EUCHNER

Application

Example integration of EKS Data Service

EN

Contents

EN

1. About this document

1.1. Version

Version	Date	Change/addition	Chapter
01-10/19	10/22/2019	Prepared	All

1.2. Scope

The purpose of this document is to aid the integration of the EKS Data Service in combination with the EKS with PROFINET interface (from version as per table) in SIEMENS TIA Portal from version V14 SP1.

Order no.	Designation	Version
163316	EKS Data Service	V1.0.4
106305	EKS-A-IIX-G01-ST02/03	V2.9.X
106306	EKS-A-IIXA-G01-ST02/03/04	V2.9.X
122352	EKS-A-AIX-G18	V1.X.X
122353	EKS-A-AIXA-G18	V1.X.X

1.3. Target group

Design engineers, installation planners as well as setup and servicing staff possessing expertise in the installation, setup, programming and diagnostics of programmable logic controllers (PLC) and bus systems.

1.4. Supplementary documents

The overall documentation for this application consists of the following documents:

Document title (document number)	Contents	
Manual (2528408)	EKS Data Service PC	
Manual (2528410)	EKS Data Service PLC	
Manual (2516210)	Electronic-Key-System Manual EKS and EKS *FSA* with PROFINET IO interface	
Manual (093336)	Electronic-Key-Manager EKM	
Application (AP000238)	Integration of EKS with PROFINET interface in TIA Portal V13/V14/V15	
Possibly enclosed data sheets	Item-specific information about deviations or additions	

1.5. Notice

This application is based on the manual for the EKS with PROFINET interface, the manuals EKS Data Service PC and PLC as well as the application AP000238 "Integration of EKS with PROFINET interface in TIA Portal V13/V14/V15". Please refer to the manuals for technical details and other information. In the rest of this document the EKS with PROFINET interface is referred to as the "EKS" and the EKS Data Service as the "service" for short.

2. Components/software used

2.1. EUCHNER components

Description	Order number / item
EKS with PROFINET interface	106305 / EKS-A-IIX-G01-ST02/03
EKS with PROFINET interface *FSA*	106306 / EKS-A-IIXA-G01-ST02/03/04

TIP!

More information and downloads about the aforementioned EUCHNER products can be found at www.euchner.com. Simply enter the order number in the search box.

2.2. Other components

Description	Order number / item
SIMATIC S7-1516F-3 PN/DP	6ES7516-3FN00-0AB0
SIMATIC S7-315F-2 PN/DP	6ES7315-2FH13-0AB0
Communication processor CP 343-1	6GK7343-1EX30-0XE0
Desktop PC	-

2.3. Software

Description	Version
Totally Integrated Automation Portal	Version V14 SP1 update 7
STEP 7 Professional	Version V14 SP1 update 7
STEP 7 Manager	STEP 7 V5.5
Electronic-Key-Manager full version (order no. 093322)	Build 1.5.2.0
EKS Data Service (order no. 163316)	V 1.0.4

3. Functional description

In EKS applications in the PLC world, in most cases these days there is no data comparison with a central database or retrieval of data from a central database. This means that data are almost always read from the Electronic-Key decentrally and then further processed individually in the PLC. In this way the access information is only transported via the Electronic-Key. Here many EKS operators want to compare data from the PLC world with EKM data in the PC world. The core issue is the desire to be able to disable Electronic-Keys centrally and retrieve other data if necessary.

The content of the Electronic-Key-Manager EKM database is exported to a universal file in the CSV format in the PC environment. In this EKM CSV export file, it is possible to evaluate a disable code after the serial number (KeyID) of the Electronic-Key: this code is set to the value '1' as soon as the Electronic-Key is disabled.

Other data elements assigned to the Electronic-Key serial number can also be retrieved. These data elements differ for specific applications.

If an Electronic-Key is placed in the adapter, a query is sent by the EKS Data Service PLC to the EKS Data Service PC. Based on the KeyID, the EKS Data Service PC searches for the entry in the EKM CSV export file and sends the data back to the EKS Data Service PLC. There the data are available to the user for further processing. Furthermore, the data queried are saved in a backup cache. Should the connection to the PC fail, the data for Electronic-Keys placed previously are read from the backup strategy cache. The backup strategy cache contains as a maximum the data on the last 100 Electronic-Keys (size of the backup cache can be customized).

Note on the backup strategy: if the backup strategy cache is full and data queried for a new Electronic-Key, the Electronic-Key for which data has not been queried the longest is replaced. Because the entries for the backup strategy are updated cyclically, the first Electronic-Key placed will not necessarily be the Electronic-Key replaced.

There are 2 backup strategies. In backup strategy 1, the EKS Data Service PC cannot access the original EKM CSV export file and uses the local backup of the EKM CSV export file. In backup strategy 2, the EKS Data Service PC cannot make available any data because the communication is interrupted and the data in the internal PLC data block are used.

EN

4. Electronic-Key-Manager EKM

4.1. EKM server

Start the EKM server and open the settings. On the *Export* tab, select the *Start export after any change* check box. Then specify in *Export-File* where the csv file is to be saved and its name. The service must be able to access the csv file later.

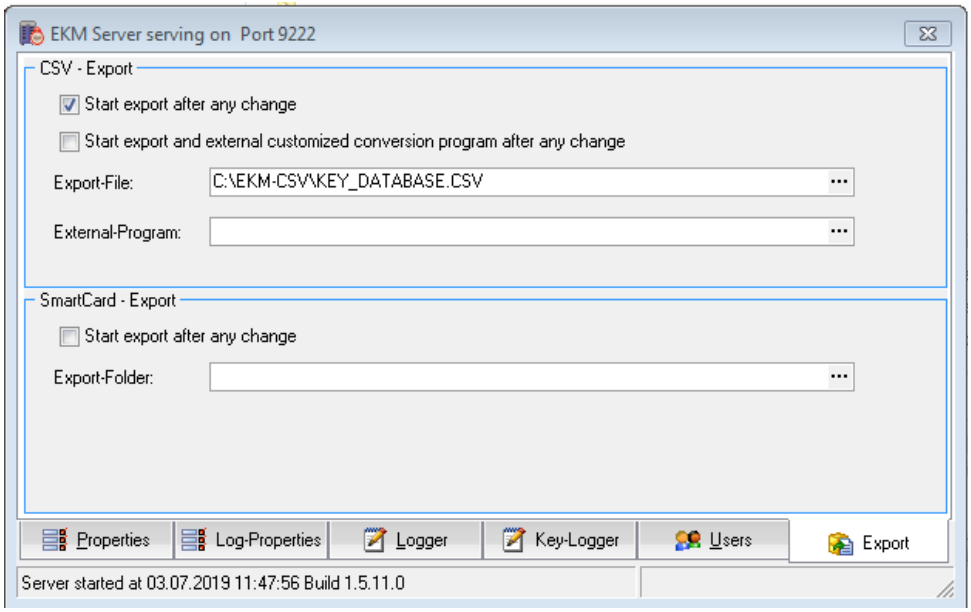


Figure 1: EKM server configuration

4.2. EKM client

A complete EKM application is a prerequisite for the use of the EKS Data Service. This application has a data entry window and a database. The following example EKM data entry window and database are used in the application.

Figure 2: EKM client (example)

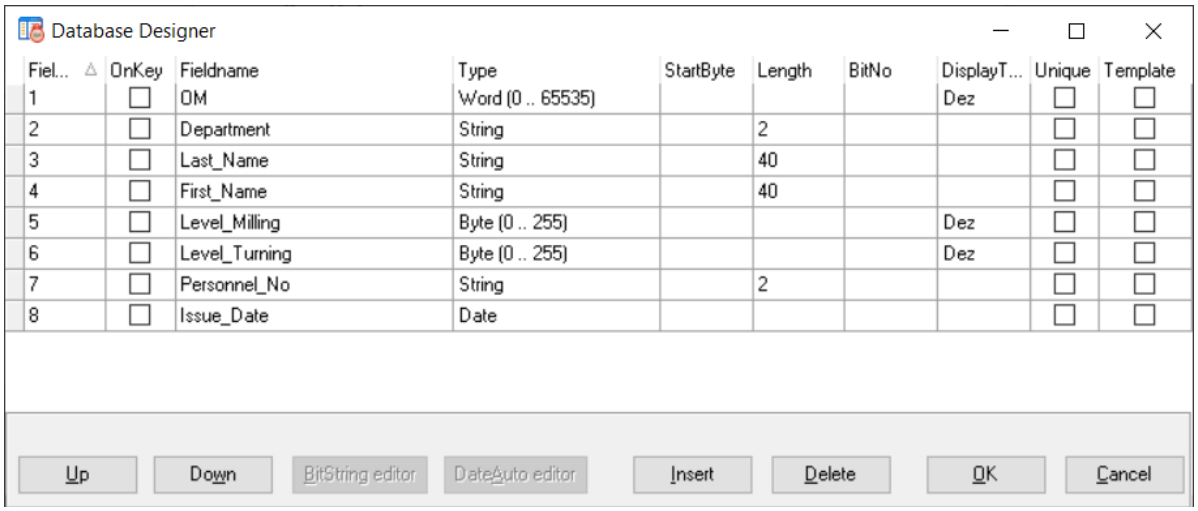


Figure 3: EKM Database Designer (example)

EN

5. EKS Data Service setup

5.1. Installation of the service

Please refer to the EKS Data Service PC manual for the procedure to install the service.

5.2. Configuration

5.2.1. Network configuration

The server on which the service is installed must be connected to the machine network via a network connection.

The service's web interface for configuring the service is opened on the server using the following default address:

http://localhost:65080/EKSDataService

	**NOTICE!** **TCP port 102:** RFC1006 connection of SIMATIC control systems to the service. This port cannot be changed and must therefore also not be reserved by other software products. Siemens software products, e.g. TIA-Portal assign a service to port 102 by default. If parallel installation cannot be avoided, the SIMATIC service S7DOS Help Service (s7oiehsx64.exe) must be stopped and deactivated.

5.2.2. Configuration of the service

You must login to configure the service. The default user has the following login data:

User: *admin*

Password: *admin*

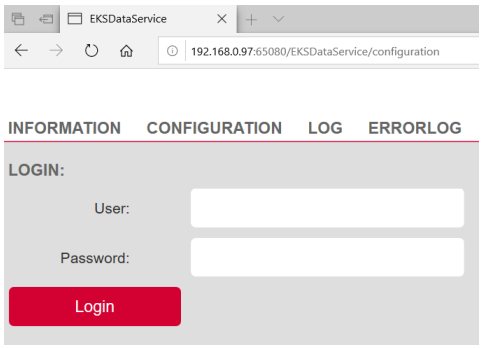


Figure 4: Login

The one-off configuration of the service starts after login. Not all values are configured in the following example. All values that can be configured are explained in the manual and can be set individually.

1. Enter the path and the file name for the csv file. Then click *Read and check now...* . If it is possible to read the csv file, you will receive positive feedback in *File Status*. If negative feedback is output, please check the path information and the file name and check whether the data can be accessed by the server.

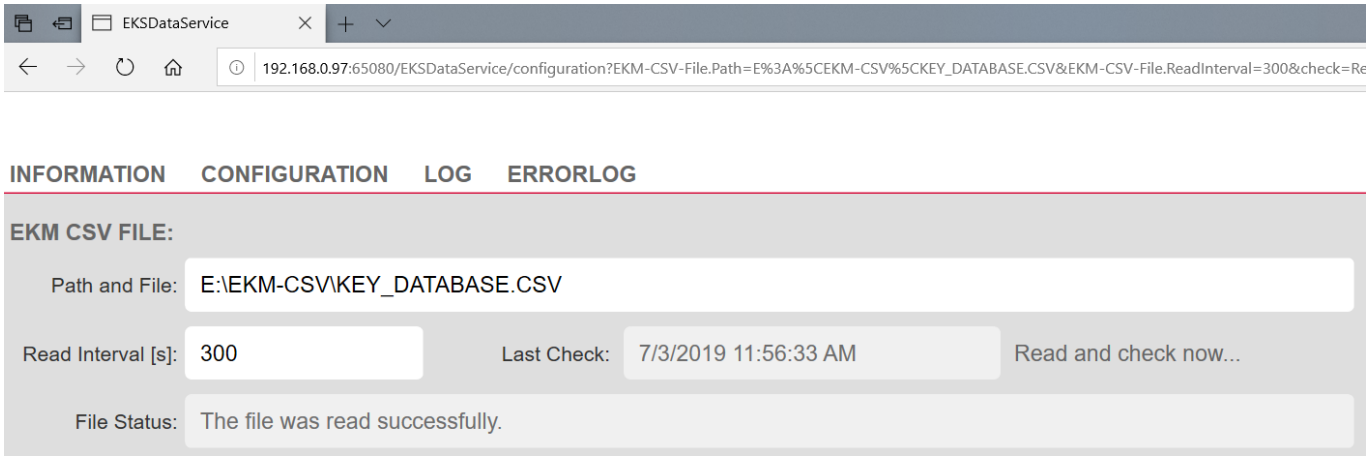


Figure 5: Path information for csv file and reading for the first time

2. After the csv file has been read correctly, the names of the columns from the csv file are entered automatically in *EKM PLC DATA TYPES*. You can specify which data you want to transfer to the PLC. In this example, all data are transferred to the PLC.

   The columns *KEYID* and *LOCKED* are default columns that every EKM database contains. The values are already specified there. The *PLC-Index* is incremented by the value 1 for each column. Then the related data types (*PLC-Type*) are entered. The allocation of the EKM data types to the PLC data types is described in the EKS Data Service PC manual.

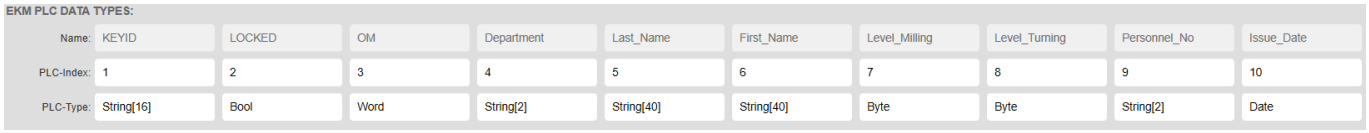


Figure 6: Allocation of data types

3. Save the configuration by clicking *Apply*.

EN

6. Integration of PLC library for S7-1200/1500 into TIA Portal V14 SP1

For the S7-1200/1500 you will need the following libraries:

EKS_Data_Service_PLC_Library_TIA_1200_1500_20190227

Library_EKS_Data_Service-TIAV14SP1_PLC-1200_1500_YYYYMMDD

6.1. Retrieving the library

1. Change to the *Task Card* view (*shortcut: Ctrl+3*) and select *Libraries*.
2. Open the context menu by clicking the *Global libraries* area using the right mouse button and select *Retrieve library…* Select the folder with the library and retrieve it to the required destination folder. The library is updated while retrieving the library using TIA Portal V15 or later because it was prepared using TIA Portal V14 SP1.

