

Inhalt

1.	Allgemeine Hinweise	4
1.1.	Verwendung des Handbuchs	4
1.2.	Voraussetzung an den Anwender	4
1.3.	Systemvoraussetzungen	4
2.	Allgemeine Funktionen der Anwendung	5
3.	Verwendungszweck	6
4.	S7-1200 / S7-1500 TIA Portal	7
4.1.	Bibliothek im TIA Portal einbinden	7
4.2.	Bibliothek EKSDDataService	7
4.2.1.	EKSDDataServiceBase	7
4.2.2.	EKSDDataServiceExample	8
4.3.	EKSDDataService Bausteinbeschreibung	10
4.3.1.	DB_EKSDData1	10
4.3.2.	Daten platzierter Schlüssel (type_EKSDatabase)	10
4.3.3.	FB_EKSDDataService	13
5.	S7-300 / S7-400 TIA Portal	19
5.1.	Bibliothek im TIA Portal einbinden	19
5.2.	Bibliothek EKSDDataService	19
5.2.1.	Verwendete Standard Siemens Bausteine	19
5.2.2.	Baustein FB_EKSDDataService	20
5.2.3.	Baustein DB_EKSDData1	20
5.2.4.	Datentypen type_EKSxxxx	20
5.2.5.	Baustein FC_EKS_Euchner	20
5.3.	Einstellungen (TCP/IP Konfiguration)	22
5.3.1.	Verbindung in der CPU einrichten	22
5.4.	EKSDDataService Bausteinbeschreibung	24
5.4.1.	DB_EKSDData1	24
5.4.2.	FB_EKSDDataService	29
6.	S7-300 / S7-400 SIMATIC STEP7	35
6.1.	Bibliothek in SIMATIC STEP7 einbinden	35
6.2.	Bibliothek EKSDDataService	35
6.2.1.	Verwendete Standard Siemens Bausteine	35
6.2.2.	Verwendete Systembausteine	36
6.2.3.	Baustein FB_EKSDDataService	36
6.2.4.	Baustein DB_EKSDData1	36
6.2.5.	Datentypen type_EKSxxxx	36
6.2.6.	Baustein FC_EKS_Euchner	37
6.3.	Einstellungen NetPro (TCP/IP Konfiguration)	38
6.3.1.	Einrichtung EKS Data Service PC	38
6.4.	EKS Data Service Bausteinbeschreibung	42
6.4.1.	DB_EKSDData1	42
6.4.2.	FB_EKSDDataService	46

7. Statusmeldungen..... 51

7.1. Verbindungsaufbau (Con_Parametrization_Errorcode)51

7.2. Datenempfang (Con_Communication_Errorcode).....53

7.3. Status des FB_EKSDataService (Status)54

7.4. Status der Datenanfrage (serviceStatus).....54

1. Allgemeine Hinweise

1.1. Verwendung des Handbuchs

Dieses Handbuch erläutert die Funktionen der EKS Data Service PLC Software-Komponenten als Bestandteil des EUCHNER EKS Data Service Integrationskits (Best. Nr. 8163316).

1.2. Voraussetzung an den Anwender

Für die sachgerechte Verwendung des EKS Data Service PLC müssen Sie über Vorkenntnisse im Bereich SPS Programmieren unter Verwendung von SIMATIC Manager STEP7 bzw. TIA Portal besitzen. Weiter sollten Sie über Vorkenntnisse in der Konfiguration von Netzwerken verfügen.

1.3. Systemvoraussetzungen

Folgende Siemens Steuerungen und Projektierungssoftware werden unterstützt:

Hardware:	SIMATIC S7-1200
	SIMATIC S7-1200 + CP1243
	SIMATIC S7-1500
	SIMATIC S7-1500 + CP1543
	SIMATIC S7-300 + CP343
	SIMATIC S7-400 + CP443
Software:	SIMATIC S7-300, S7-400, S7-1200, S7-1500 – ab TIA Portal V14 SP1
	SIMATIC S7-300, S7-400 – ab SIMATIC STEP7 V5.5

2. Allgemeine Funktionen der Anwendung

In EKS Anwendungen der SPS-Welt ist ein Datenabgleich bzw. ein Datenabruf aus einer zentralen Datenbank heute meist nicht etabliert. Das bedeutet, dass die Daten fast immer dezentral vom Schlüssel gelesen und dann individuell in der SPS weiterverarbeitet werden. Damit wird die Zugriffs-Information in diesem Fall ausschließlich über den Schlüssel transportiert. Hier besteht das Anliegen vieler EKS Betreiber einen Datenabgleich aus der SPS-Welt mit EKM Daten in der PC-Welt zu realisieren. Das Kernthema dabei ist der Wunsch Schlüssel zentral sperren und ggf. weitere Daten abrufen zu können.

Der Export des Electronic-Key-Manager EKM Datenbankinhalts erfolgt in eine universell nutzbare Datei im CSV-Format in der PC-Umgebung. In dieser EKM CSV Exportdatei kann hinter der Seriennummer (KeyID) des Schlüssels ein Sperrkennzeichen ausgewertet werden, welches auf Wert ,1' gesetzt ist, sobald der Schlüssel gesperrt ist.

Weitere, der Schlüssel Seriennummer zugeordnete Datenelemente können ebenfalls abgeholt werden. Diese Datenelemente sind anwendungsspezifisch unterschiedlich.

