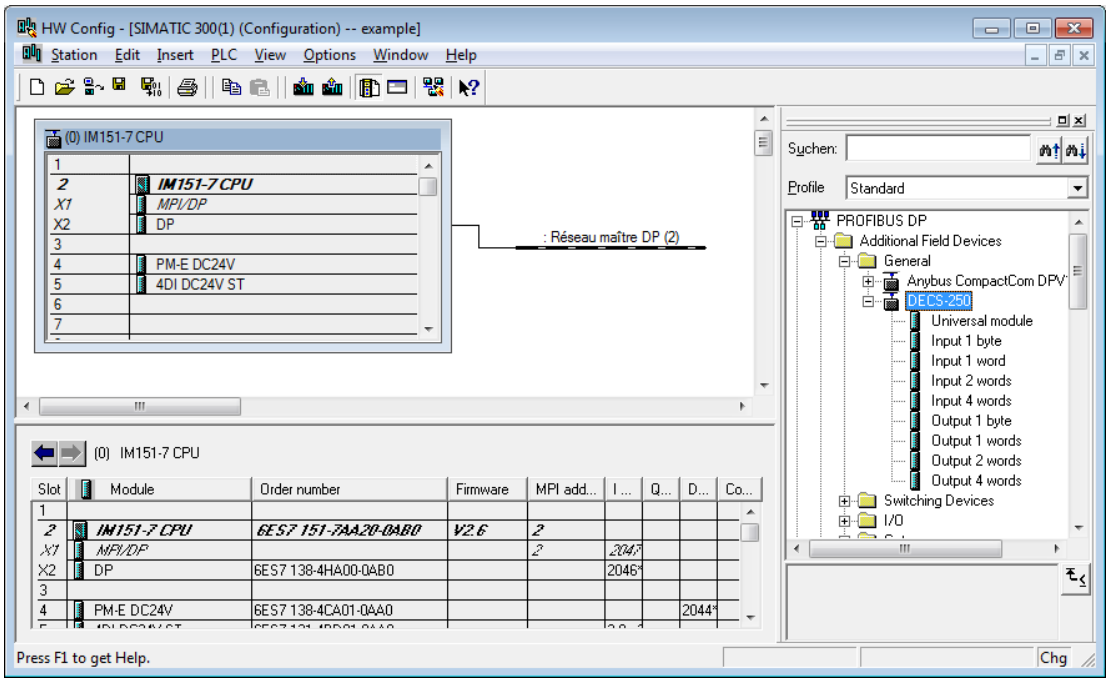

Figure 9. Installing a New GSD File in "HW Config"

The “DECS-250” entry in the hardware catalogue was opened and, as shown in Figure 10, the DECS-250 module was dragged-and-dropped from the catalogue to the network.

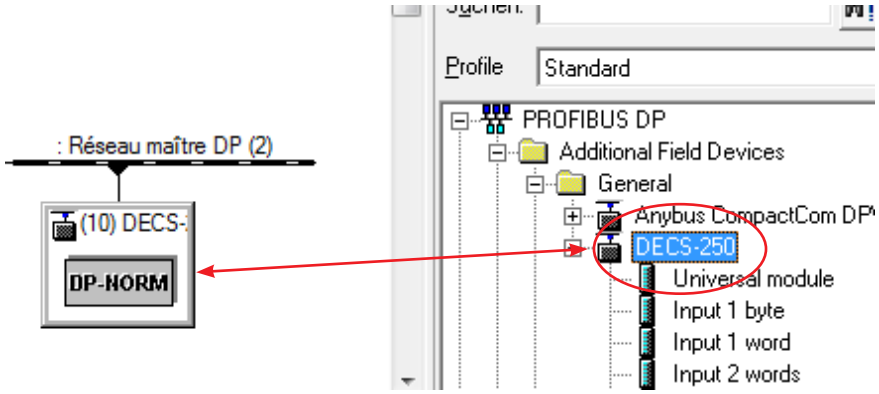

Figure 10. Dragging-and-Dropping the DECS-250 Module to the Network

Then, it was necessary to configure the DECS-250 module itself. This task consisted of node address setup, configuration of the input and output data area sizes, and assignment of the offset address.