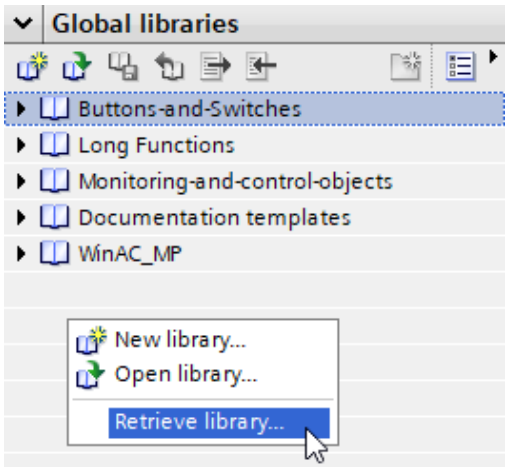


Figure 7: Retrieving library

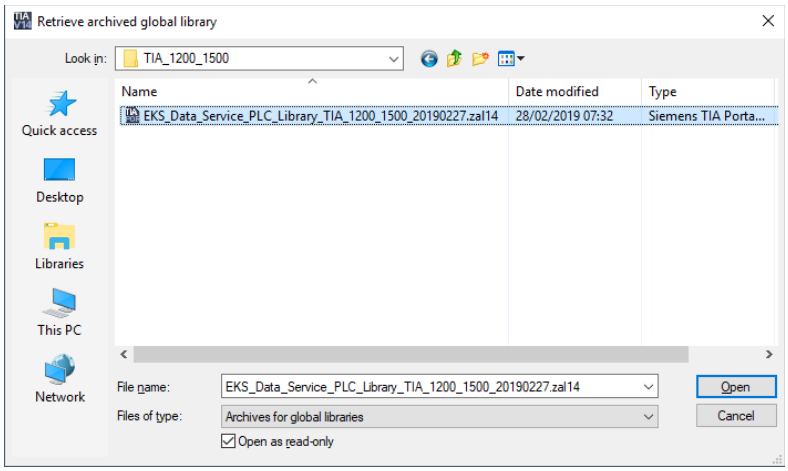


Figure 8: Selecting library

TIP!

If the library has already been retrieved, select *Open library…* to add the library to the project.

6.2. Integration of the EKS Data Service into the PLC

6.2.1. Copying blocks to the project

1. Open the *library* and copy the blocks to the folders in the *project navigation* to suit your control system environment.

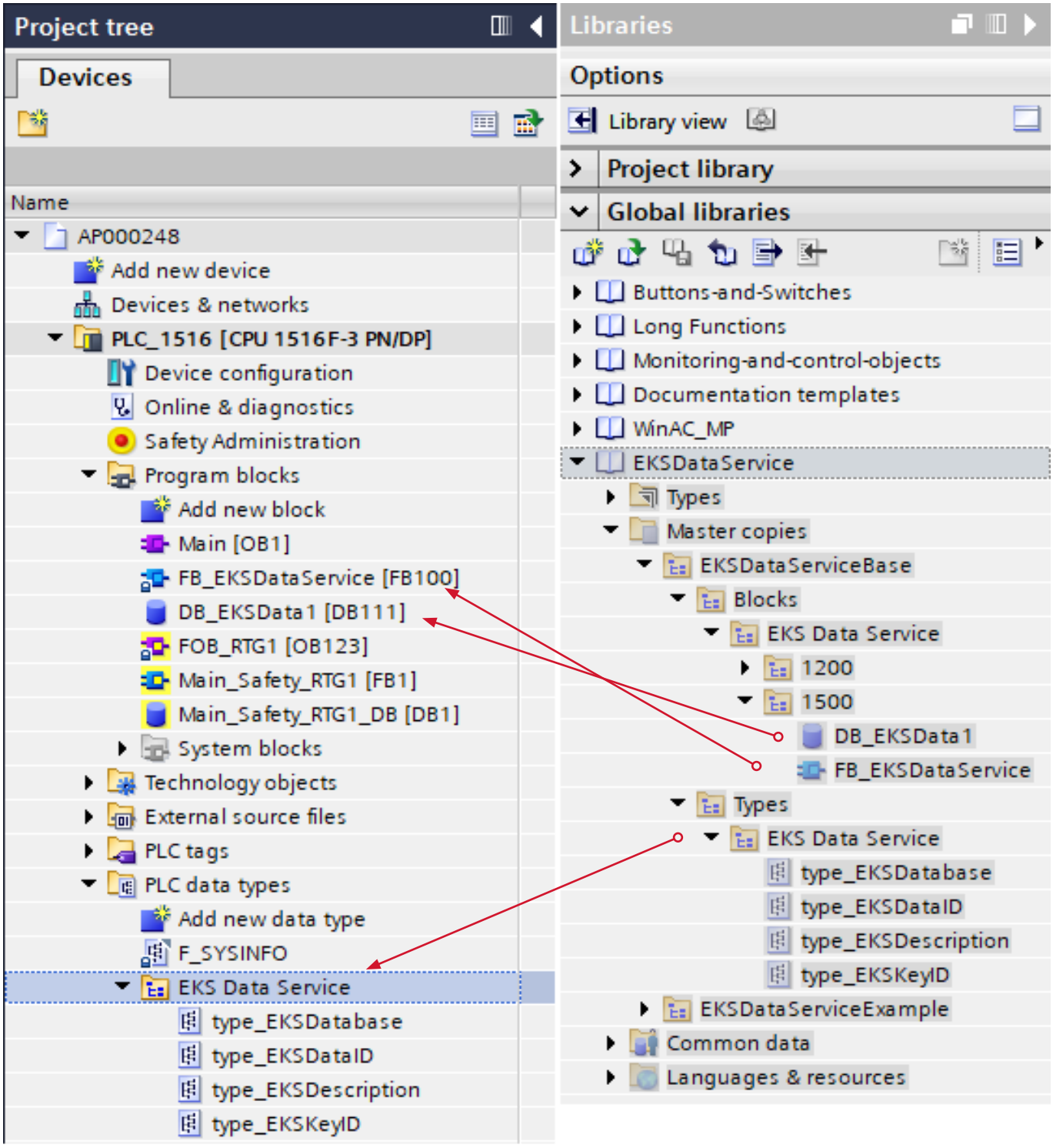


Figure 9: Copying blocks to the project

EN

2. Open the library Library_EKS_Data_Service-TIAV14SP1_PLC-1200_1500_YYYYMMDD and copy the blocks to the folders in the *project navigation*. This library is used to read the EKS Electronic-Key. It is not imperative to use this library.

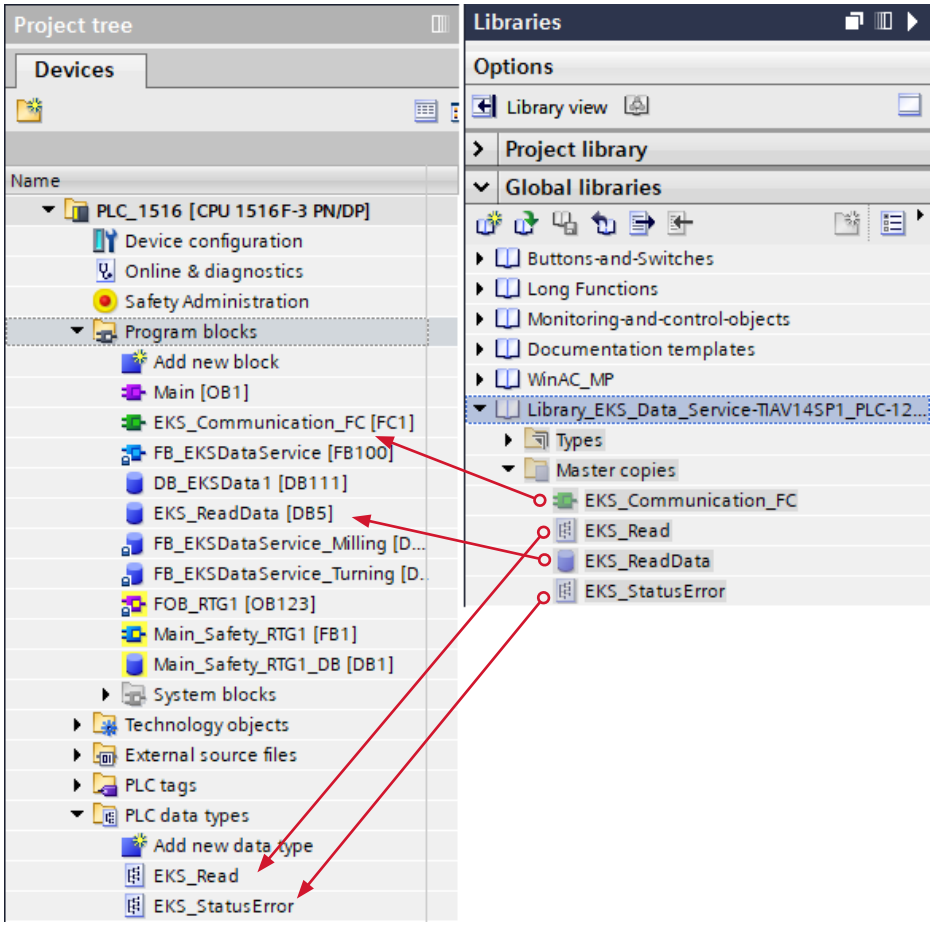


Figure 10: Copying blocks to the project

6.2.2. Customizing the data type *type_EKSDescription*

Open the data type *type_EKSDescription*. Map the *EKM PLC DATA TYPES Name* and *PLC-Index* configured in the service.

type_EKSDescription

	Name	Data type	Default value	Accessible f...	Writa...	Visible in ...	Setpoint
1	KeyID	Int	1	☐	☐	☐	☐
2	LOCKED	Int	2	☐	☐	☐	☐
3	OM	Int	3	☐	☐	☐	☐
4	Department	Int	4	☐	☐	☐	☐
5	Last_Name	Int	5	☐	☐	☐	☐
6	First_Name	Int	6	☐	☐	☐	☐
7	Level_Milling	Int	7	☐	☐	☐	☐
8	Level_Turning	Int	8	☐	☐	☐	☐
9	Personnel_No	Int	9	☐	☐	☐	☐
10	Date	Int	10	☐	☐	☐	☐

Figure 11: Customizing the data type *type_EKSDescription*

6.2.3. Customizing the data type *type_EKSDatabase*

Open the data type *type_EKSDatabase*. Map the *EKM PLC DATA TYPES Name* and *PLC-Type* configured in the service. Pay attention to the allocation of the EKM data types and PLC data types as described in the EKS Data Service PC manual.

type_EKSDatabase

	Name	Data type	Default value	Accessible f...	Writa...	Visible in ...	Setpoint
1	KeyID	String[16]	''	☐	☐	☐	☐
2	LOCKED	Bool	false	☐	☐	☐	☐
3	OM	Word	16#0	☐	☐	☐	☐
4	Department	String[2]	''	☐	☐	☐	☐
5	Last_Name	String[40]	''	☐	☐	☐	☐
6	First_Name	String[40]	''	☐	☐	☐	☐
7	Level_Milling	Byte	16#0	☐	☐	☐	☐
8	Level_Turning	Byte	16#0	☐	☐	☐	☐
9	Personnel_No	String[2]	''	☐	☐	☐	☐
10	Issue_Date	Date	D#1990-01-01	☐	☐	☐	☐

Figure 12: Customizing the data type *type_EKSDatabase*

6.2.4. Configuring hardware

In this example, two EKS devices with PROFINET interface are configured. All Electronic-Key information is transferred from the csv file in this application. For this reason it is only necessary to read the serial number from the EKS Electronic-Key. For this purpose you can configure the smallest submodules (read/write) in the EKS (*Device view*) (see application AP000238). In this way it is possible to save memory in the control system. For this example it is necessary to configure the start address 116 in the submodule for reading the EKS.

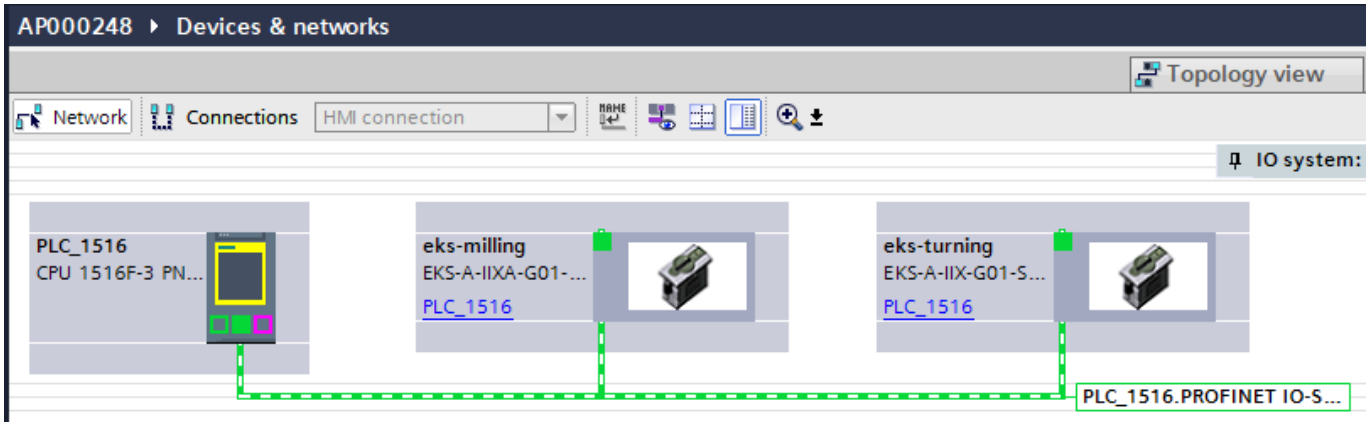


Figure 13: TIA Portal *Network view*

ⓘ	**NOTICE!** The configuration of the parameters for the submodules is described in the manual for the EKS and EKS FSA with PROFINET IO interface.