Wird ein Schlüssel platziert, wird eine Anfrage vom EKS Data Service PLC an den EKS Data Service PC gesendet. Der EKS Data Service PC sucht anhand der KeyID den Eintrag in der EKM CSV Exportdatei und schickt die Daten an den EKS Data Service PLC zurück. Dort stehen nun die Daten dem Benutzer für die weitere Verarbeitung zu Verfügung. Weiter werden die angefragten Daten in einen Notlaufspeicher abgelegt. Sollte die Verbindung zum PC abbrechen, werden die Daten von zuvor platzierten Schlüsseln aus dem Speicher der Notlaufstrategie geladen. In dem Speicher der Notlaufstrategie stehen die letzten maximal 100 Schlüsseldaten (Größe des Notlaufspeichers kann angepasst werden).

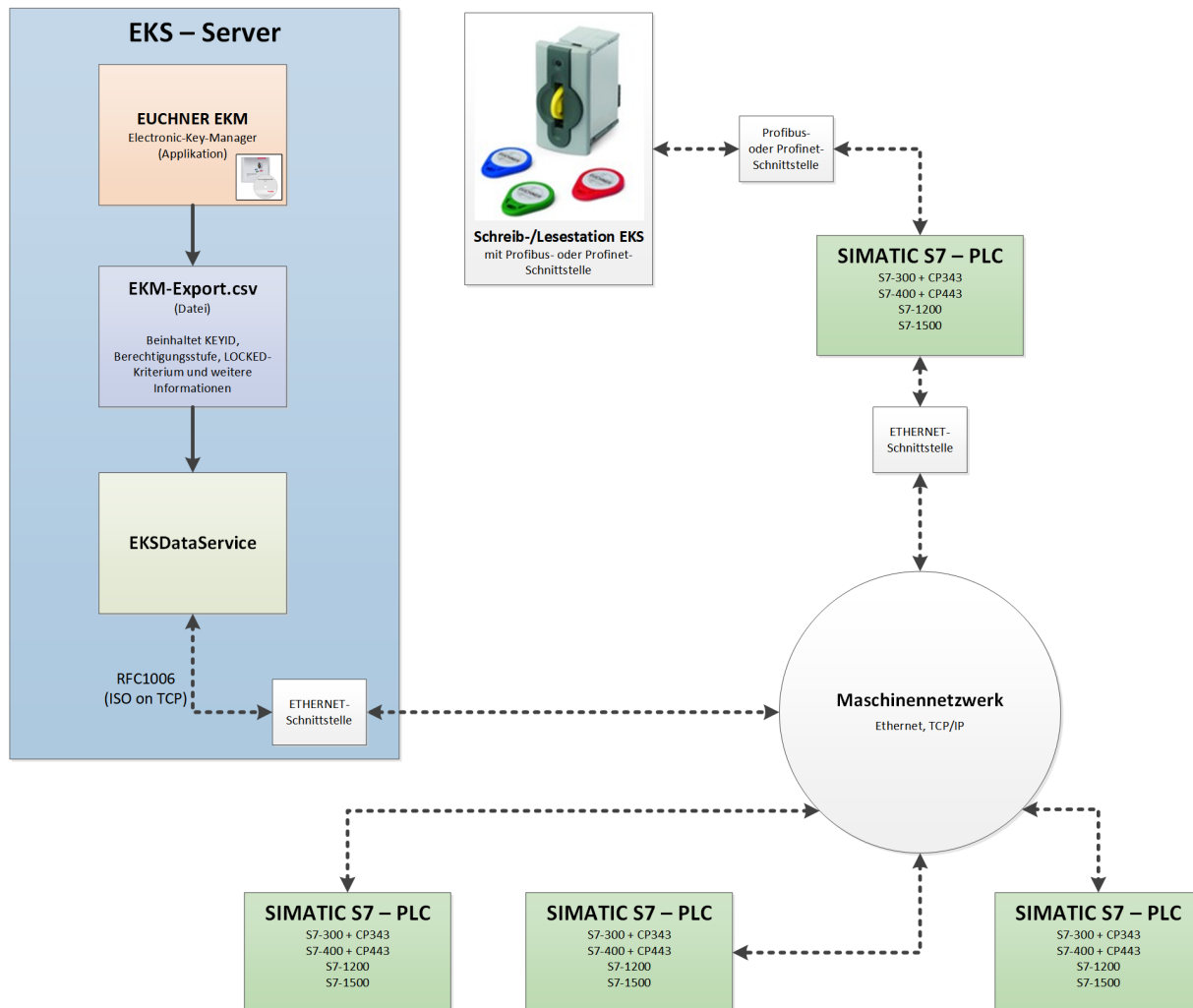
Hinweis zur Notlaufstrategie: Sollte der Speicher der Notlaufstrategie voll sein, und ein neuer Schlüssel wird angefragt, so wird der am längsten nicht abgefragte Schlüssel ersetzt. Da die Einträge der Notlaufstrategie zyklisch aktualisiert werden, wird nicht zwingend der erste Schlüssel der platziert wurde ersetzt.

Es gibt 2 Notlaufstrategien. Bei Notlaufstrategie 1 kann der EKS Data Service PC die original EKM CSV Exportdatei nicht erreichen und verwendet die lokale Backupdatei der EKM CSV Exportdatei. Bei Notlaufstrategie 2 kann der EKS Data Service PC keine Daten zur Verfügung stellen, weil die Kommunikation gestört ist und es werden die Daten des internen SPS Datenbausteins verwendet.