Figure 11 shows the mouse-click sequence used to open the node address edit window and set the desired node address. Mouse-click sequence ① and ② opened the edit window and sequence ③ accessed the address setting. The address was set to equal the address of the DECS-250 module.

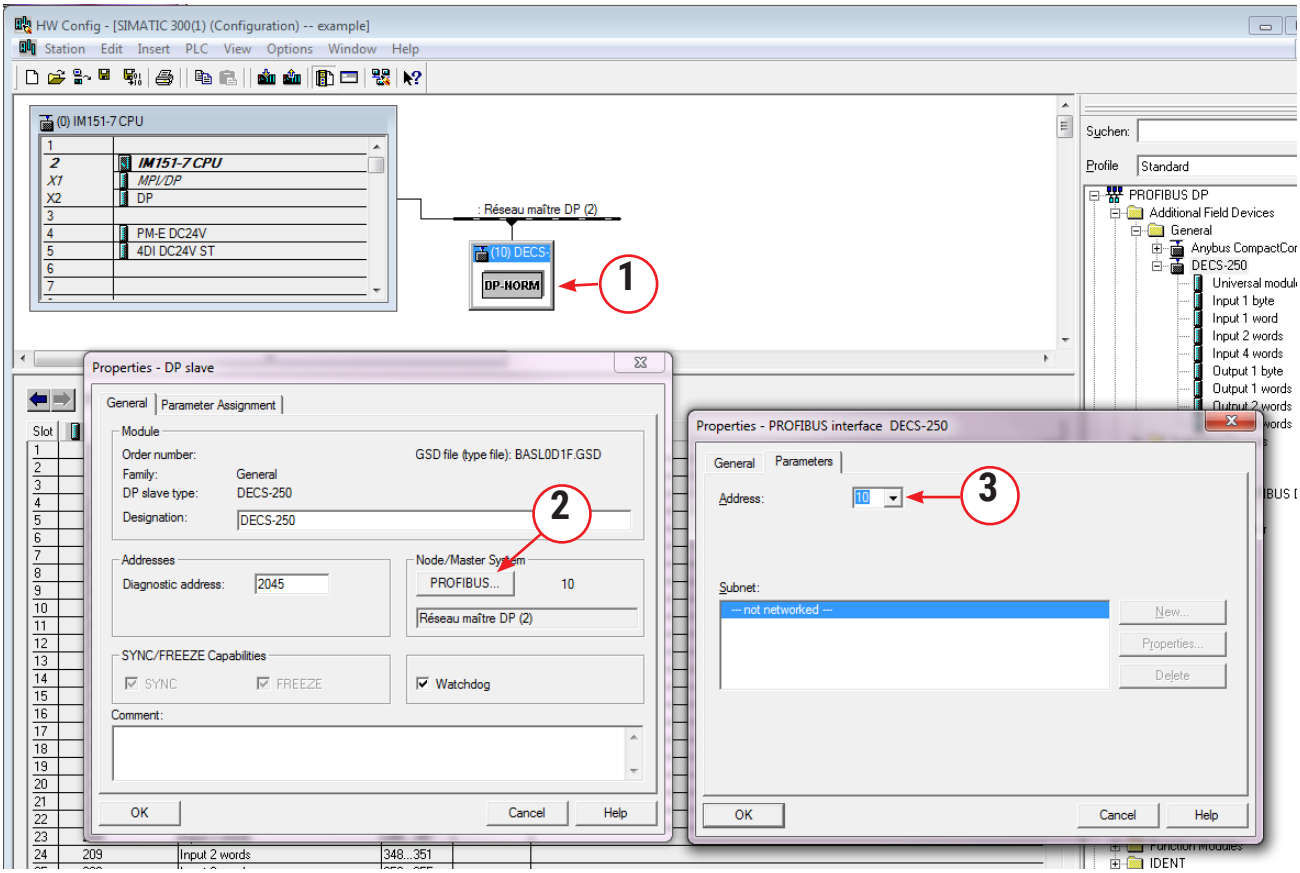

Figure 11. Adjusting the Node Address in "HW Config"