EN

6.2.5. Customizing the data block *DB_EKSData1*

The data necessary for the *FB_EKSDataService* are declared in the data block *DB_EKSData1*

Variable declaration *DB_EKSData1*

Variable	Data type	Description
Data_Key_Reader_Milling *Data_Key_Reader_Turning*	*type_EKSDatabase*	The data for the Electronic-Key currently placed in the EKS are saved in this variable
Description	*type_EKSDescription*	The structure of the data type *type_EKSDatabase* is provided to the FB in this variable
ID	*Array[0..100] of 'type_EKSDataID'*	The *ID* variable is required to store additional information such as the time stamp for Electronic-Key placement or the time stamp for the last update, the KeyID and the checksum for the EKS Data Service. This additional information is entered and stored in the DB for each Electronic-Key placed. This data type is not allowed to be changed and is saved as an array in the DB. An ID array of the same size is required for each data saving operation (Data Array). Since the data are required after a power failure/restart as well, they must be marked as remanent.
Data	*Array[0..100] of 'type_EKSDatabase'*	The same data type for data from the Electronic-Keys placed is also used for saving the data for the backup strategy. All data from the placed Electronic-Keys are entered in the data array. The data will not be overwritten until the number of Electronic-Keys placed exceeds the number of array entries. If data are overwritten, the oldest Electronic-Key entry is always overwritten first. If an Electronic-Key is not listed during the query to the EKS Data Service PC, the Electronic-Key is also deleted in the data array. The instances of the *FB_EKSDataService* can all access the same data (*Description, ID, Data*) in *DB_EKSData1*. If the data are to be available after a power failure as well, they must be marked as remanent.
EKS_Data_Temp_Milling *EKS_Data_Temp_Turning*	*type_EKSDatabase*	The *FB_EKSDataService* requires a cache outside the block of type *type_EKSDatabase*. The cache is used for copying tasks.
EKS_StatusError_Milling *EKS_StatusError_Turning*	*EKS_StatusError*	A data type *EKS_StatusError* has been added for this application. The errors and data this type contains are described in the EKS Data Service PLC manual.

DB_EKSData1

	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint
1	Static							
2	Data_Key_Reader_Milling	"type_EKSDatabase"			☑	☑	☑	
3	Data_Key_Reader_Turning	"type_EKSDatabase"			☑	☑	☑	
4	Description	"type_EKSDescription"						
5	ID	Array[0..100] of "type_EKSDataID"		☑				
6	Data	Array[0..100] of "type_EKSDataba...		☑				
7	EKS_Data_Temp_Milling	"type_EKSDatabase"						
8	EKS_Data_Temp_Turning	"type_EKSDatabase"						
9	EKS_StatusError_Milling	"EKS_StatusError"			☑	☑	☑	
10	EKS_StatusError_Turning	"EKS_StatusError"			☑	☑	☑	

Figure 14: Declaration *DB_EKSData1*

6.2.6. Customizing the block *EKS_ReadData*

Customize the data block *EKS_ReadData*. For each EKS it is necessary to declare a variable of data type *EKS_Read* (for reading all the data from the input area of the control system), *Word* (fault display) and *type_EKSKeyID* (separation of the Electronic-Key serial number).

EKS_ReadData

		Name	Data type
1		Static	
2		ReadData_Milling	"EKS_Read"
3		Error_Read_Milling	Word
4		ReadData_Turning	"EKS_Read"
5		Error_Read_Turning	Word
6		EKSKeyID_Milling	"type_EKSKeyID"
7		EKSKeyID_Turning	"type_EKSKeyID"

Figure 15: Configuring the parameters for the data block *EKS_ReadData*

6.2.7. Calling the function *EKS_Communication_FC*

Call the function *EKS_Communication_FC* in your function block (here: *FB_EKS_Euchner)*. This block must be called and the parameters configured for each EKS. Assign the variables to the inputs and outputs. On the input *EKS_PN_ReadModule*, enter the hardware identifier for the *Read* module in the corresponding EKS. The data from the input area are copied to the *EKS_ReadData* data block for the related EKS.

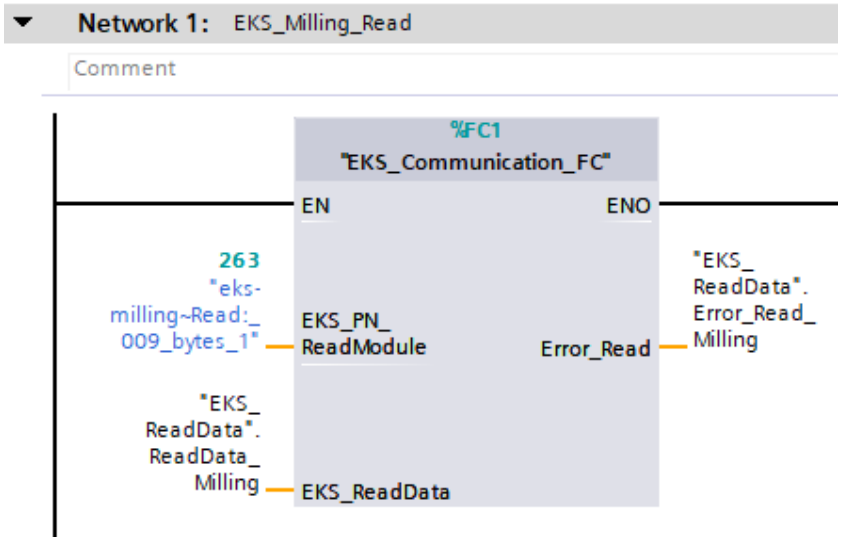


Figure 16: Calling the *EKS_Communication_FC* (for example: *Milling*)

EN

Then the Electronic-Key serial number must be copied from the EKS receive data (*Receive_data*) to the serial number variable (*type_EKSKeyID*). This action can be accomplished with two move commands.

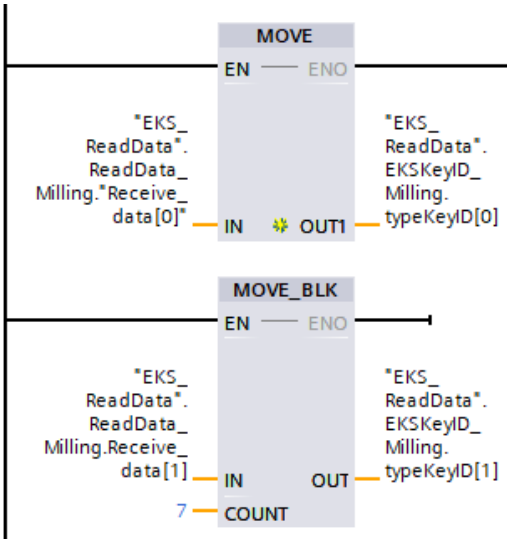


Figure 17: Copying serial number to the variable *type_EKSKeyID*

	NOTICE! It is to be noted that, due to a 16-bit limit imposed by Siemens, the first byte of the memory in the Electronic-Key in the DB (*EKS_ReadData*) is NOT in the *ReceiveData* array, instead it must be listed as an extra byte.

6.2.8. Calling the function block *FB_EKSDataService*

Call the function block *FB_EKSDataService* in the main program *Main* (*OB1*). This block must be called and the parameters configured for each EKS.

	NOTICE! The inputs and outputs of the *FB_EKSDataService* are described in detail in the EKS Data Service PLC manual.

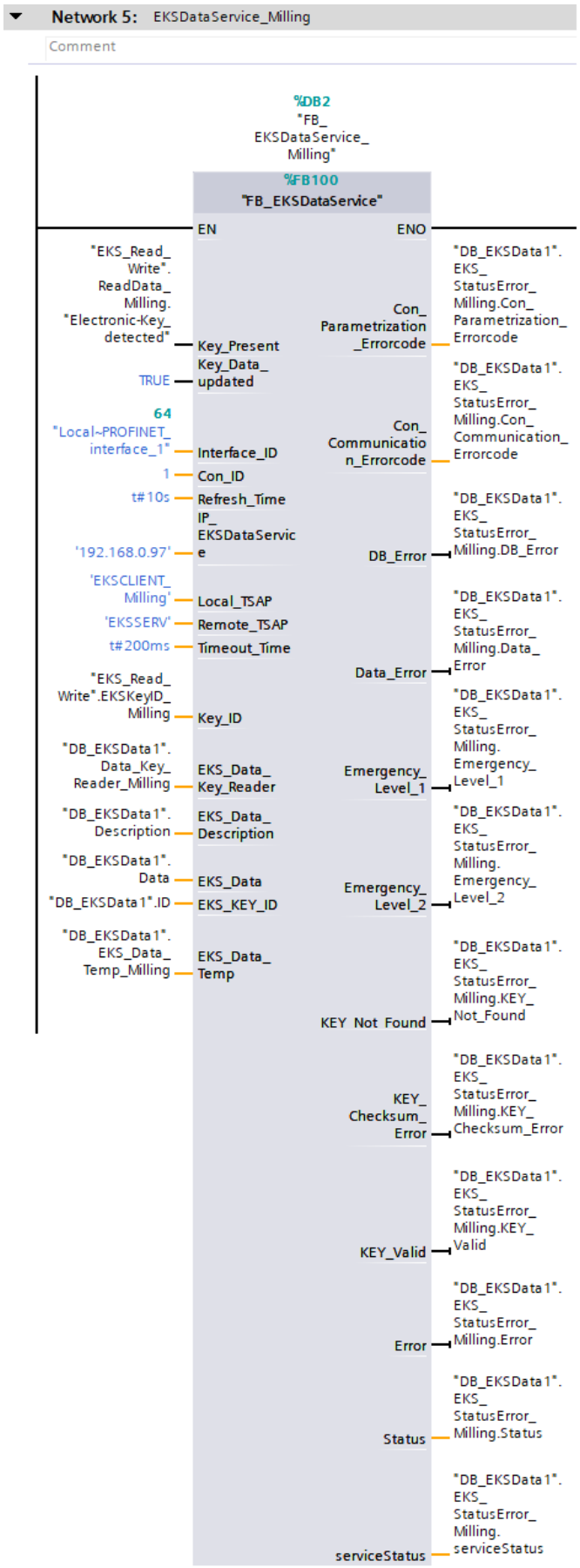


Figure 18: Calling a *FB_EKSDataService* (for example: *Milling*)

6.2.9. Monitoring the data in the data block *DB_EKSData1*

After the parameters have been configured for all blocks and UDTs, load your program into the control system. Check on the *LOG* tab in the service on the PC whether the connection to the control system has been established. Program a couple of Electronic-Keys in the EKM and place them in your system. As soon as you view the block DB_EKSData1 online in TIA Portal, you will see the data retrieved from the csv file.

DB_EKSData1

	Name	Data type	Start value	Monitor value	Retain	Accessible f...	Writa...	Visible in ...
1	Static				☐	☐	☐	☐
2	Data_Key_Reader_Milling	"type_EKSDatabase"			☐	☑	☑	☑
3	KeyID	String[16]	''	'02877825FF001032'	☐	☐	☐	☐
4	LOCKED	Bool	false	FALSE	☐	☐	☐	☐
5	OM	Word	16#0	16#F00F	☐	☐	☐	☐
6	Department	String[2]	''	'QS'	☐	☐	☐	☐
7	Last_Name	String[40]	''	'Doe'	☐	☐	☐	☐
8	First_Name	String[40]	''	'John'	☐	☐	☐	☐
9	Level_Milling	Byte	16#0	16#01	☐	☐	☐	☐
10	Level_Turning	Byte	16#0	16#03	☐	☐	☐	☐
11	Personnel_No	String[2]	''	'10'	☐	☐	☐	☐
12	Issue_Date	Date	D#1990-01-01	D#2019-07-04	☐	☐	☐	☐
13	Data_Key_Reader_Turning	"type_EKSDatabase"			☐	☑	☑	☑
14	Description	"type_EKSDescription"			☐	☐	☐	☐
15	ID	Array[0..100] of "type_EKSDataID"			☑	☐	☐	☐
16	Data	Array[0..100] of "type_EKSDataba...			☑	☐	☐	☐
17	EKS_Data_Temp_Milling	"type_EKSDatabase"			☐	☐	☐	☐
18	EKS_Data_Temp_Turning	"type_EKSDatabase"			☐	☐	☐	☐
19	EKS_StatusError_Milling	"EKS_StatusError"			☐	☑	☑	☑
20	EKS_StatusError_Turning	"EKS_StatusError"			☐	☑	☑	☑

Figure 19: Data from csv file in *DB_EKSData1*

7. Integration of PLC library for S7-300/400 into TIA Portal V14 SP1

For the S7-300/400 you will need the following libraries:

EKS_Data_Service_PLC_Library_TIA_300_400_YYYYMMDD

Library_EKS_Data_Service-TIAV14SP1_PLC-300_400_YYYYMMDD

7.1. Retrieving the library

Retrieve the libraries as described in section 6.1.

7.2. Integration of the EKS Data Service into the PLC

7.2.1. Copying blocks to the project

1. Open the *library* EKS_Data_Service_PLC_Library_TIA_300_400_YYYYMMDD and copy the blocks to the folders in the *project navigation* to suit your control system environment. All the blocks from the library are not used in this example.