3. Verwendungszweck

Der Dienst „**EUCHNER EKS Data Service PC**“ wird auf einem Server installiert und dient der Verteilung von EKS Schlüssel-daten an SIMATIC Steuerungen, die mit diesem Server über Ethernet TCP/IP vernetzt sind. Die EKS Schlüssel-daten werden durch den EUCHNER Electronic-Key-Manager EKM erzeugt.

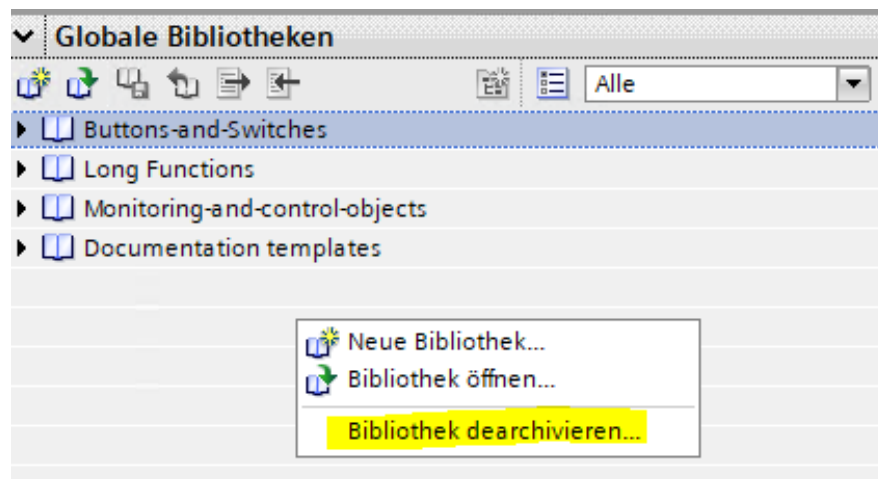
Die Kurzbezeichnung lautet **EKSDataService**. Mit Hilfe von SPS Bausteinen, die in der EKSDatService Bibliothek für S7-300, S7-400, S7-1200 und S7-1500 bereitgestellt werden, wird eine Verbindung von der SPS zum Dienst aufgebaut und Schlüssel-daten des platzierten Schlüssels angefragt.



4. S7-1200 / S7-1500 TIA Portal

4.1. Bibliothek im TIA Portal einbinden

Um die Bibliothek in Ihrem Projekt verwenden zu können, müssen Sie die Bibliothek dearchivieren. Machen Sie im Bereich der Globalen Bibliotheken einen Rechtsklick und wählen Sie *Bibliothek dearchivieren...*



Navigieren Sie anschließend zur Bibliothek `PLC_Library\TIA_1200_1500\EKS_Data_Service_PLC_Library_TIA_1200_1500_20190227.za14` und wählen Sie diese aus. Anschließend geben Sie den Speicherort der dearchivierten Bibliothek an. Falls Sie eine höhere Version als TIA Portal V14 SP1 haben, werden Sie gefragt, ob Sie die Bibliothek hochrüsten möchten (dieser Vorgang ist mit TIA Portal V15 erfolgreich getestet worden). Anschließend befindet sich die Bibliothek zur Auswahl im Bereich *Globale Bibliotheken*.

4.2. Bibliothek EKSDatService

In der Bibliothek EKSDatService befinden sich zwei Ordner in den Kopiervorlagen.

1. EKSDatServiceBase
2. EKSDatServiceExample

4.2.1. EKSDatServiceBase

In diesem Ordner befinden sich für den EKS Data Service PLC notwendigen Bausteine und Datentypen.

Baustein FB_EKSDatService

Der Baustein FB_EKSDatService ist für den Datenaustausch mit dem EKS Data Service PC und der Bereitstellung der Daten in der SPS zuständig.

Baustein DB_EKSDat1

Der Datenbaustein DB_EKSDat1 ist ein Beispiel für den Aufbau der notwendigen Daten für den FB_EKSDatService.

Datentypen (UDTs) type_EKSxxxx

Diese Datentypen sind notwendig für die Deklaration von Bausteinparametern und Datentypen im DB.

- type_EKSDatabase
- type_EKSDatID
- type_EKSDescription
- type_EKSKeyID

4.2.2. EKSDataServiceExample

In diesem Ordner befindet sich ein komplettes Beispiel für die Verwendung des EKS Data Service PLC. Es werden zwei EKS Lesegeräte (PROFINET, PROFIBUS) verwendet. Für jedes EKS wurde eine Instanz des FB_EKSDataService angelegt. Die vom FB_EKSDataService angefragten Daten werden in einem Datenbaustein DB_EKSData1 gespeichert.

Baustein EKS_Euchner

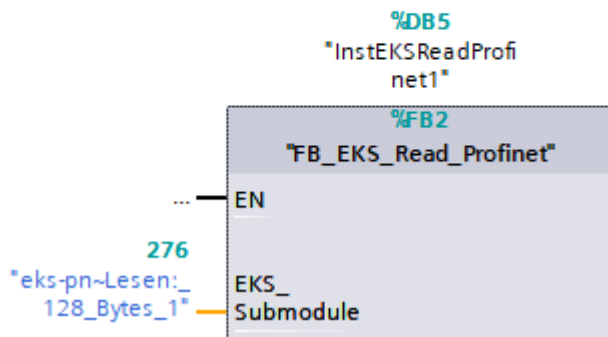
Der Baustein wird im Hauptprogramm Main (OB1) aufgerufen und verwaltet alle Euchner Bausteine im Zusammenhang mit dem EKS Data Service PLC.