I/O Configuration (Cyclic Parameters)

All cyclic parameters were set in the SIMATIC hardware configuration tool as shown in Figure 12. Communication will not start unless the table is completed. DECS-250 PROFIBUS parameters are listed in the "PROFIBUS Communication" chapter of the DECS-250 instruction manual.

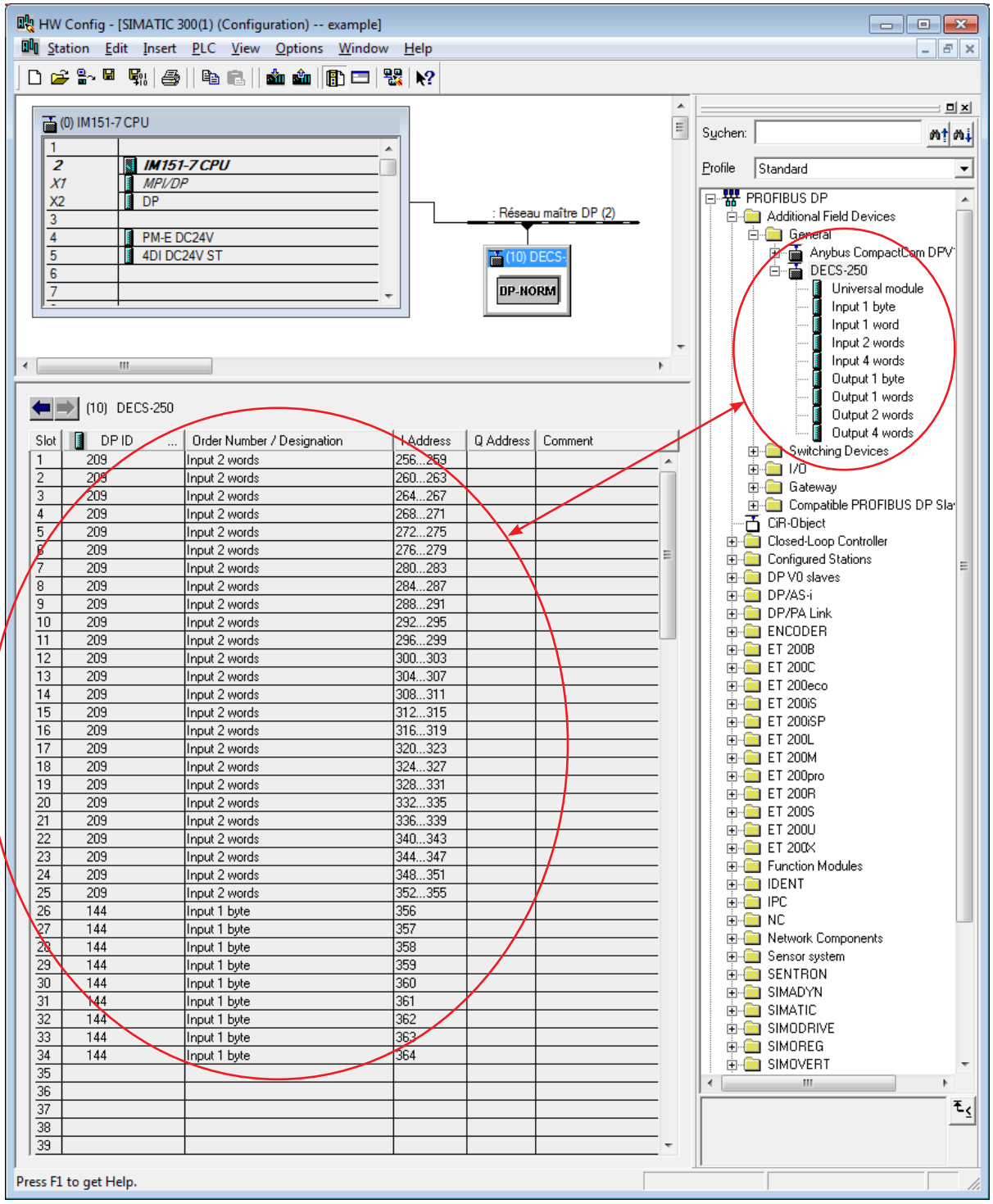

Figure 12. Selecting Cyclic Input Modules with "Drag and Drop"

Data organization

Configuration of data organization consisted of establishing the data float type and configuring the cyclic data packets.

Float Type

Float type is composed of two words (or four bytes). From the *PROFIBUS Parameters* table in the DECS-250 instruction manual, we can see that the table includes 25 consecutive float parameters.

In Figure 12 above, the "address" column provides the means to point to the parameters we later want to address.

Float Example

In a function block, the following LIST instruction was used to read the Generator Voltage Setpoint:

```
L   PED 324
```

From Figure 12, address 324 corresponds to the 18th float type parameter which corresponds to the 18th float parameter in the PROFIBUS Parameters table in the DECS-250 instruction manual.

UINT8

From Figure 12, the start address for UINT8 data storage was set to 356. This byte corresponds to the first UINT8 parameter in Instance #6 of the *PROFIBUS Parameters* table in the DECS-250 instruction manual.