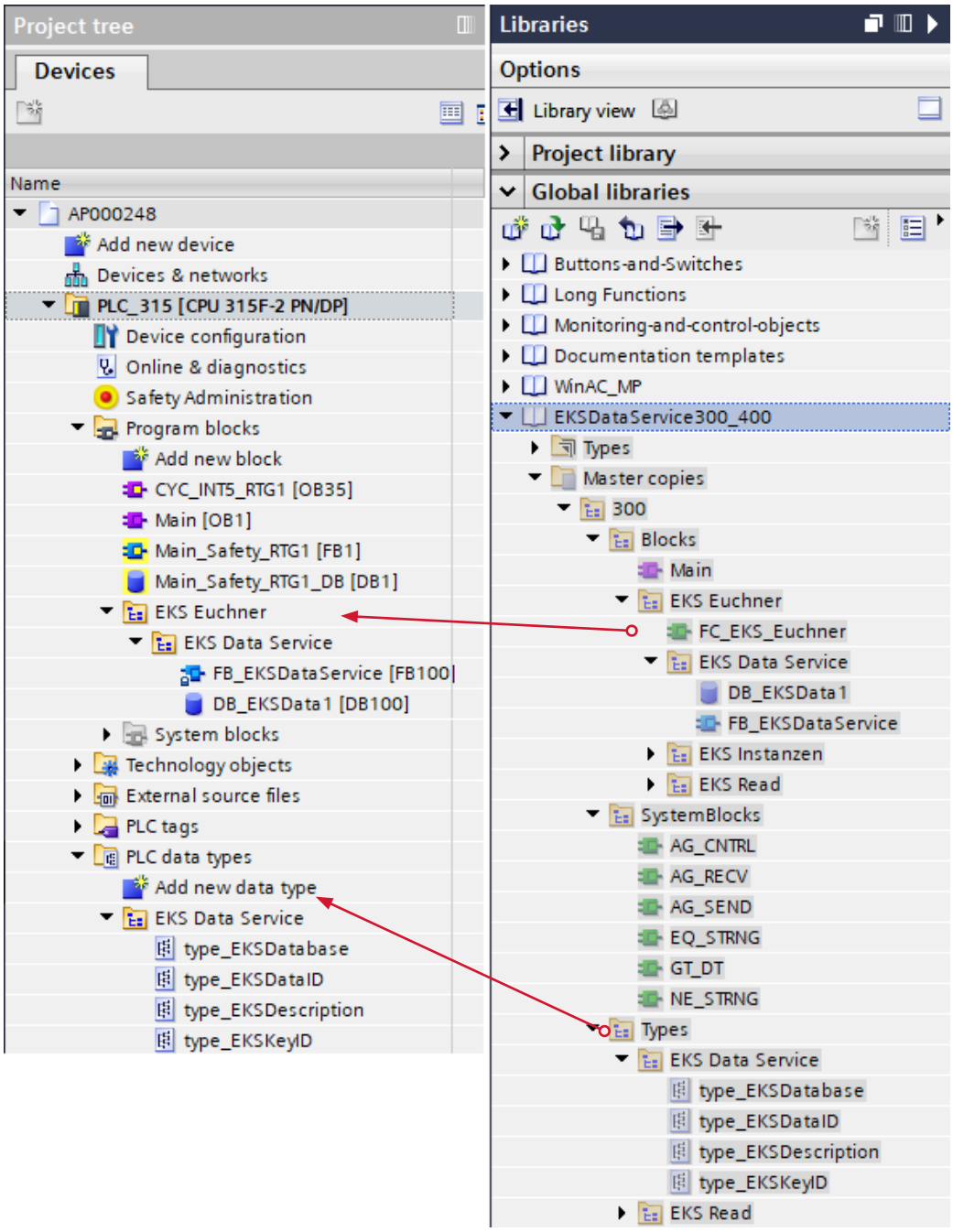


Figure 20: Copying blocks to the project

EN

2. Open the library Library_EKS_Data_Service-TIAV14SP1_PLC-300_400_YYYYMMDD and copy the blocks to the folders in the *project navigation*. This library is used to read the EKS Electronic-Key. It is not imperative to use this library.

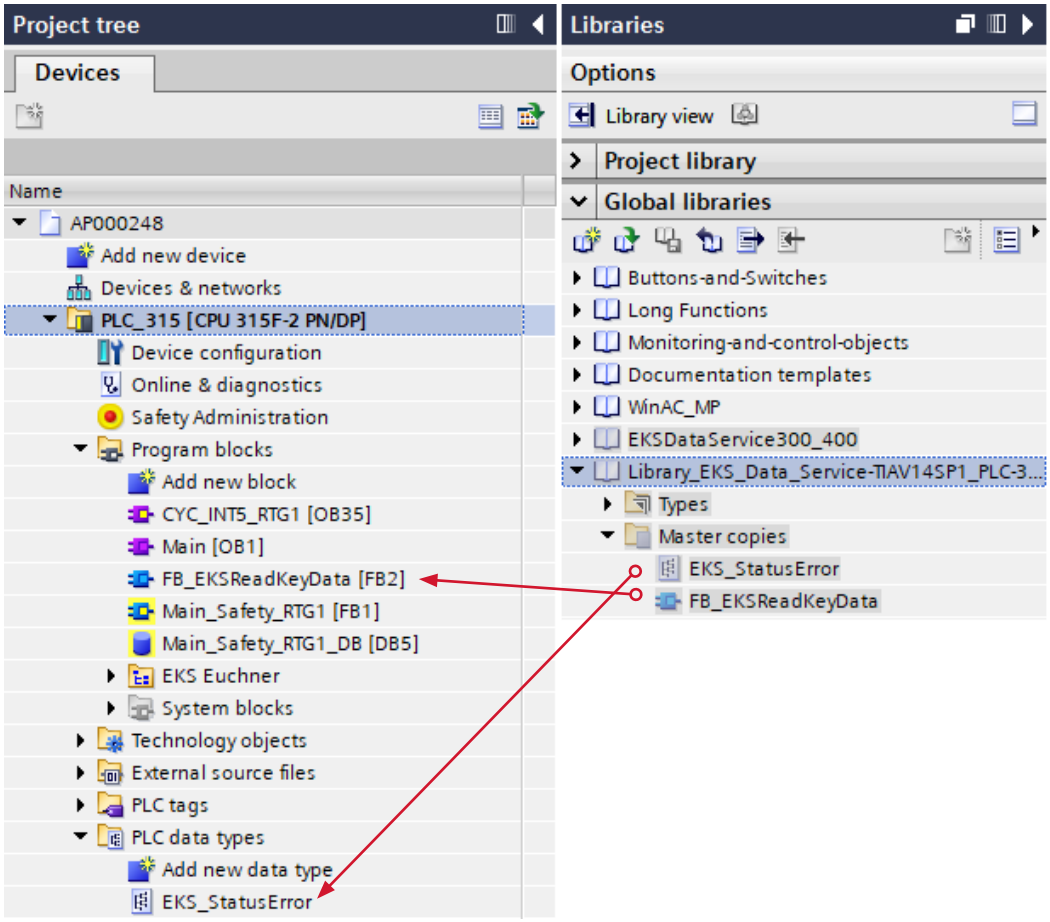


Figure 21: Copying blocks to the project

7.2.2. Customizing the data type *type_EKSDescription*

Open the data type *type_EKSDescription*. Map the *EKM PLC DATA TYPES Name* and *PLC-Index* configured in the service.

type_EKSDescription

	Name	Data type	Default value	Accessible f...	Writa...	Visible in ...	Setpoint
1	KeyID	Int	1	☐	☐	☐	☐
2	LOCKED	Int	2	☐	☐	☐	☐
3	OM	Int	3	☐	☐	☐	☐
4	Department	Int	4	☐	☐	☐	☐
5	Last_Name	Int	5	☐	☐	☐	☐
6	First_Name	Int	6	☐	☐	☐	☐
7	Level_Milling	Int	7	☐	☐	☐	☐
8	Level_Turning	Int	8	☐	☐	☐	☐
9	Personnel_No	Int	9	☐	☐	☐	☐
10	Date	Int	10	☐	☐	☐	☐

Figure 22: Customizing the data type *type_EKSDescription*

7.2.3. Customizing the data type *type_EKSDatabase*

Open the data type *type_EKSDatabase*. Map the *EKM PLC DATA TYPES Name* and *PLC-Type* configured in the service. Pay attention to the allocation of the EKM data types and PLC data types as described in the EKS Data Service PC manual.

type_EKSDatabase

	Name	Data type	Default value	Accessible f...	Writa...	Visible in ...	Setpoint
1	KeyID	String[16]	''	☐	☐	☐	☐
2	LOCKED	Bool	false	☐	☐	☐	☐
3	OM	Word	16#0	☐	☐	☐	☐
4	Department	String[2]	''	☐	☐	☐	☐
5	Last_Name	String[40]	''	☐	☐	☐	☐
6	First_Name	String[40]	''	☐	☐	☐	☐
7	Level_Milling	Byte	16#0	☐	☐	☐	☐
8	Level_Turning	Byte	16#0	☐	☐	☐	☐
9	Personnel_No	String[2]	''	☐	☐	☐	☐
10	Issue_Date	Date	D#1990-01-01	☐	☐	☐	☐

Figure 23: Customizing the data type *type_EKSDatabase*

7.2.4. Creating the data type *type_EKSRead*

Because in this application example, data are retrieved exclusively from a database, only the bits *Device_ready_for_operation* and *ElectronicKey_detected* as well as *EKSKeyID* of data type *type_EKSKeyID* are required.

type_EKSRead

	Name	Data type	Default value	Visible in ...	Setpoint
1	Device_ready_for_operation	Bool	false	☑	☐
2	ElectronicKey_detected	Bool	false	☑	☐
3	EKSKeyID	"type_EKSKeyID"		☑	☐

Figure 24: Creating the data type *type_EKSRead*

7.2.5. Configuring hardware and adding connections

In this example, two EKS devices with PROFINET interface are configured. All Electronic-Key information is transferred from the csv file in this application. For this reason it is only necessary to read the serial number from the EKS Electronic-Key. For this purpose you can configure the smallest submodules (read/write) in the EKS (*Device view*) (see application AP000238). In this way it is possible to save memory in the control system. For this example it is necessary to configure the start address 116 in the submodule for reading the EKS.

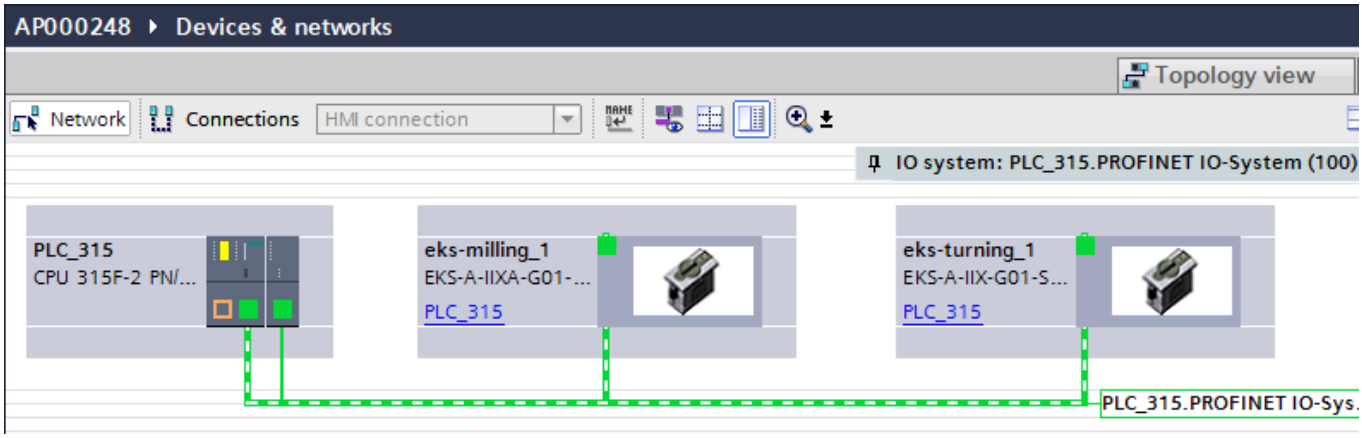


Figure 25: TIA Portal *Network view*

EN

Open the properties for the communication processor and open the *Ethernet addresses* view. Connect the *Subnet* interface to the PLC's PROFINET network.

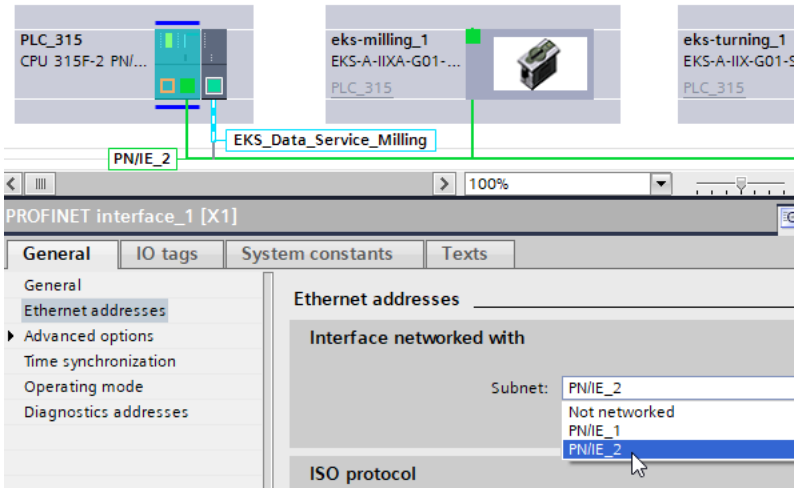


Figure 26: Connecting interface to subnet

A connection to the EKS Data Service PC must be added in the PLC for each instance of the block *FB_EKSDataService* (for each EKS). To add a connection, proceed as follows:

1. In the Network view, click *Connections* to display the connections.

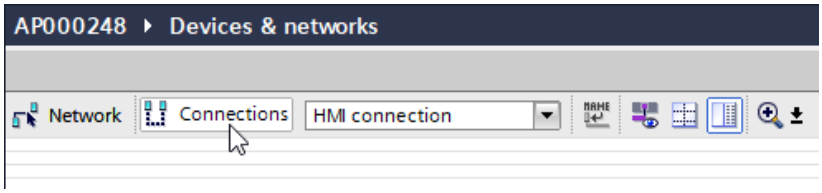


Figure 27: Displaying connections

2. Add a new connection. Click the control system using the right mouse button and select *Add new connection*.

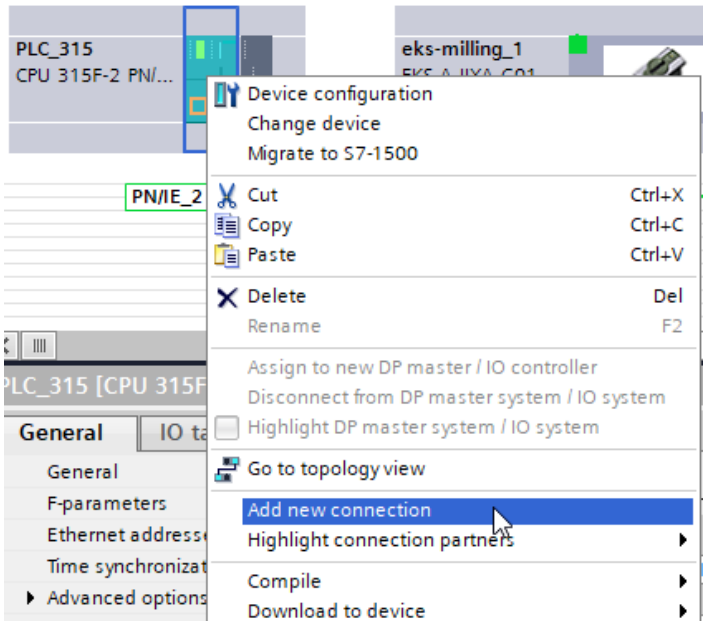


Figure 28: Adding new connection

3. Select the type for the connection. This is always an *ISO-on-TCP connection*. Click *Establish active connection* and complete the action using *Add*.