Im Beispiel werden Hardware-Adressen verwendet, die in der Gerätekonfiguration mit der Deklaration der Lesegeräte erzeugt werden. Die richtigen Adressen können nach der Projektierung der Hardware im Konfigurationsteil entnommen werden.

Adressverweis FB_EKS_Read_Profinet

Netzwerk 1: EKS Data read Profinet 1

Kommentar



Netzübersicht					Verbindungen	E/A-Kommunikation	VPN	TeleControl
Gerät	Typ		Adresse im Subn..	Subnetz				
▼ GSD-Geraet_1	GSD-Geraet							
▶ eks-pn	EKS-A-IIXA-G01-ST02/0...							
▼ S71500/ET200MP-Station_1	S71500/ET200MP-Station							
▶ PLC_1	CPU 1516F-3 PN/DP							
▼ GSD-Geraet_2	GSD-Geraet							
eks	EKS-A-IDX-G01-ST09/03		3	PROFIBUS_1				

GSD-Geraet_1 [Device]			
Allgemein	IO-Variablen	Systemkonstanten	Texte
Hardware-Systemkonstanten anzeigen			
Name	Typ	HW-Kennung	
eks-pn~IODevice	Hw_Device	270	
eks-pn~Interface~Port_1	Hw_Interface	274	
eks-pn~Interface	Hw_Interface	273	
eks-pn~Proxy	Hw_SubModule	272	
eks-pn~Head	Hw_SubModule	275	
eks-pn~Lesen:_128_Bytes_1	Hw_SubModule	276	
eks-pn~Schreiben:_128_Bytes_1	Hw_SubModule	277	

Adressverweis FB_EKS_Read_Profibus

Falls ein EKS Profibus zum Einsatz kommt, enthält die Bibliothek einen Beispielbaustein (FB_EKS_Read_Profibus) zum Auslesen eines Profibus EKS. Für die Konfiguration des Bausteins muss der Eingangsadressbereich des Profibus EKS konfiguriert werden.

GETIO_PART

EN

"eks~Read_Write:_128_120_Byte_I_O_4_1"

ID

#instGetIoPartK
ID

GETIO_PART

EN

"eks~Read_Write:_128_120_Byte_I_O_4_4"

ID

EKS_DataService ▶ Geräte & Netze

Netzübersicht

Verbindungen

E/A-Kommunikation

VPN

Gerät	Typ	Adresse im Subn.
▶ CPU1200	CPU 1215C DC/DC/DC	
▼ GSD-Geraet_1	GSD-Geraet	
▼ eks-pn	EKS Profinet modular F...	
▶ Interface	eks-pn	192.168.12.1
▼ GSD-Geraet_3	GSD-Geraet	
eks	EKS-A-IDX-G01-ST09/03	3
▼ GSD-Geraet_2	GSD-Geraet	

GSD-Geraet_3 [Device]

Allgemein

IO-Variablen

Systemkonstanten

Texte

Hardware-Systemkonstanten anzeigen ▼

Name	Typ	HW-Kennung
eks~DPSlave	Hw_DpSlave	267
eks~Head	Hw_Interface	269
eks~Read_Write:_128_120_Byte_I_O_4_1	Hw_SubModule	270
eks~Read_Write:_128_120_Byte_I_O_4_2	Hw_SubModule	271
eks~Read_Write:_128_120_Byte_I_O_4_3	Hw_SubModule	272
eks~Read_Write:_128_120_Byte_I_O_4_4	Hw_SubModule	273

4.3. EKSDataService Bausteinbeschreibung

Die Steuerung benötigt zwei Bausteine für das Arbeiten mit dem EKS Data Service PC.

Der erste Baustein ist ein FB (FB_EKSDataService) in welchem alle Funktionalitäten programmiert sind. Der zweite Baustein ist ein DB (DB_EKSData1) in welchem die Daten für die Notlaufstrategie, die aktuellen Daten für den platzierten Schlüssel und die Deklaration der Daten liegen.

4.3.1. DB_EKSData1






























Alle Daten, die hier abgelegt sind, können auch auf andere Datenbausteine verteilt werden. In der Vorlage sind alle notwendigen Daten in einem DB vereint.

4.3.2. Daten platzierter Schlüssel (type_EKSDatabase)

Der Datentyp *type_EKSDatabase* beschreibt die Struktur der abgefragten Daten von der Datenbank.

Alle Schlüsseldaten, die in der SPS von der EKM Datenbank benötigt werden, müssen hier in einer Struktur festgelegt werden. Es ist notwendig, dass die Struktur mit der Beschreibung *type_EKSDescription* übereinstimmt (siehe hierzu auch das Software Handbuch EKS Data Service PC, Kapitel 7.2.2).