Items with the type UINT8 listed in the *PROFIBUS Parameters* table are all bit-packed binary data. This allows us to transmit up to eight single-bit parameters in each byte of data. When configuring an instance of UINT8-type objects, the data type is "Input 1 byte." The size is determined by the number of objects in the instance divided by eight and rounded up to the next integer. Table 1 shows the sizes for the cyclic data packets of UINT8 type.

Table 1. Cyclic Data Packet Sizes, UINT8 Data Type

Instance	Parameters	Parameters ÷ 8	Total Data Size
6	5	0.625	1 byte
7	7	0.875	1 byte
8	5	0.625	1 byte
9	6	0.75	1 byte
10	16	2	2 bytes
11	12	1.5	2 bytes
12	8	1	1 byte

Within these packets the data is packed in the order listed, with the first item being in the lowest bit of the first byte. If there are unused bits, they are filled with the value zero. The following examples show the bit pack order for instances 8 (Controller Status Cyclic) and 11 (Local Contact Outputs Cyclic).

For more information, refer to the "PROFIBUS Communication" chapter in the DECS-250 manual.

UINT8 - Example 1

Instance 8:

Controller Statys Cyclic	8	UINT8	R	DECSCONTROL_IN_AVR_MODE
Controller Statys Cyclic	8	UINT8	R	DECSCONTROL_IN_FCR_MODE
Controller Statys Cyclic	8	UINT8	R	DECSCONTROL_IN_FVR_MODE
Controller Statys Cyclic	8	UINT8	R	DECSCONTROL_IN_PF_MODE
Controller Statys Cyclic	8	UINT8	R	DECSCONTROL_IN_VAR_MODE

Instance Number	Bit #							
	7	6	5	4	3	2	1	0
8	0	0	0	Var	PF	FVR	FCR	AVR

0 - Unused Position

Reading a value of 0x02 (0000 0010) for instance 8 would indicate that the device was operating in FCR mode.