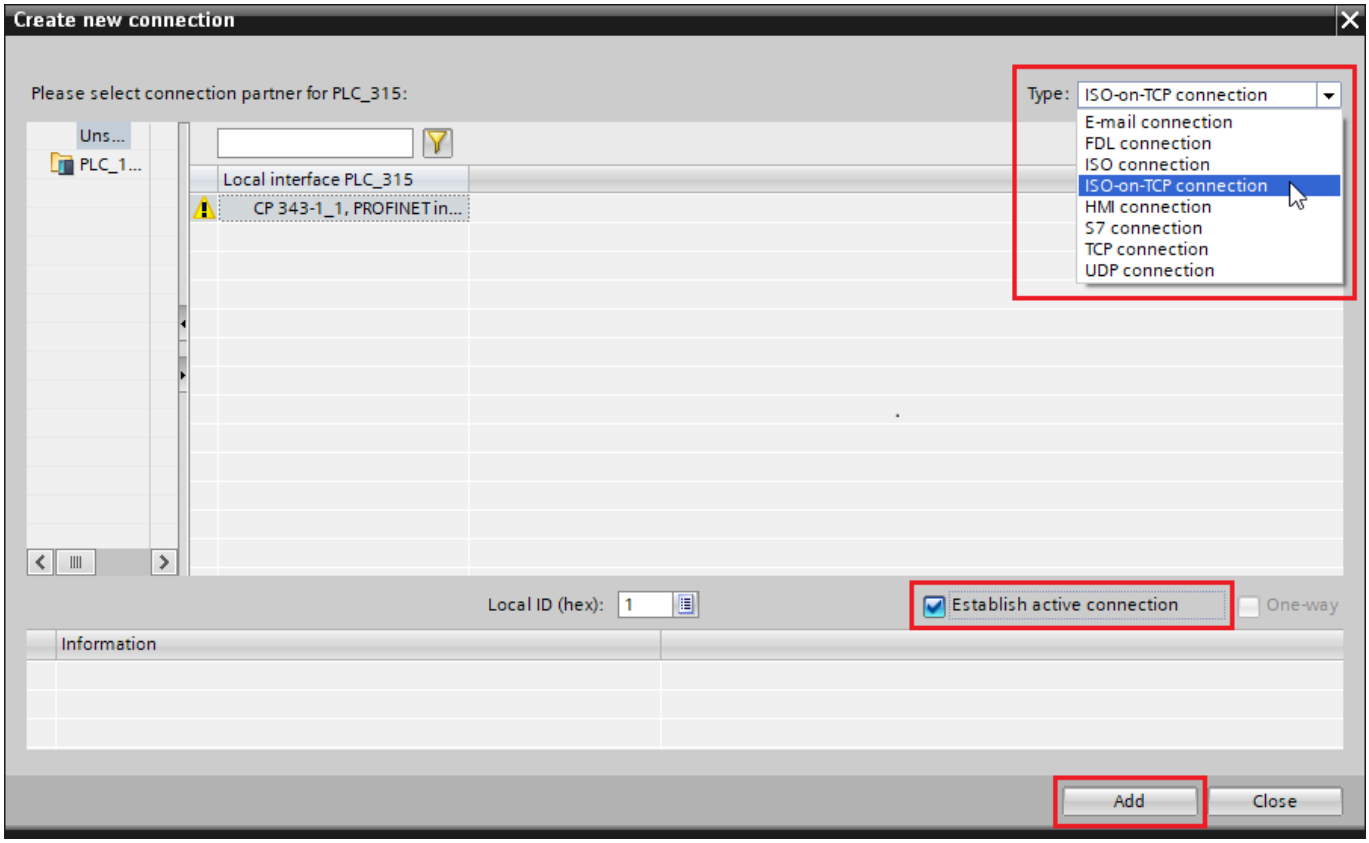


Figure 29: Adding connection

4. Open the general properties for the connection added. The name of the connection can be changed here for better identification.

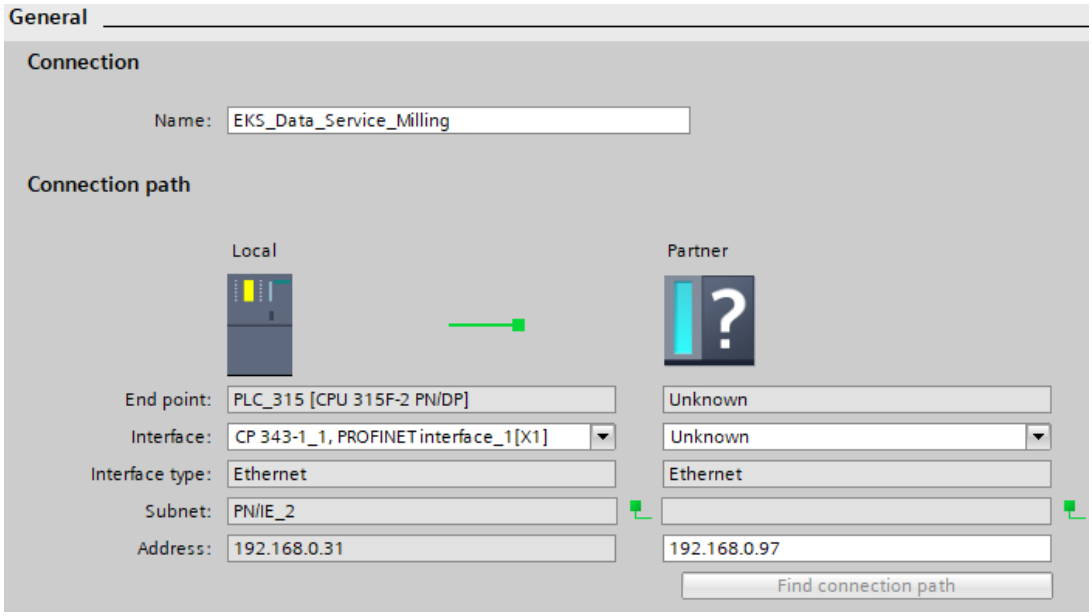


Figure 30: General properties

EN

5. Now open the *Address details*. In the Partner column, complete the *IP (dec)* and *TSAP (ASCII)* rows. The partner IP address is the address of the server on which EKS Data Service PC is installed. The partner *TSAP* must be called *EKSSERV* for every connection. The local *TSAP* should be unambiguous if there are several instances of the *FB_EKSDataService*. This name is displayed in EKS Data Service PC.

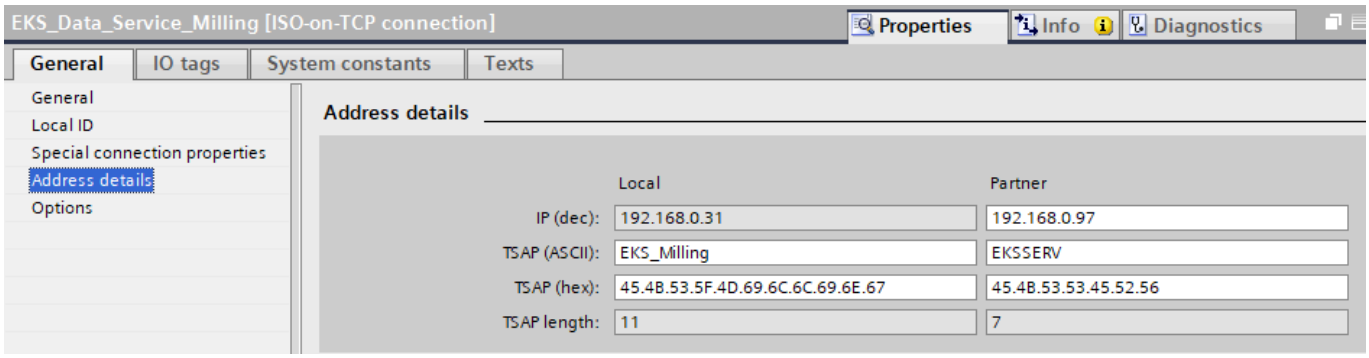


Figure 31: Settings for address details

6. During the subsequent configuration of the parameters for the *FB_EKSDataService*, you will need the *Local ID* (*Con_ID*) as well as the hardware address/*LADDR* (*Interface_ID*) for each connection.

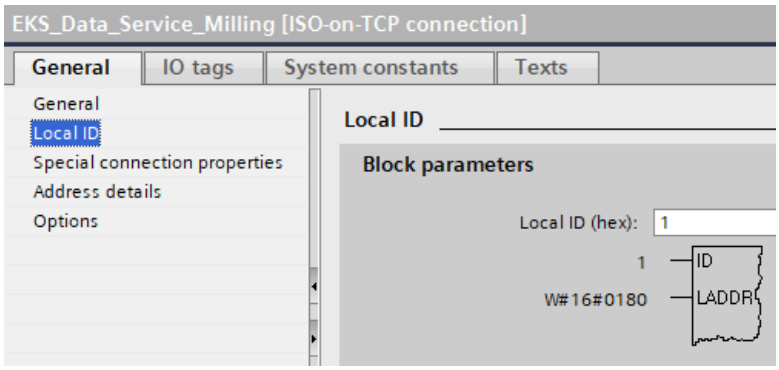


Figure 32: Local ID settings

Our example contains two EKS. Two connections are added and the parameters configured.

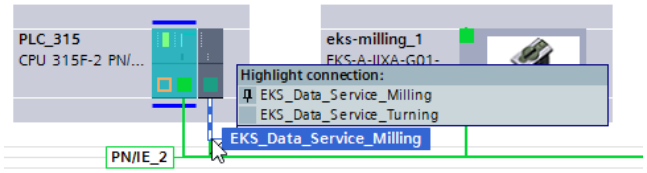


Figure 33: Connections with configured parameters

	NOTICE! These steps must be repeated for each instance of the block *FB_EKSDataService*.

7.2.6. Customizing the data block *DB_EKSData1*

Variable	Data type	Description
Data_Key_Reader_Milling *Data_Key_Reader_Turning*	*type_EKSDatabase*	The data for the Electronic-Key currently placed in the EKS are saved in this variable
Description	*type_EKSDescription*	The structure of the data type *type_EKSDatabase* is provided to the FB in this variable
ID	*Array[0..100] of 'type_EKSDataID'*	The *ID* variable is required to store additional information such as the time stamp for Electronic-Key placement or the time stamp for the last update, the KeyID and the checksum for the EKS Data Service. This additional information is entered and stored in the DB for each Electronic-Key placed. This data type is not allowed to be changed and is saved as an array in the DB. An ID array of the same size is required for each data saving operation (Data Array). Since the data are required after a power failure/restart as well, they must be marked as remanent.
Data	*Array[0..100] of 'type_EKSDatabase'*	The same data type for data from the Electronic-Keys placed is also used for saving the data for the backup strategy. All data from the placed Electronic-Keys are entered in the data array. The data will not be overwritten until the number of Electronic-Keys placed exceeds the number of array entries. If data are overwritten, the oldest Electronic-Key entry is always overwritten first. If an Electronic-Key is not listed during the query to the EKS Data Service PC, the Electronic-Key is also deleted in the data array. The instances of the *FB_EKSDataService* can all access the same data (*Description, ID, Data*) in *DB_EKSData1*. If the data are to be available after a power failure as well, they must be marked as remanent.
EKS_StatusError_Milling *EKS_StatusError_Turning*	*EKS_StatusError*	A data type *EKS_StatusError* has been added for this application. The errors and data this type contains are described in the EKS Data Service PLC manual.

DB_EKSData1

	Name	Data type	Offset	S..	Retain	Visible in ...	Setpoint
1	Static				☐	☐	☐
2	Data_Key_Reader_Milling	"type_EKSDatabase"	0.0		☑	☑	☐
3	ErrorStatus_Milling	"EKS_StatusError"	118.0		☑	☑	☐
4	Data_Key_Reader_Turning	"type_EKSDatabase"	128.0		☑	☑	☐
5	ErrorStatus_Turning	"EKS_StatusError"	246.0		☑	☑	☐
6	Description	"type_EKSDescription"	256.0		☑	☑	☐
7	ID	Array[0..20] of "type_EKSDataID"	276.0		☑	☑	☐
8	Data	Array[0..20] of "type_EKSDatabase"	822.0		☑	☑	☐

Figure 34: Configuring the parameters for *DB_EKSData1*

EN

7.2.7. Calling the function block *FB_EKSReadKeyData*

Call the function *FB_EKSReadKeyData* in a function block you prepared previously (here: *FB_EKS_Euchner)*. This block must be called and the parameters configured for each EKS. So that you do not need to create countless data blocks per instance, you can use multi-instances for this purpose. Enter the input address of the status byte from the EKS on the input *StartAddressStatus* and the start address at which the Electronic-Key serial number starts in your application on the input *StartAddressKeyID*. In our example the Electronic-Key serial number starts in input area of the control system for the EKS *Milling* at 1 because the start address 116 is configured in the parameters for the EKS Read module.

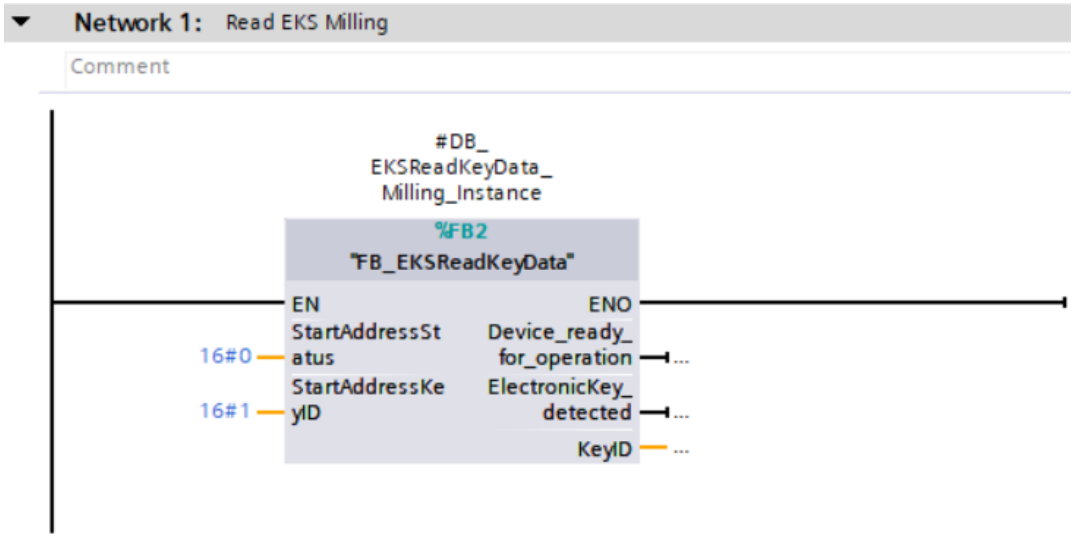


Figure 35: Calling *FB_EKSReadKeyData*

	NOTICE! Please note that hexadecimal values must be entered on the block's inputs.