Beispiel aus der Bibliothek:

type_EKSDatabase			
	Name	Datentyp	Defaultwert
	KeyID	String[16]	"
	LOCKED	Bool	false
	Key_Bit	Bool	false
	Key_ShortInt	Byte	16#0
	Key_Byte	Byte	16#0
	Key_SmallInt	Int	0
	Key_Word	Word	16#0
	Key_Integer	DInt	0
	Key_Float	Real	0.0
	Key_String	String[12]	"
	Key_StringBlankFilled	String[10]	"
	Key_Time	Time	T#0ms
	Key_TimeAscii	String[8]	"
	Key_Date	Date	D#1990-01-01
	Key_DateAscii	String[8]	"
	Key_BitString	Word	16#0
	KEYCRC	Word	16#0
	DB_Bit	Bool	false
	DB_ShortInt	Byte	16#0
	DB_Byte	Byte	16#0
	DB_SmallInt	Int	0
	DB_Word	Word	16#0
	DB_Integer	DInt	0
	DB_Float	Real	0.0
	DB_String	String[14]	"
	DB_StringBlankFilled	String[12]	"
	DB_Time	Time	T#0ms
	DB_Date	Date	D#1990-01-01
	DB_BitString	Byte	16#0

Im DB kann für jedes EKS ein eigenes Datenfeld erzeugt werden in welchem die Daten des aktuell platzierten Schlüssels abgelegt werden.

DB_EKSDData1	
Name	Datentyp
Static	
▶ Data_Key_Reader_Profinet	"type_EKSDatabase"
▶ Data_Key_Reader_Profibus	"type_EKSDatabase"

Beschreibung der angefragten Daten (type_EKSDescription)

Die Daten werden beim EKS Data Service PC angefragt. Diese Anfrage wird über eine Struktur definiert, die der Struktur vom Typ `type_EKSDatabase` entspricht (siehe hierzu auch das Software Handbuch EKS Data Service PC Kapitel 7.2.2).

Beispiel aus der Bibliothek:

type_EKSDatabase			
Name	Datentyp	Defaultwert	
KeyID	String[16]	"	
LOCKED	Bool	false	
Key_Bit	Bool	false	
Key_ShortInt	Byte	16#0	
Key_Byte	Byte	16#0	
Key_SmallInt	Int	0	
Key_Word	Word	16#0	
Key_Integer	Dint	0	
Key_Float	Real	0.0	
Key_String	String[12]	"	
Key_StringBlankFilled	String[10]	"	
Key_Time	Time	T#0ms	
Key_TimeAscii	String[8]	"	
Key_Date	Date	D#1990-01-01	
Key_DateAscii	String[8]	"	
Key_BitString	Word	16#0	
KEYCRC	Word	16#0	
DB_Bit	Bool	false	
DB_ShortInt	Byte	16#0	
DB_Byte	Byte	16#0	
DB_SmallInt	Int	0	
DB_Word	Word	16#0	
DB_Integer	Dint	0	
DB_Float	Real	0.0	
DB_String	String[14]	"	
DB_StringBlankFilled	String[12]	"	
DB_Time	Time	T#0ms	
DB_Date	Date	D#1990-01-01	
DB_BitString	Byte	16#0	

Die Defaultwerte sind dabei wichtig und können im EKS Data Service PC eingestellt werden.

Im DB wird die Beschreibung des Typs *EKS Description* einmal einer Variablen zugeordnet:

DB_EKSDData1	
Name	Datentyp
Static	
▶ Data_Key_Reader_Profinet	"type_EKSDatabase"
▶ Data_Key_Reader_Profibus	"type_EKSDatabase"
▶ Description	"type_EKSDescription"

ID Array für die Verwaltung der Daten (type_EKSDatalD)

Die ID wird benötigt, um zusätzliche Informationen wie den Zeitstempel für das Platzieren des Schlüssels oder den Zeitstempel der letzten Aktualisierung, die KeyID und die Checksumme des EKS Data Service zu speichern. Diese zusätzlichen Informationen werden für jeden platzierten Schlüssel eingetragen und im DB abgelegt.

Dieser Datentyp darf nicht verändert werden und wird als Array in den DB abgelegt.

Für jede Datenspeicherung (Data Array) benötigt man ein ID Array mit der gleichen Größe.

Da die Daten auch nach Spannungsausfall/Neustart benötigt werden, ist es notwendig diese als remanent zu markieren.



Hinweis zur Checksumme

Der EKS Data Service berechnet über die angefragten Daten eine Checksumme (im weiteren Text auch kurz „CRC“ genannt) zur Gewährleistung der Integrität der angefragten Daten. Diese Daten werden vom Anwender nicht verarbeitet. Diese Checksumme ist nicht mit der Checksumme zu verwechseln, welche die Electronic-Key-Manager EKM Software berechnet. Die Checksumme des EKM (im weiteren Text auch kurz „Key CRC“ genannt) bezieht sich immer auf einen bestimmten Datenbereich, welcher im EKM konfiguriert werden kann. Die Key CRC des EKM wird auf dem Schlüssel mit abgespeichert und kann ausschließlich auf „On-Key“ Felder angewendet werden. Im Gegensatz dazu berechnet der EKS Data Service über alle angefragten Daten die CRC (sowohl „On-Key“ als auch EKM Datenbank-Werte).

type_EKSDataID		
	Name	Datentyp
	▶ timestampPlaced	DTL
	▶ timestampRefresh	DTL
	▶ KeyID	*type_EKSKeyID*
	CRC	Word

Beispiel aus der Bibliothek:

DB_EKSData1		
	Name	Datentyp
	▼ Static	
	▶ Data_Key_Reader_Profinet	*type_EKSDatabase*
	▶ Data_Key_Reader_Profibus	*type_EKSDatabase*
	▶ Description	*type_EKSDescription*
	▶ ID	Array[0..100] of *type_EKSDataID*

Data Array für die Speicherung der Notlauf-Daten (type_EKSDatabase)

Der gleiche Datentyp für Daten der platzierten Schlüssel, wird auch für die Speicherung der Daten für die Notlaufstrategie verwendet. Alle Daten der platzierten Schlüssel werden in das Data Array eingetragen. Die Daten werden erst überschrieben, wenn die Anzahl der platzierten Schlüssel größer ist, als die Anzahl der Array Einträge. Sollten die Daten überschrieben werden, dann wird immer der älteste Schlüsseleintrag überschrieben. Wird ein Schlüssel bei der Anfrage beim EKS Data Service PC nicht gelistet, dann wird der Schlüssel auch im Data Array gelöscht. Die Instanzen des FB_EKSDataService können alle auf die gleichen Daten (Description, ID, Data) im DB_EKSData1 zugreifen. Sollten die Daten auch nach einem Spannungsausfall zu Verfügung stehen, ist es notwendig diese als remanent zu markieren.

Beispiel aus der Bibliothek:

DB_EKSData1		
	Name	Datentyp
	▼ Static	
	▶ Data_Key_Reader_Profinet	*type_EKSDatabase*
	▶ Data_Key_Reader_Profibus	*type_EKSDatabase*
	▶ Description	*type_EKSDescription*
	▶ ID	Array[0..100] of *type_EKSDataID*
	▶ Data	Array[0..100] of *type_EKSDatabase*

4.3.3. FB_EKSDataService

Der FB_EKSDataService für die S7-1200 / S7-1500 besitzt die folgende Schnittstellenbeschreibung.

INPUT		
Name	Datentyp	Beschreibung
Key_Present	Bool	Schlüssel ist platziert
Key_Data_updated	Bool	Schlüsseldaten werden bei Platzierung erneuert
Interface_ID	HW_ANY	Profinet Schnittstelle Hardware Kennung
Con_ID	CONN_OUC	Eindeutige Kennung der Verbindung
Timeout_Time	Time	Zeitüberschreitung Kommunikation
Refresh_Time	Time	Aktualisierungszeit für Schlüsseldaten
IP_EKSDataService	String[20]	IP Adresse des EKS Daten Service
Local_TSAP	String[32]	TSAP der eigenen Verbindung
Remote_TSAP	String[32]	TSAP des EKS Data Service
OUTPUT		
Name	Datentyp	Beschreibung
Con_Parametrization_Errorcode	Word	Parametrierungsfehler der Kommunikation
Con_Communication_Errorcode	Word	Kommunikationsfehler Server Dienst
DB_Error	Bool	Database Länge nicht richtig
Data_Error	Bool	Datenfehler vom Server Dienst
Emergency_Level_1	Bool	Notlaufstrategie 1 aktiv
Emergency_Level_2	Bool	Notlaufstrategie 2 aktiv
KEY_Not_Found	Bool	Schlüssel ID nicht gefunden
KEY_Checksum_Error	Bool	CRC Fehler (der angefragten Daten)
KEY_Valid	Bool	Schlüssel platziert und Daten verfügbar
Error	Bool	Bausteinfehler
Status	Word	Status des Bausteins
serviceStatus	Word	Status des Server Dienstes
INOUT		
Name	Datentyp	Beschreibung
Key_ID	type_EKSKeyID	Seriennummer des platzierten Schlüssels
EKS_Data_Key_Reader	Variant	Datenablage für den platzierten Schlüssel
EKS_Data_Description	Variant	Daten der Beschreibung
EKS_Data	Variant	Daten Array der SPS Daten
EKS_KEY_ID	Variant	ID Daten für die Datenstruktur
EKS_Data_Temp	Variant	Datenspeicher für Zwischenspeicherung



Hinweis

Der FB_EKSDataService ist durch ein Kennwort geschützt um ungewollte Manipulation zu verhindern. In manchen Fällen ist es notwendig, den FB_EKSDataService erneut zu übersetzen, wenn Sie ihn in Ihr Projekt mit eingebunden haben. Um den Baustein übersetzen zu können, benötigen Sie ein Kennwort. Bitte führen Sie keine Änderungen an der Implementierung des FB_EKSDataService durch.

Kennwort: EKS1234

DE

Key_Present

Beim Lesen des platzierten Schlüssels vom EKS Lesegerät wird ein Signal erzeugt, das die gelesene Schlüssel Seriennummer bestätigt. Dieses Signal ist für die Anfrage der Schlüsseldaten erforderlich. Wechselt Key_Present von FALSE auf TRUE, wird eine Datenanfrage an den EKS Data Service PC gesendet.

Key_Data_updated

Wird der Parameter mit TRUE beschaltet, dann werden die Schlüsseldaten des platzierten Schlüssels immer mit der Datenbank aktualisiert. Dies bedeutet, dass sich die Daten während der Platzierung ändern können.