UINT - Example 2

Instance 11:

Local Contact Outputs Cyclic	11	UINT8	R	CONTACTOUTPUTS_ WATCHDOGOUTPUT
Local Contact Outputs Cyclic	11	UINT8	R	CONTACTOUTPUTS_OUTPUT1
Local Contact Outputs Cyclic	11	UINT8	R	CONTACTOUTPUTS_OUTPUT2
Local Contact Outputs Cyclic	11	UINT8	R	CONTACTOUTPUTS_OUTPUT3
Local Contact Outputs Cyclic	11	UINT8	R	CONTACTOUTPUTS_OUTPUT4
Local Contact Outputs Cyclic	11	UINT8	R	CONTACTOUTPUTS_OUTPUT5
Local Contact Outputs Cyclic	11	UINT8	R	CONTACTOUTPUTS_OUTPUT6
Local Contact Outputs Cyclic	11	UINT8	R	CONTACTOUTPUTS_OUTPUT7
Local Contact Outputs Cyclic	11	UINT8	R	CONTACTOUTPUTS_OUTPUT8
Local Contact Outputs Cyclic	11	UINT8	R	CONTACTOUTPUTS_OUTPUT9
Local Contact Outputs Cyclic	11	UINT8	R	CONTACTOUTPUTS_OUTPUT10
Local Contact Outputs Cyclic	11	UINT8	R	CONTACTOUTPUTS_OUTPUT11

Instance Number	Bit #								
	7	6	5	4	3	2	1	0	
11	OUT7	OUT6	OUT5	OUT4	OUT3	OUT2	OUT1	WD	First Byte
	0	0	0	0	OUT11	OUT10	OUT9	OUT8	Second Byte

0 - Unused Position

A value of 0xA4 06 (1010 0100 0000 0110) for instance 11 indicates that outputs 2, 5, 7, 9 and 10 are closed.

Acyclic Data manipulation examples

Acyclic Read

Acyclic Read was performed with SFB 52 "RDREC". The following excerpt from Siemens publication A5E02789976-01 provides information about the SFB 52 "RDREC" interface.

Note: The SFB52 "RDREC" interface is identical to the FB "RDREC" defined in the standard "PROFIBUS Guideline PROFIBUS Communication and Proxy Function Blocks according to IEC 61131-3".

Description:

With the SFB52 "RDREC" (read record) you read a data record with the number INDEX from a component. This may be a module in a central rack or a distributed component (PROFIBUS DP or PROFINET IO).

Specify the maximum number of bytes you want to read in MLEN. The selected length of the target area RECORD should have at least the length of MLEN bytes.

TRUE on output parameter VALID verifies that the data record has been successfully transferred into the target area RECORD. In this case, the output parameter LEN contains the length of the fetched data in bytes.

The output parameter ERROR indicates whether a data record transmission error has occurred. In this case, the output parameter STATUS contains the error information.

Operating Principle:

The SFC 52 "RDREC" operates asynchronously, that is, processing covers multiple SFC calls. Start the job by calling SFC52 with REQ = 1.

The job status is displayed via the output parameter BUSY and the two middle bytes of output parameter STATUS. Here, the two middle bytes of STATUS correspond to the output parameter RET_VAL of the asynchronous SFCs (see also Meaning of REQ, RET_VAL and BUSY with Asynchronously Operating SFCs).

Data record transmission is completed when the output parameter BUSY = FALSE.

Parameters	Declaration	Data Type	Memory Area	Description
REQ	INPUT	BOOL	I, Q, M, D, L, constant	REQ = 1: Transfer data record
ID	INPUT	DWORD	I, Q, M, D, L, constant	Diagnostic Address
INDEX	INPUT	INT	I, Q, M, D, L, constant	Data Record Number
MLEN	INPUT	INT	I, Q, M, D, L, constant	Maximum length in bytes of the data record information to be fetched
VALID	OUTPUT	BOOL	I, Q, M, D, L	New data record was received and valid
BUSY	OUTPUT	BOOL	I, Q, M, D, L	BUSY = 1: The read process is not yet terminated
ERROR	OUTPUT	BOOL	I, Q, M, D, L	ERROR = 1: A read error has occurred
STATUS	OUTPUT	DWORD	I, Q, M, D, L	Block status or error information
LEN	OUTPUT	INT	I, Q, M, D, L	Length of the fetched data record information
RECORD	OUTPUT	ANY	I, Q, M, D, L	Destination area for the read data record Note: The RECORD parameter of S7-300 CPUs always requires the full specification of the DB parameters (for example: P#DB13.DBX0.0 byte 100). Omitting an explicit DB number is not permitted for S7-300 CPUs and causes an error message in the user program.