7.2.8. Calling the function block *FB_EKSDataService*

Call the function block *FB_EKSDataService* in the same FB (*FB_EKS_Euchner)* as *FB_EKSReadKeyData* previously. This block *FB_EKSDataService* must be called and the parameters configured for each individual EKS. Then call the function block *FB_EKS_Euchner* in the main program *Main* (*OB1*).

	NOTICE! The inputs and outputs of the *FB_EKSDataService* are described in detail in the EKS Data Service PLC manual.

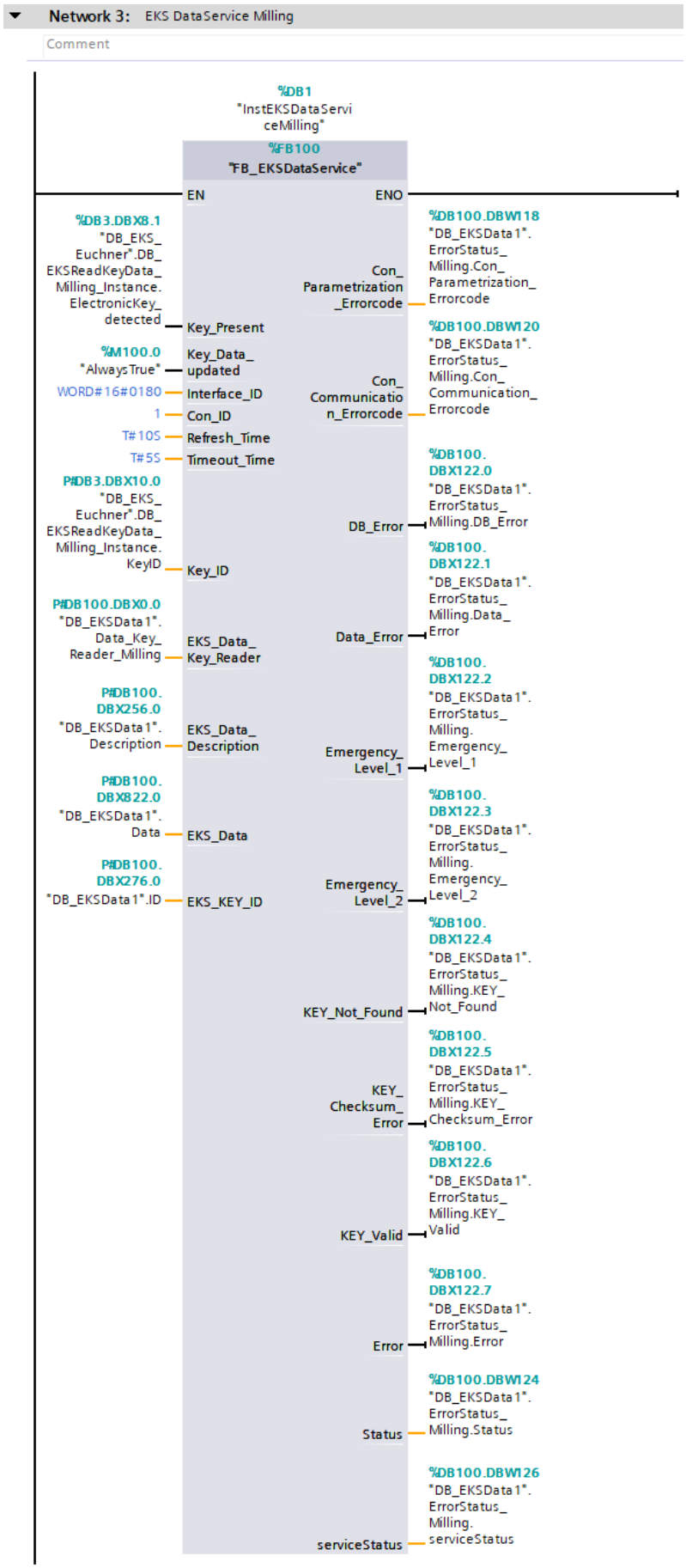


Figure 36: Calling a *FB_EKSDataService* (for example: *Milling*)

EN

7.2.9. Monitoring the data in the data block *DB_EKSData1*

After the parameters have been configured for all blocks and UDTs, load your program into the control system. Check on the *LOG* tab in the service on the PC whether the connection to the control system has been established. Program a couple of Electronic-Keys in the EKM and place them in your system. As soon as you view the block DB_EKSData1 online in TIA Portal, you will see the data retrieved from the csv file.

DB_EKSData1

	Name	Data type	Offset	Start value	Monitor value	Retain	Visible in ...
1	Static					☐	☐
2	Data_Key_Reader_Milling	"type_EKSDatabase"	0.0			☑	☑
3	KeyID	String[16]	0.0	''	'02877825FF0010...	☑	☐
4	LOCKED	Bool	18.0	false	FALSE	☑	☐
5	OM	Word	20.0	16#0	16#F00F	☑	☐
6	Department	String[2]	22.0	''	'QS'	☑	☐
7	Last_Name	String[40]	26.0	''	'Doe'	☑	☐
8	First_Name	String[40]	68.0	''	'John'	☑	☐
9	Level_Milling	Byte	110.0	16#0	16#01	☑	☐
10	Level_Turning	Byte	111.0	16#0	16#03	☑	☐
11	Personnel_No	String[2]	112.0	''	'10'	☑	☐
12	Issue_Date	Date	116.0	D#1990-01-01	D#2019-07-04	☑	☐
13	ErrorStatus_Milling	"EKS_StatusError"	118.0			☑	☑
14	Data_Key_Reader_Turning	"type_EKSDatabase"	128.0			☑	☑
15	ErrorStatus_Turning	"EKS_StatusError"	246.0			☑	☑
16	Description	"type_EKSDescription"	256.0			☑	☑
17	ID	Array[0..20] of "type_EKSDataID"	276.0			☑	☑
18	Data	Array[0..20] of "type_EKSDatabase"	822.0			☑	☑

Figure 37: Data from csv file in *DB_EKSData1*

8. Integration of PLC library for S7-300/400 into STEP7 Manager

Use the following library for S7-300/400:

EKS_Data_Service_PLC_Library_STEP_7_YYMMDD

8.1. Retrieving the library

1. On the menu bar, click *File* and select *Retrieve*.
2. Select the folder with the library and retrieve it to the required destination folder.

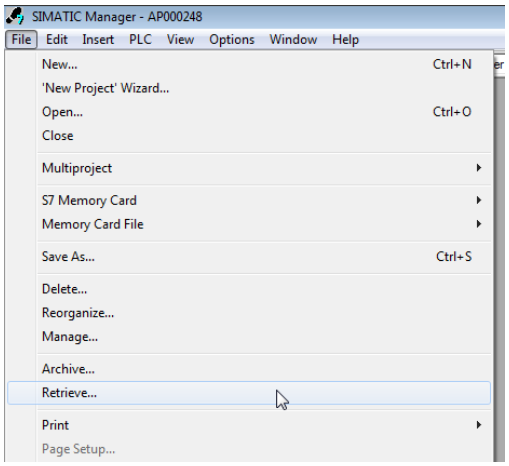


Figure 38: Retrieving library

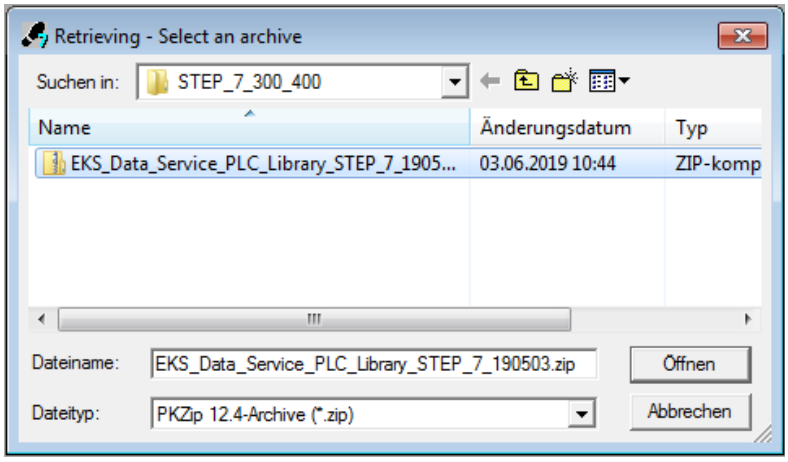


Figure 39: Selecting library

TIP!

If the library has already been retrieved, click *Open...* and on the *Libraries* tab select the library you want to add to the project.

8.2. Integration of the EKS Data Service into the PLC

8.2.1. Copying blocks to the project

After you have opened the library, copy the blocks to the folders in the project navigation as shown in the figure below to suit your control system environment.

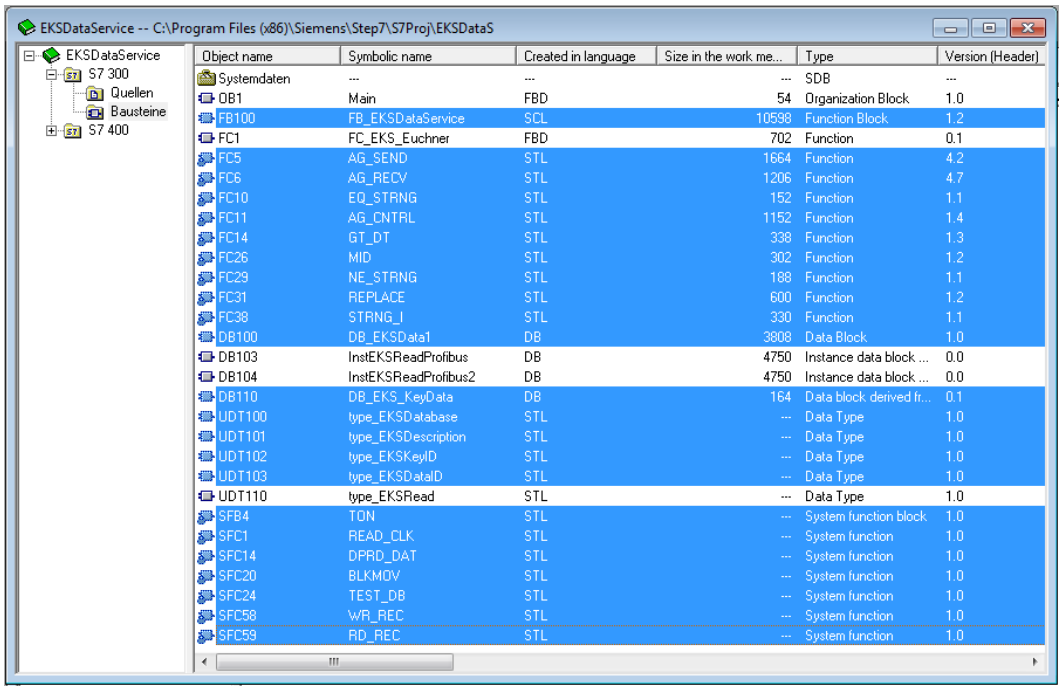


Figure 40: Copying blocks to the project

EN

8.2.2. Customizing the data type *type_EKSDescription*

Open the data type *type_EKSDescription*. Map the *EKM PLC DATA TYPES Name* and *PLC-Index* configured in the service.

Address	Name	Type	Initial value
0.0		STRUCT	
+0.0	KeyID	INT	1
+2.0	LOCKED	INT	2
+4.0	OM	INT	3
+6.0	Department	INT	4
+8.0	Last_Name	INT	5
+10.0	First_Name	INT	6
+12.0	Level_Milling	INT	7
+14.0	Level_Turning	INT	8
+16.0	Personnel_No	INT	9
+18.0	Issue_Date	INT	10
=20.0		END_STRUCT	

Figure 41: Customizing the data type *type_EKSDescription*

8.2.3. Customizing the data type *type_EKSDatabase*

Open the data type *type_EKSDatabase*. Map the *EKM PLC DATA TYPES Name* and *PLC-Type* configured in the service. Pay attention to the allocation of the EKM data types and PLC data types as described in the EKS Data Service PC manual.

Address	Name	Type	Initial value
0.0		STRUCT	
+0.0	KeyID	STRING[16]	''
+18.0	LOCKED	BOOL	FALSE
+20.0	OM	WORD	W#16#0
+22.0	Department	STRING[2]	''
+26.0	Last_Name	STRING[40]	''
+68.0	First_Name	STRING[40]	''
+110.0	Level_Milling	BYTE	B#16#0
+111.0	Level_Turning	BYTE	B#16#0
+112.0	Personnel_No	STRING[2]	''
+116.0	Issue_Date	DATE	D#1990-1-1
=118.0		END_STRUCT	

Figure 42: Customizing the data type *type_EKSDatabase*

8.2.4. Configuring hardware and adding connections

In this example, two EKS devices with PROFINET interface are configured. All Electronic-Key information is transferred from the csv file in this application. For this reason it is only necessary to read the serial number from the EKS Electronic-Key. For this purpose you can configure the smallest submodules (read/write) in the EKS (*HW Config*). In this way it is possible to save memory in the control system. For this example it is necessary to configure the start address 116 in the submodule for reading the EKS.

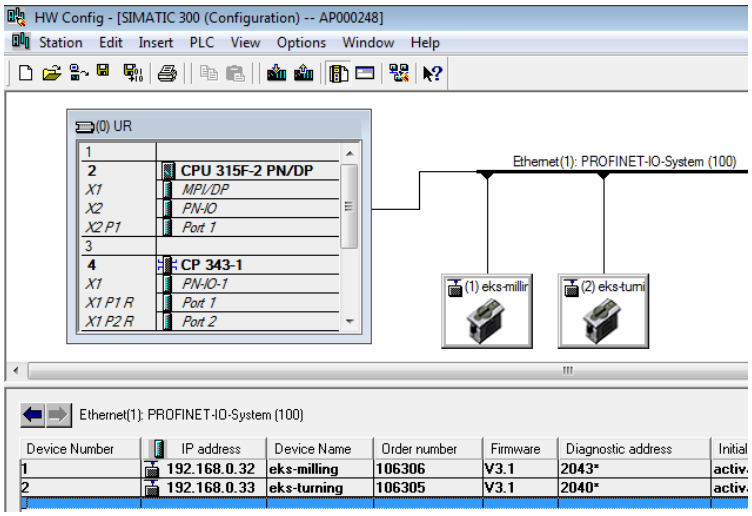


Figure 43: HW Config view

Open the *object properties* for the PROFINET interface on the communication processor and then open the *Properties* in the *Interface* group box. Assign to the communication processor an IP address in the same subnet as the existing PLC subnet and select the existing *subnet* in the PROFINET network for the PLC.