Beispiel:

Ein Schlüssel ist im EKS platziert und die angefragten Daten sind bereits in der SPS hinterlegt (DB_EKSData1.Data_Key_Reader_Profinet) und die Datenaktualisierung (Refresh_Time) ist aktiviert. D.h. die Daten der Notlaufstrategie werden im Hintergrund zyklisch aktualisiert. Sollten sich der aktualisierte Wert des platzierten Schlüssels und der bereits hinterlegte Wert unterscheiden, so wird bei Key_Data_updated = TRUE der Wert des aktuell platzierten Schlüssel (DB_EKSData1.Data_Key_Reader_Profinet) überschrieben. Andernfalls wird der aktualisierte Wert erst bei erneutem Schlüssel platzieren geladen.

Interface_ID

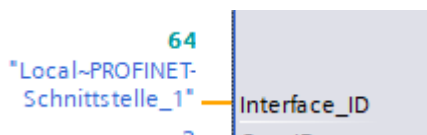
Hardware Kennung der lokalen Schnittstelle über welche mit dem EKS Data Service PC kommuniziert werden soll.



Hinweis

Mehrere Instanzen des FB_EKSDataService können über die selbe Schnittstelle kommunizieren.

Name	Typ	HW-Kennung
Local~PROFINET-Schnittstelle_1~Port_1	Hw_Interface	65
Local~PROFINET-Schnittstelle_1~Port_2	Hw_Interface	66
Local~PROFINET-Schnittstelle_1	Hw_Interface	64



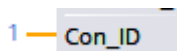
Con_ID

Die Verbindungs-ID muss für jede eigene Verbindung innerhalb einer SPS eindeutig sein. Dies ist auch erforderlich bei der Verwendung von mehreren Instanzen.

Beispiel:

Instanz A = 1

Instanz B = 2



Timeout_Time

Bei einer Zeitüberschreitung während der Kommunikation mit dem EKS Data Service PC, wird ein Fehler erzeugt, der die Kommunikation abbricht und einen Error mit Status ausgibt. Im Fehlerfall wird die Notlaufstrategie 2 ausgelöst und die angefragten Daten werden aus dem Speicher der Notlaufstrategie (DB_EKSData1.Data) geladen.

t#200ms → Timeout_Time

Refresh_Time

In diesem Zeittakt werden die Daten der gelisteten Schlüssel im DB_EKSData1.Data vom EKS Data Service PC abgeholt und geprüft, ob diese aktuell sind. Andernfalls werden die Werte im DB_EKSData1.Data aktualisiert. D.h. bereits angefragte Daten werden im Hintergrund ohne Platzierung des Schlüssels aktualisiert. Der Baustein geht nach jedem Takt zum nächsten Eintrag und fragt die Daten beim EKS Data Service PC an. Bei einer Einstellung von 10 s benötigt der Dienst 100 s für eine Abfrage von 10 gespeicherten Schlüsseldaten.

Die Aktualisierung der Daten kann über den Wert 0ms abgeschaltet werden. Werden mehrere Instanzen des FB_EKSDataService benötigt und greifen alle Instanzen auf die gleichen Daten zu, dann muss nur eine Instanz die Aktualisierung vornehmen.

Beispiel:

Instanz A: Refresh_Time = 10 s (sorgt für die Aktualisierung)

Instanz B: Refresh_Time = 0 ms (Aktualisierung ausgeschaltet)

Instanz C: Refresh_Time = 0 ms (Aktualisierung ausgeschaltet)

t#10s → Refresh_Time

IP_EKSDataService

Hier wird die IP-Adresse vom EKS Data Service PC eingetragen.

'192.168.12.101' → IP_EKSDataService

Die IP-Adresse wird als String mit Hochkomma eingetragen. ('192.168.12.101')

Local_TSAP

Hier wird die eigene TSAP Adresse eingetragen. Diese wird für die Identifizierung der Kommunikation verwendet und wird im EKS Data Service PC angezeigt. Für mehrere Instanzen ist es notwendig unterschiedliche Beschreibung zu verwenden. Dies ist auch hilfreich für das Debugging im EKS Data Service PC, hier wird der Name für den Log-Eintrag verwendet.

'EKSCLIENT1500_Profinet' → Local_TSAP Zum Beispiel für den Profinet Leser

'EKSCLIENT_1500_Profibus' → Local_TSAP oder Profibus Leser

Der TSAP wird als String mit Hochkomma eingetragen. ('EKSCLIENT1500_Profinet')

Remote_TSAP

Der Remote TSAP ist eine feste Adresse für den EKS Data Service PC. Diese muss immer wie folgt eingetragen werden:

'EKSSERV' → Remote_TSAP

Der TSAP wird als String mit Hochkomma eingetragen. ('EKSSERV')

Con_Parametrization_Errorcode

Wird beim Verbindungsaufbau ein Fehler erzeugt, wird der entsprechende Fehlercode hier eingetragen. Eine Liste der Fehlercodes finden Sie im Kapitel 7.1. *Verbindungsaufbau (Con_Parametrization_Errorcode)*.

Con_Communication_Errorcode

Wird beim Empfangen der Daten ein Fehler erzeugt, wird der entsprechende Fehlercode hier eingetragen. Eine Liste der Fehlercodes finden Sie im Kapitel 7.2. *Datenempfang (Con_Communication_Errorcode)*.

DB_Error

Das Fehler-Bit wird TRUE, wenn die Länge der Struktur *type_EKSDatabase* nicht mit der gesendeten Datengröße vom EKS Data Service PC übereinstimmt.