Error Information: for an interpretation of the STATUS parameter, see Receiving an Interrupt with SFB54 "RALRM"

Important: Partial Acyclic Read may be possible by specifying an MLEN smaller than the instance parameter count.

Acyclic FLOAT Read example
The following instructions were used to read Float Inst#29 (Control Panel Setpoint Metering):

```
// Request triggered only if communication is not busy
U M 100.0
UN DB52.DBX 10.1   // DB52.BUSY
= DB52.DBX 0.0     // DB52.REQ

// ID
L 2045             // Diagnostic Address of the Profibus PN Slave
T DB52.DBD 2       // DB52.ID

// Index
L 28               // Data record (DECS-250 INST# - 1)
T DB52.DBW 6       // DB52.INDEX

// MLEN
L 5                // Key count to Read
L 4                // Byte per Key (for Float type)
*D
T DB52.DBW 8       // DB52.MLEN

// Call the SFB52 with instance DB52
CALL "RDREC" , DB52       -- Read a Process Data Record
REQ  :=DB52.DBX0.0
ID   :=DB52.DBD2
INDEX       :=DB52.DBW6
MLEN        :=DB52.DBW8
VALID :=DB52.DBX10.0
BUSY        :=DB52.DBX10.1
ERROR :=DB52.DBX10.2
STATUS:=DB52.DBD12
LEN  :=DB52.DBW16
RECORD:=P#DB201.DBX0.0 BYTE 100        // DB201, start from Byte:0, length:100 Bytes

// Error count
UN DB52.DBX 10.2          // DB52.ERROR
SPB noer
L #cptError
INC 1
T #cptError

// Valid count
noer: UN DB52.DBX 10.0    // DB52.VALID
SPB nova
L #cptValid
INC 1
T #cptValid

// End
nova: NOP 0
L #cptValid
L #cptError
```

Acyclic UINT8 Read example
The following instructions were used to read Float Inst#40 (Contact Output Status):

```
// Request triggered only if communication is not busy
U M 100.0
UN DB52.DBX 10.1   // DB52.BUSY
= DB52.DBX 0.0     // DB52.REQ

// ID
L 2045             // Diagnostic Address of the Profibus PN Slave
T DB52.DBD 2       // DB52.ID

// Index
L 39               // Data record (DECS-250 INST# - 1)
T DB52.DBW 6       // DB52.INDEX

// MLEN
L 12               // Key count to Read
L 8                // Bit per Byte (don't change this, it's always 8)
/R
RND+               // Round up to the nearest Integer
T DB52.DBW 8       // DB52.MLEN

// Call the SFB52 with instance DB52
CALL "RDREC" , DB52 SFB52 -- Read a Process Data Record
REQ  :=DB52.DBX0.0
ID   :=DB52.DBD2
INDEX :=DB52.DBW6
MLEN        :=DB52.DBW8
VALID :=DB52.DBX10.0
BUSY        :=DB52.DBX10.1
ERROR :=DB52.DBX10.2
STATUS:=DB52.DBD12
LEN  :=DB52.DBW16
RECORD:=P#DB202.DBX0.0 BYTE 100       // DB202, start from Byte:0, length:100 Bytes

// Error count
UN DB52.DBX 10.2         // DB52.ERROR
SPB noer
L #cptError
INC 1
T #cptError

// Valid count
noer: UN DB52.DBX 10.0 // DB52.VALID
SPB nova
L #cptValid
INC 1
T #cptValid

// End
nova: NOP 0
L #cptValid
L #cptError
```