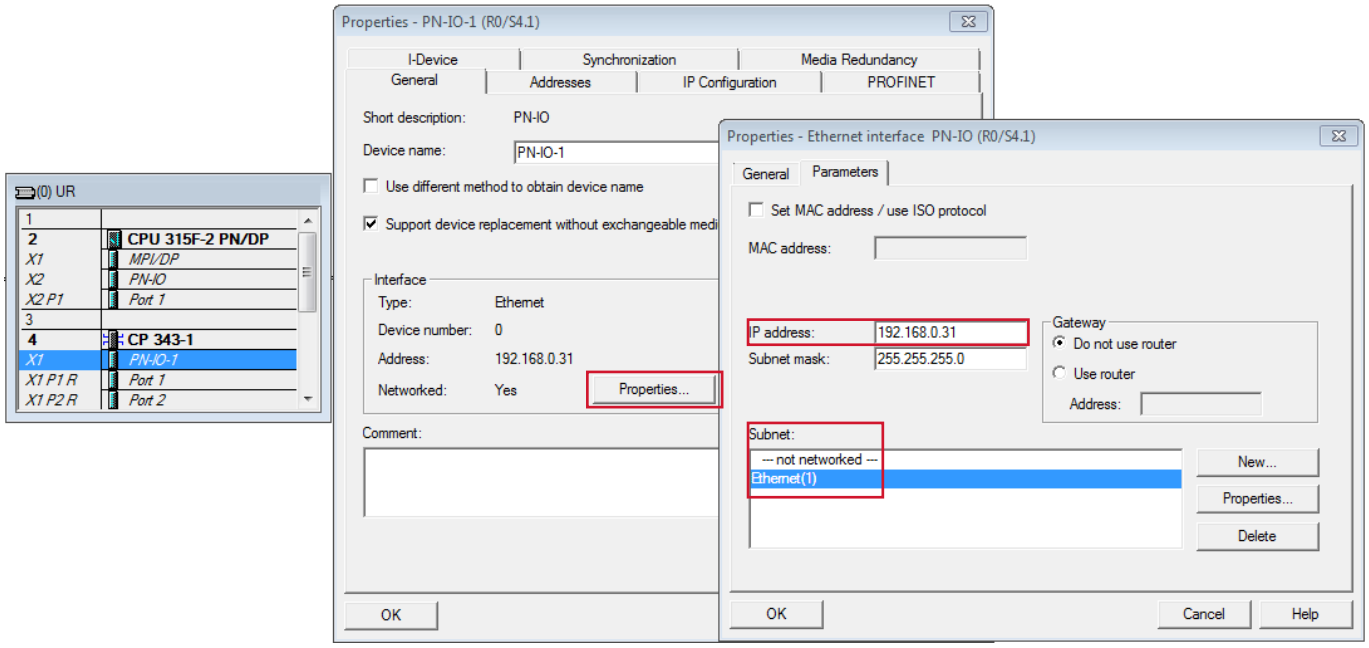


Figure 44: Connecting interface to subnet

EN

A connection to the EKS Data Service PC must be added in the PLC for each instance of the block *FB_EKSDataService* (that is for each EKS). Open NetPro using the icon on the toolbar

1. The EKS Data Service PC must be established in the NetPro configuration to be able to add a connection. On the right, add an *Other Station* to the network using drag&drop and open the *Properties* by double-clicking. On the *General* tab you can assign a different name to the station, in our example *EKSDataService*. Then open the *Interfaces* tab.

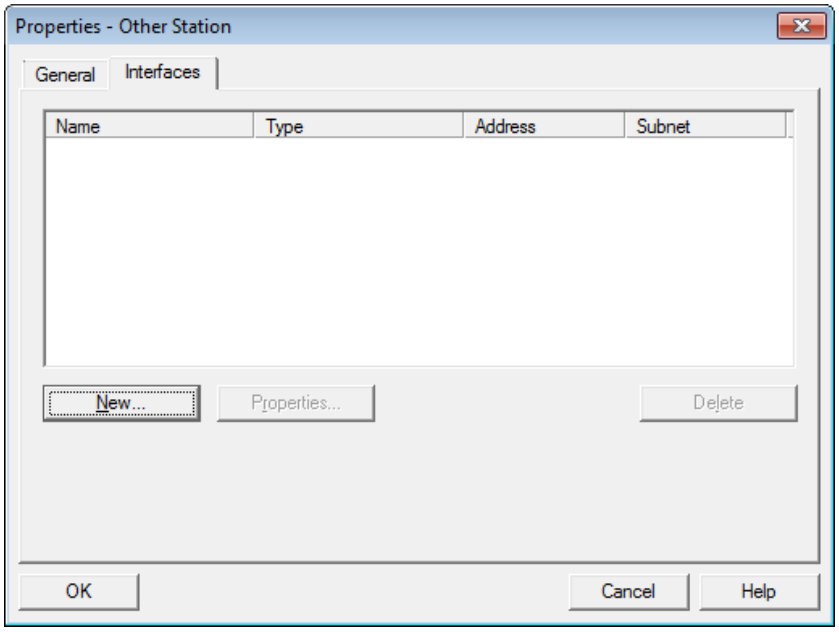


Figure 45: Adding interface for *Other Station* (*EKSDataService*)

2. Click *New...* and select the type *Industrial Ethernet*.

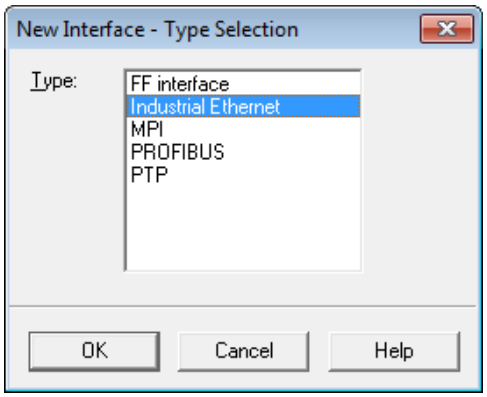


Figure 46: Selecting type of connection

3. In the next step the *Properties* for the *Ethernet interface* on the *Parameters* tab are opened automatically. It is important here to clear the *Set MAC address / use ISO protocol* check box. Then enter the *IP address* and the *subnet mask* for the server and the *subnet* for the PLC (in the example: Ethernet(1)). Accept the entry by clicking *OK*.

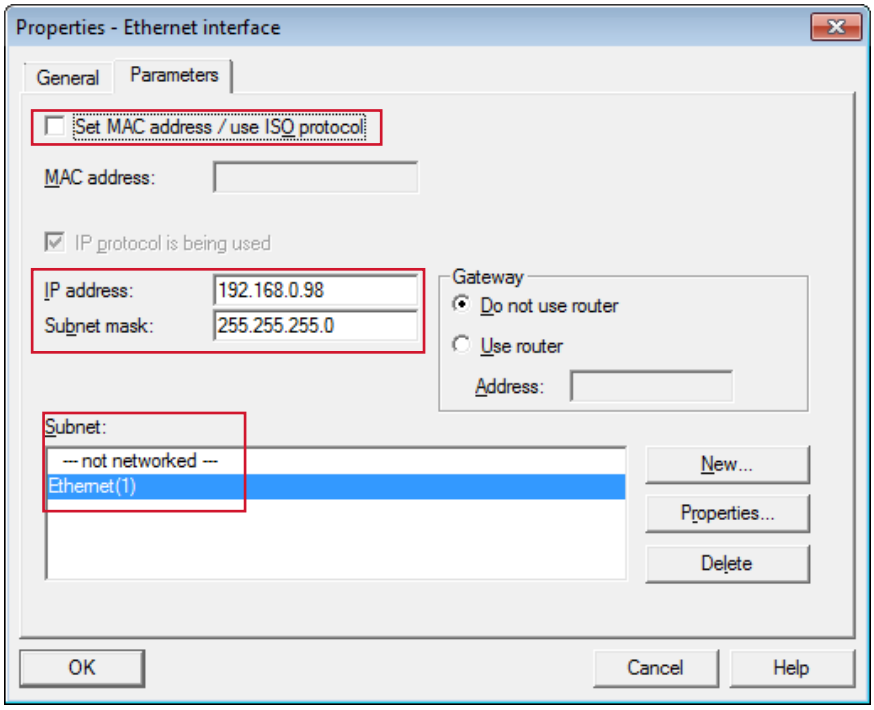


Figure 47: *Properties - Ethernet interface*

4. You can now add the connections. For this purpose, select the CPU in the NetPro configurator as shown in the figure below. Then click the *Add connection* button on the toolbar 

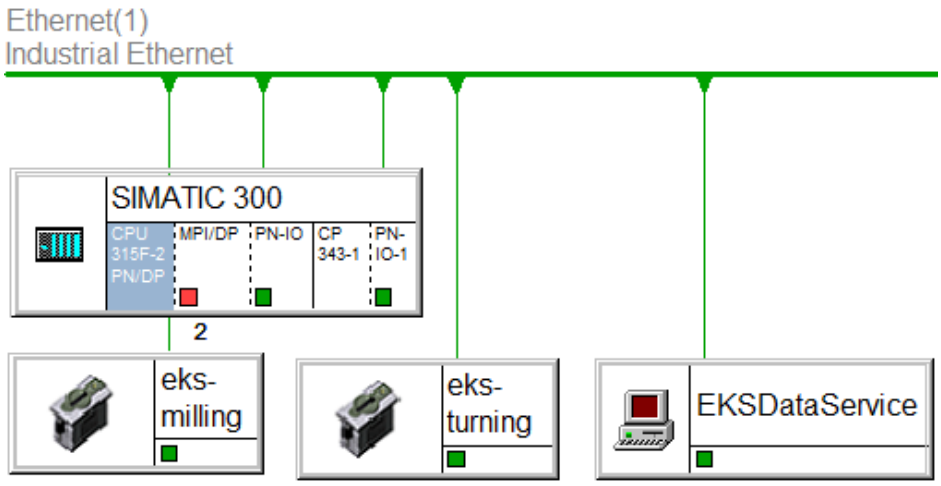


Figure 48: Selecting the CPU to add connection

EN

5. Select as *Unspecified* connection partner and the connection type *ISO-on-TCP connection*.

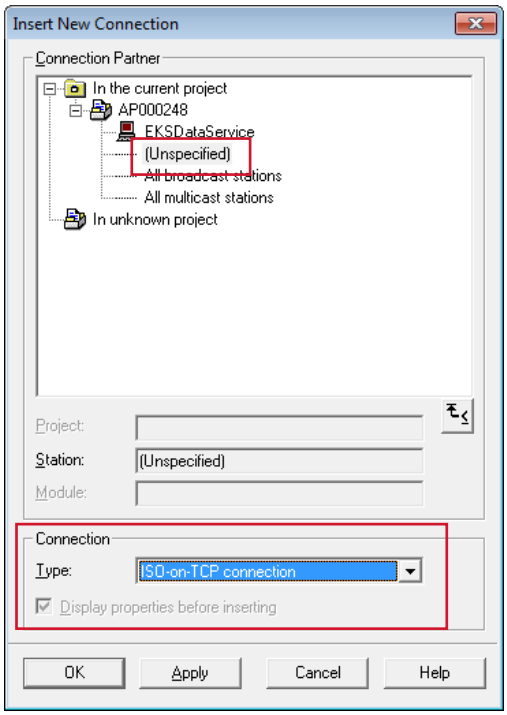


Figure 49: *Adding a new connection*

6. The Properties window for the connection to be added is opened automatically. You can give the connection a name on the General Information tab. It is important to select the *Active connection establishment* check box.

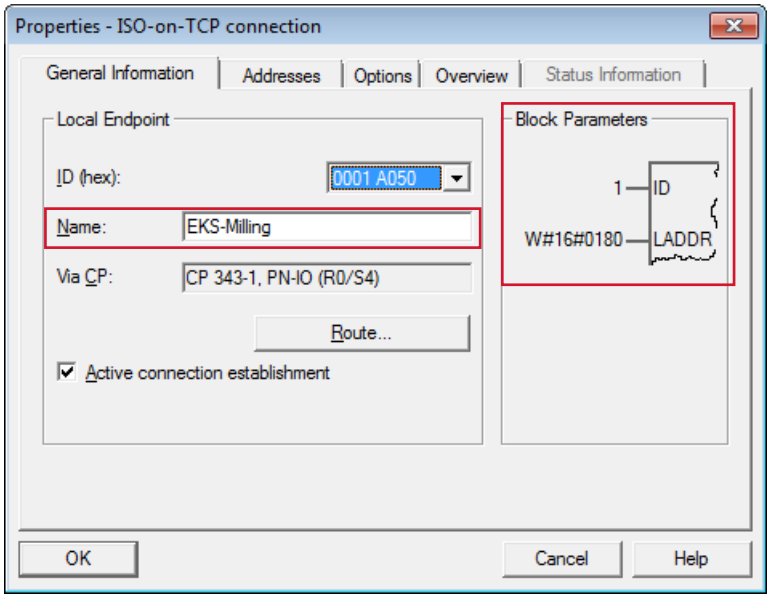


Figure 50: General and Block parameters for a new connection

	NOTICE! During the subsequent configuration of the parameters for the *FB_EKSDataService*, you will need the *Local ID* (*Con_ID*) as well as the hardware address/*LADDR* (*Interface_ID*) for each connection.

7. Open the *Addresses* tab. Here you must enter the *TSAP*. Give the local *TSAP* an arbitrary name and the remote *TSAP* the name *EKSSERV* (fixed word, see DataService PLC manual). Enter the IP address of the server on which EKS Data Service PC is installed. Accept your entries by clicking *OK*.

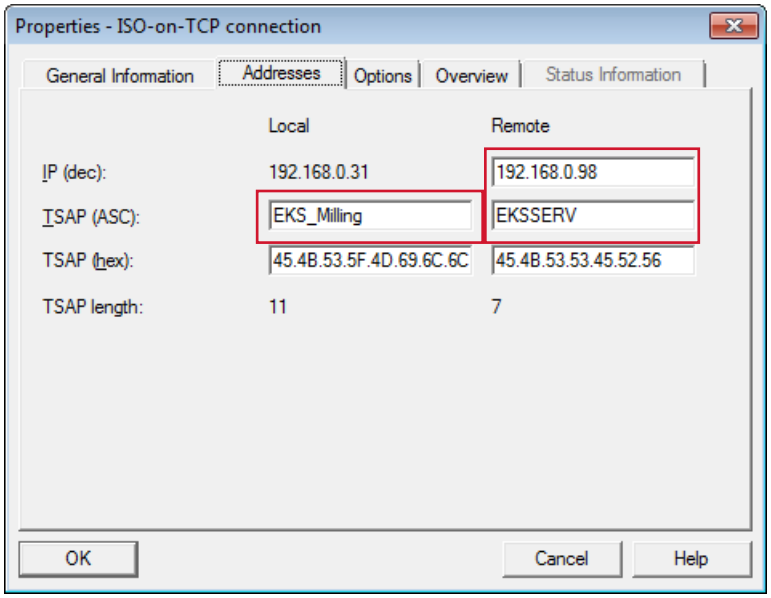


Figure 51: Assigning *TSAP*

NOTICE!

The steps 4 to 7 must be repeated to suit the number of EKS instances. Make sure that each connection is unambiguous (name and *TSAP*).

Two connections have been added for this application.

Local ID	Partner	Type	Active connection	Subnet	Local interface	Local address	Partner address
0001 A050	EKS-Milling	ISO-on-TCP connection	Yes	Ethernet(1) [IE]	PN-IO-1	192.168.0.31	192.168.0.98
0002 A050	EKS-Turning	ISO-on-TCP connection	Yes	Ethernet(1) [IE]	PN-IO-1	192.168.0.31	192.168.0.98

Figure 52: Connections added

EN

8.2.5. Customizing the block *DB_EKSData1*

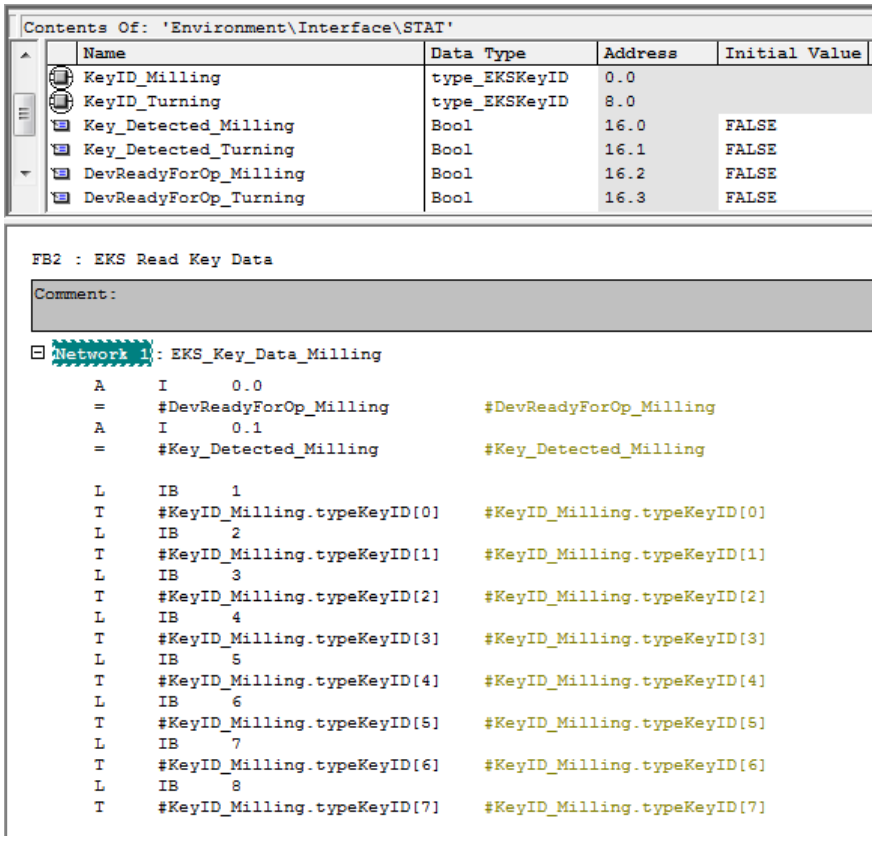
Variable	Data type	Description
Data_Key_Reader_Milling *Data_Key_Reader_Turning*	*type_EKSDatabase*	The data for the Electronic-Key currently placed in the EKS are saved in this variable
Description	*type_EKSDescription*	The structure of the data type *type_EKSDatabase* is provided to the FB in this variable
ID	*Array[0..100] of 'type_EKSDataID'*	The *ID* variable is required to store additional information such as the time stamp for Electronic-Key placement or the time stamp for the last update, the KeyID and the checksum for the EKS Data Service. This additional information is entered and stored in the DB for each Electronic-Key placed. This data type is not allowed to be changed and is saved as an array in the DB. An ID array of the same size is required for each data saving operation (Data Array). Since the data are required after a power failure/restart as well, they must be marked as remanent.
Data	*Array[0..100] of 'type_EKSDatabase'*	The same data type for data from the Electronic-Keys placed is also used for saving the data for the backup strategy. All data from the placed Electronic-Keys are entered in the data array. The data will not be overwritten until the number of Electronic-Keys placed exceeds the number of array entries. If data are overwritten, the oldest Electronic-Key entry is always overwritten first. If an Electronic-Key is not listed during the query to the EKS Data Service PC, the Electronic-Key is also deleted in the data array. The instances of the *FB_EKSDataService* can all access the same data (*Description, ID, Data*) in *DB_EKSData1*. If the data are to be available after a power failure as well, they must be marked as remanent.
EKS_StatusError_Milling *EKS_StatusError_Turning*	*EKS_StatusError*	A data type *EKS_StatusError* has been added for this application. The errors and data this type contains are described in the EKS Data Service PLC manual.

Address	Name	Type	Initial val	Comment
0.0		STRUCT		
+0.0	Data_Key_Reader_Milling	"type_EKSDatabase"		
+118.0	ErrorStatus_Milling	"EKS_StatusError"		
+128.0	Data_Key_Reader_Turning	"type_EKSDatabase"		
+246.0	ErrorStatus_Turning	"EKS_StatusError"		
+256.0	Description	"type_EKSDescription"		
+276.0	ID	ARRAY[0..20]		
*26.0		"type_EKSDataID"		
+822.0	Data	ARRAY[0..20]		
*118.0		"type_EKSDatabase"		
=3300.0		END_STRUCT		

Figure 53: Configuring the parameters for *DB_EKSData1*

8.2.6. Creating an *EKSReadKeyData* block

The values for the Electronic-Key necessary for this application and the status information for each EKS are retrieved in the following. For this purpose a separate FB (*FB_EKSReadKeyData*) is created that is then called in a further FB (*FB_CallEKS-DataService*) (see chapter 8.2.7).



Contents Of: 'Environment\Interface\STAT'

Name	Data Type	Address	Initial Value
KeyID_Milling	type_EKSKeyID	0.0	
KeyID_Turning	type_EKSKeyID	8.0	
Key_Detected_Milling	Bool	16.0	FALSE
Key_Detected_Turning	Bool	16.1	FALSE
DevReadyForOp_Milling	Bool	16.2	FALSE
DevReadyForOp_Turning	Bool	16.3	FALSE

```
FB2 : EKS Read Key Data
Comment:
Network 1: EKS_Key_Data_Milling
      A     I      0.0
      =     #DevReadyForOp_Milling          #DevReadyForOp_Milling
      A     I      0.1
      =     #Key_Detected_Milling           #Key_Detected_Milling

      L     IB     1
      T     #KeyID_Milling.typeKeyID[0]     #KeyID_Milling.typeKeyID[0]
      L     IB     2
      T     #KeyID_Milling.typeKeyID[1]     #KeyID_Milling.typeKeyID[1]
      L     IB     3
      T     #KeyID_Milling.typeKeyID[2]     #KeyID_Milling.typeKeyID[2]
      L     IB     4
      T     #KeyID_Milling.typeKeyID[3]     #KeyID_Milling.typeKeyID[3]
      L     IB     5
      T     #KeyID_Milling.typeKeyID[4]     #KeyID_Milling.typeKeyID[4]
      L     IB     6
      T     #KeyID_Milling.typeKeyID[5]     #KeyID_Milling.typeKeyID[5]
      L     IB     7
      T     #KeyID_Milling.typeKeyID[6]     #KeyID_Milling.typeKeyID[6]
      L     IB     8
      T     #KeyID_Milling.typeKeyID[7]     #KeyID_Milling.typeKeyID[7]
```

Figure 54: Retrieving data from the EKS

NOTICE!

A variable of data type *type_EKSKeyID* must be used for the Electronic-Key's serial number. This variable is used to query the data on the server.

EN

8.2.7. Calling the function blocks *FB_EKSReadKeyData* and *FB_EKSDataService*

Call the function block *FB_EKSReadKeyData* and *FB_EKSDataService* in the FB (*FB_CallEKSDataService)*. This block *FB_EKSDataService* must be called and the parameters configured for each individual EKS. Then call the function block *FB_EKS_Euchner* in the main program *OB1* (*CYCL_EXC*).

NOTICE!

The inputs and outputs of the *FB_EKSDataService* are described in detail in the EKS Data Service PLC manual.

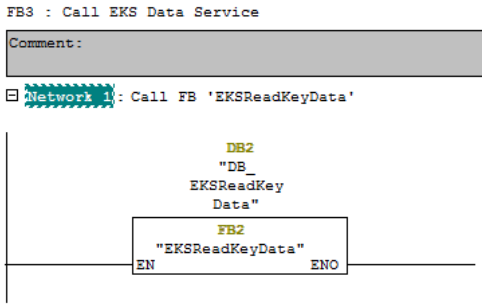


Figure 55: Calling *FB_EKSReadKeyData*

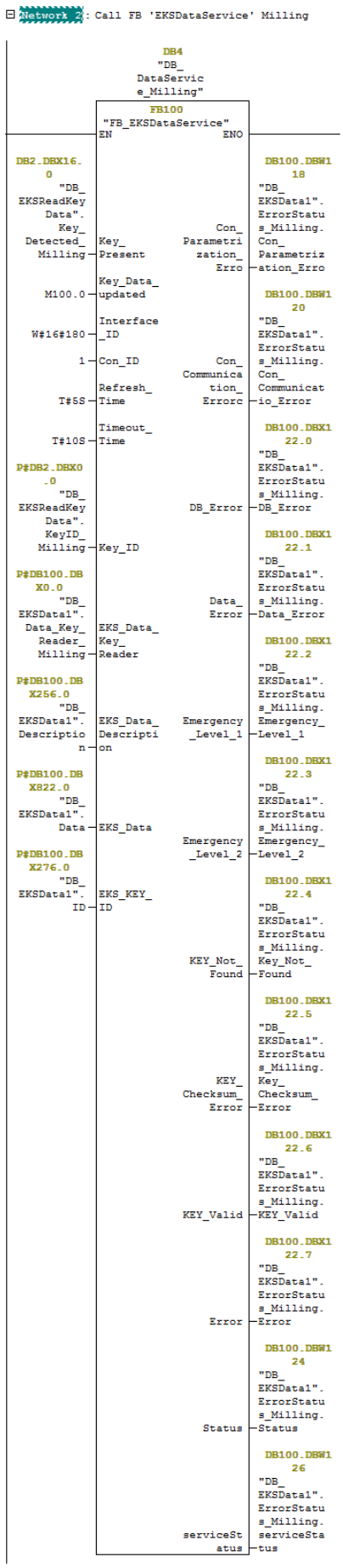


Figure 56: Calling *FB_EKSDataService* (for example: *Milling*)

EN

9. Important note – please observe carefully!

This document is intended for a design engineer who possesses the requisite knowledge in safety engineering and knows the applicable standards, e.g. through training for qualification as a safety engineer. Only with the appropriate qualification is it possible to integrate the example provided into a complete safety chain.

The example represents only part of a complete safety chain and does not fulfill any safety function on its own. In order to fulfill a safety function, the energy switch-off function for the danger zone and the software within the safety evaluation must also be considered, for example.

The applications provided are only examples for solving certain safety tasks for protecting safety doors. The examples cannot be comprehensive due to the application-dependent and individual protection goals within a machine/installation.

If questions concerning this example remain open, please contact us directly.

According to the Machinery Directive 2006/42/EC, the design engineer of a machine or installation has the obligation to perform a risk assessment and take measures to reduce the risk. While doing this, the engineer must comply with the applicable national and international safety standards. Standards generally represent the current state-of-the-art. Therefore, the design engineer should continuously inform himself about changes in the standards and adapt his considerations to them. Relevant standards for functional safety include EN ISO 13849 and EN 62061. This application must be regarded only as assistance for the considerations about safety measures.

The design engineer of a machine/installation has the obligation to assess the safety technology him/herself. The examples must not be used for an assessment, because only a small excerpt of a complete safety function was considered in terms of safety engineering here.

In order to be able to use the safety switch applications correctly on safety doors, it is indispensable to observe the standards EN ISO 13849-1, EN ISO 14119 and all relevant C-standards for the respective machine type. Under no circumstances does this document replace the engineer's own risk assessment, and it cannot serve as the basis for a fault assessment.

In particular in relation to a fault exclusion, it must be noted that a fault can only be excluded by the machine's or installation's design engineer and this action requires justification. A general fault exclusion is not possible. More information about fault exclusion can be found in EN ISO 13849-2.

Changes to products or within assemblies from third-party suppliers used in this example can lead to the function no longer being ensured or the safety assessment having to be adapted. In any event, the information in the operating instructions on the part of EUCHNER, as well as on the part of third-party suppliers, must be used as the basis before this application is integrated into an overall safety function. If contradictions should arise between the operating instructions and this document, please contact us directly.

Use of brand names and company names

All brand names and company names stated are the property of the related manufacturer. They are used only for the clear identification of compatible peripheral devices and operating environments in relation to our products.

Euchner GmbH + Co. KG
Kohlhammerstraße 16
70771 Leinfelden-Echterdingen
info@euchner.de
www.euchner.com

Edition:
AP000248-01-10/19
Title:
Application EKS
Example integration of EKS Data Service