Acyclic UINT32 Read example
The following instructions were used to read UINT32 Inst#46 (Front Panel Settings):

```
// Request triggered only if communication is not busy
U M 100.0
UN DB52.DBX 10.1   // DB52.BUSY
= DB52.DBX 0.0     // DB52.REQ

// ID
L 2045             // Diagnostic Adresse of the Profibus PN Slave
T DB52.DBD 2       // DB52.ID

// Index
L 45                     // Data record number (DECS-250 INST# - 1)
T DB52.DBW 6       // DB52.INDEX

// MLEN
L 7                // Key Number to Read
L 4                // Byte per Key
*D
T DB52.DBW 8       // DB52.MLEN

// Call the SFB52 with instance DB52
CALL "RDREC" , DB52 SFB52 - Read a Process Data Record
REQ  :=DB52.DBX0.0
ID   :=DB52.DBD2
INDEX :=DB52.DBW6
MLEN        :=DB52.DBW8
VALID :=DB52.DBX10.0
BUSY        :=DB52.DBX10.1
ERROR :=DB52.DBX10.2
STATUS:=DB52.DBD12
LEN  :=DB52.DBW16
RECORD:=P#DB203.DBX0.0 BYTE 100       // DB203, start from Byte:0, length:100 Bytes

// Error count
UN DB52.DBX 10.2         // DB52.ERROR
SPB noer
L #cptError
INC 1
T #cptError

// Valid count
noer: UN DB52.DBX 10.0 // DB52.VALID
SPB nova
L #cptValid
INC 1
T #cptValid

// End
nova: NOP 0
L #cptValid
L #cptError
```

Acyclic STRING Read example
The following instructions were used to read STRING Inst#45 (Real-Time Clock):

```
// Request triggered only if communication is not busy
U M 100.0
UN DB52.DBX 10.1          // DB52.BUSY
= DB52.DBX 0.0            // DB52.REQ

// ID
L 2045                    // Diagnostic Adresse of the Profibus PN Slave
T DB52.DBD 2              // DB52.ID

// Index
L 44                      // Data record (DECS-250 INST# - 1)
T DB52.DBW 6              // DB52.INDEX

// MLEN
L 50                      // Key count to Read
L 1                       // Byte per Key
*D
T DB52.DBW 8              // DB52.MLEN

// Call the SFB52 with instance DB52
CALL "RDREC" , DB52 SFB52 - Read a Process Data Record
REQ  :=DB52.DBX0.0
ID   :=DB52.DBD2
INDEX :=DB52.DBW6
MLEN        :=DB52.DBW8
VALID :=DB52.DBX10.0
BUSY        :=DB52.DBX10.1
ERROR :=DB52.DBX10.2
STATUS:=DB52.DBD12
LEN  :=DB52.DBW16
RECORD:=P#DB204.DBX0.0 BYTE 100 // DB204, start from Byte:0, length:100 Bytes

// Error count
UN DB52.DBX 10.2          // DB52.ERROR
SPB noer
L #cptError
INC 1
T #cptError

// Valid count
noer: UN DB52.DBX 10.0 // DB52.VALID
SPB nova
L #cptValid
INC 1
T #cptValid

// End
nova: NOP 0
L #cptValid
L #cptError
```

Acyclic Write

Acyclic Write was performed with SFB 53 "WDREC". The following excerpt from Siemens publication A5E02789976-01 provides information about the SFB 53 "WRRREC" interface.

Note: The SFB53 "WRREC" interface is identical to the FB "WRREC" defined in the standard "PROFIBUS Guideline PROFIBUS Communication and Proxy Function Blocks according to IEC 61131-3".

Description:

With the SFB53 "WRREC" (write record) you transfer a data record with the number INDEX to a DP slave/PROFINET IO device component defined by ID. This might be a module in the central rack or a distribute component (PROFIBUS DP or PROFINET IO).

Specify the byte length of the data record to be transmitted. Therefore, selected length of the source area RECORD should have at least the length of LEN bytes.

TRUE on output parameter DONE verifies that the data record has been successfully transferred to the DP Slave.

The output parameter ERROR indicates whether a data record transmission error has occurred. In this case, the output parameter STATUS contains the error information.

Operating Principle:

The SFC 53 "WRREC" operates asynchronously, that is, processing covers multiple SFC calls. Start the job by calling SFC53 with REQ = 1.

The job status is displayed via the output parameter BUSY and the middle two bytes of output parameter STATUS. Here, the middle two bytes of STATUS correspond to the output parameter RET_VAL of the asynchronously operating SFCs (see also Meaning of REQ, RET_VAL and BUSY with Asynchronously Operating SFCs).

Please note that you must assign the same value to the actual parameter of RECORD for all SFB53 calls that belong to one and the same job. The same applies to the LEN parameters.

Data record transmission is completed when the output parameter BUSY = FALSE.

Parameters	Declaration	Data Type	Memory Area	Description
REQ	INPUT	BOOL	I, Q, M, D, L, constant	REQ = 1: Transfer data record
ID	INPUT	DWORD	I, Q, M, D, L, constant	Diagnostic Address
INDEX	INPUT	INT	I, Q, M, D, L, constant	Data Record Number
LEN	INPUT	INT	I, Q, M, D, L, constant	Maximum length in bytes of the data record information to be fetched
DONE	OUTPUT	BOOL	I, Q, M, D, L	Data record was transferred
BUSY	OUTPUT	BOOL	I, Q, M, D, L	BUSY = 1: The read process is not yet terminated
ERROR	OUTPUT	BOOL	I, Q, M, D, L	ERROR = 1: A read error has occurred
STATUS	OUTPUT	DWORD	I, Q, M, D, L	Block status or error information To interpret the STATUS parameter, refer to Receiving an Interrupt with SFB 54 "RALRM"
RECORD	OUTPUT	ANY	I, Q, M, D, L	Data record Note: The RECORD parameter of S7-300 CPUs always requires the full specification of the DB parameters (for example: P#DB13.DBX0.0 byte 100). Omitting an explicit DB number is not permitted for S7-300 CPUs and causes an error message in the user program.

Error Information: for an interpretation of the STATUS parameter, see Receiving an Interrupt with SFB54 "RALRM"

Important: Partial Acyclic Write is not possible by specifying an LEN smaller than the instance parameter count! To modify a parameter, the complete instance was read, the desired parameter was updated, and finally the complete instance was written back.

Acyclic FLOAT Write example
The following instructions were used to write FLOAT Inst#66 (System Param):

```
// Request triggered only if communication is not busy
U M 100.0
UN DB53.DBX 10.1   // DB53.BUSY
= DB53.DBX 0.0            // DB53.REQ

// ID
L 2045                    // Diagnostic Adresse of the Profibus PN Slave
T DB53.DBD 2              // DB53.ID

// Index
L 65                      // Data record (DECS-250 INST# - 1)
T DB53.DBW 6              // DB53.INDEX

// MLEN
L 9                       // Key count to Write
L 4                       // Byte per Key (for Float type)
*D
T DB53.DBW 8              // DB53.LEN

CALL "WRREC" , DB53 SFB53 -- Write a Process Data Record
REQ  :=DB53.DBX0.0
ID   :=DB53.DBD2
INDEX :=DB53.DBW6
LEN  :=DB53.DBW8
DONE        :=DB53.DBX10.0
BUSY        :=DB53.DBX10.1
ERROR :=DB53.DBX10.2
STATUS:=DB53.DBD12
RECORD:=P#DB201.DBX0.0 BYTE 100 // DB201, start from Byte:0, length:100 Bytes

// Error count
UN DB53.DBX 10.2          // DB52.ERROR
SPB noer
L #cptError
INC 1
T #cptError

// Valid count
noer: UN DB53.DBX 10.0 // DB52.VALID
SPB nova
L #cptValid
INC 1
T #cptValid

// End
nova: NOP 0
L #cptValid
L #cptError
```

Basler Electric®

Highland, Illinois USA
Tel +1 618.654.2341
Fax +1 618.654.2351
email: info@basler.com

Suzhou, P.R. China
Tel +86 512.8227.2888
Fax +86 512.8227.2887
email: chinainfo@basler.com