Data_Error

Der Fehler *Data_Error* wird durch den EKS Data Service PC ausgegeben, wenn die vorgegebene Datenstruktur nicht mit der Struktur vom EKS Data Service PC übereinstimmt. Meldet der EKS Data Service PC diesen Fehler, dann wird *Data_Error* auf TRUE gesetzt.

Emergency_Level_1

Notlaufstrategie 1, der EKS Data Service PC kann die original EKM CSV Exportdatei nicht erreichen und verwendet die lokale Backupdatei der EKM CSV Exportdatei.

Emergency_Level_2

Notlaufstrategie 2, der EKS Data Service PC kann keine Daten zur Verfügung stellen, weil die Kommunikation gestört ist und es werden die Daten des internen Datenbausteins verwendet (*DB_EKSData1.Daten*).

KEY_Not_Found

Der platzierte Schlüssel wurde nicht gefunden. Diese Information kann vom EKS Data Service PC kommen oder bei Notlaufstrategie 2 von der SPS, wenn der angefragte Schlüssel nicht im Speicher (*DB_EKSData1.Data*) hinterlegt ist.

KEY_Checksum_Error

Die CRC welche über die angefragten Daten berechnet wurde, stimmt nicht überein. Dieser Fehler wird ausgegeben wenn die CRC der gesendeten Daten nicht übereinstimmt oder im Falle der Notlaufstrategie 2, wenn die CRC der gespeicherten Daten nicht korrekt ist.

KEY_Valid

Die Daten des platzierten Schlüssels stehen zur Verfügung, dies gilt auch bei den Notlaufstrategien.

Error

Ein Fehler bei der Bausteinbearbeitung ist aufgetreten. Der Fehler wird bei der nächsten Bearbeitung gelöscht und ggf. wieder ausgegeben.

Status

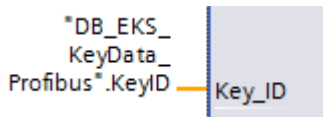
Statusmeldungen welche innerhalb der Verarbeitung im *FB_EKSDataService* generiert werden. Eine Liste der Statusmeldungen finden Sie im Kapitel 7.3. *Status des FB_EKSDataService (Status)*.

serviceStatus

Statuscodes welche vom EKS Data Service PC an den *FB_EKSDataService* übergeben werden. Eine Liste der Statusmeldungen finden Sie im Kapitel 7.4. *Status der Datenanfrage (serviceStatus)*.

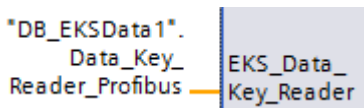
Key_ID

Die Seriennummer des platzierten Schlüssels muss hier angegeben werden, um die Daten vom EKS Data Service PC anzufragen. Der Datentyp `type_EKSKeyID` muss hierzu verwendet werden.



EKS_Data_Key_Reader

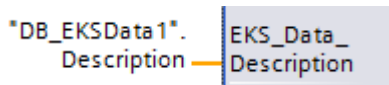
Die Daten des platzierten Schlüssels werden an diese Adresse kopiert. Es muss der Datentyp `type_EKSDatabase` verwendet werden.



► Data_Key_Reader_Profinet	*type_EKSDatabase*
► Data	Array[0..100] of *type_EKSDatabase*

EKS_Data_Description

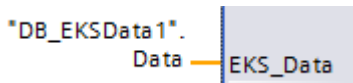
Die Beschreibung der Daten wird als Datentyp Variant übergeben. Diese Datenstruktur wird benutzt um die geforderten Daten vom EKS Data Service PC anzufragen. Es muss die Datenstruktur vom Datentyp `type_EKSDescription` verwendet werden.



► Description	*type_EKSDescription*
---------------	-----------------------

EKS_Data

Das Datenarray für die Datensicherung auf SPS Seite wird als Datentyp Variant angelegt (siehe Data DB_EKSDData1).

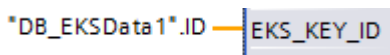


► Data	Array[0..100] of *type_EKSDatabase*
--------	-------------------------------------

Das Datenarray muss die gleiche Größe wie das EKS_KEY_ID Array haben.

EKS_KEY_ID

Für die Verwaltung der Datensicherung gibt es den ID Bereich (Siehe ID DB_EKSDData1).



► ID	Array[0..100] of *type_EKSDataID*
------	-----------------------------------

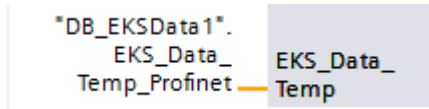
Das Datenarray muss die gleiche Größe wie das EKS_Data Array haben.

EKS_Data_Temp

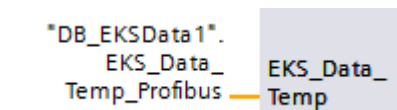
Der Baustein benötigt einen Zwischenspeicher außerhalb des Bausteins vom Typ `type_EKSDatabase`. Der Zwischenspeicher wird für Kopieraufgaben verwendet. Die Struktur wird als Datentyp Variant übergeben und kann je nach Anwendung einen anderen Inhalt haben. Die internen Kopiervorgänge können jedoch nur Varianten kopieren und benötigen einen Speicherplatz mit der gleichen Struktur.

Der Speicherplatz kann auch im DB_EKSData1 erzeugt werden. Wichtig ist hier einen Speicherplatz für jede Instanz.







Instanz Profinet:



Instanz Profibus:



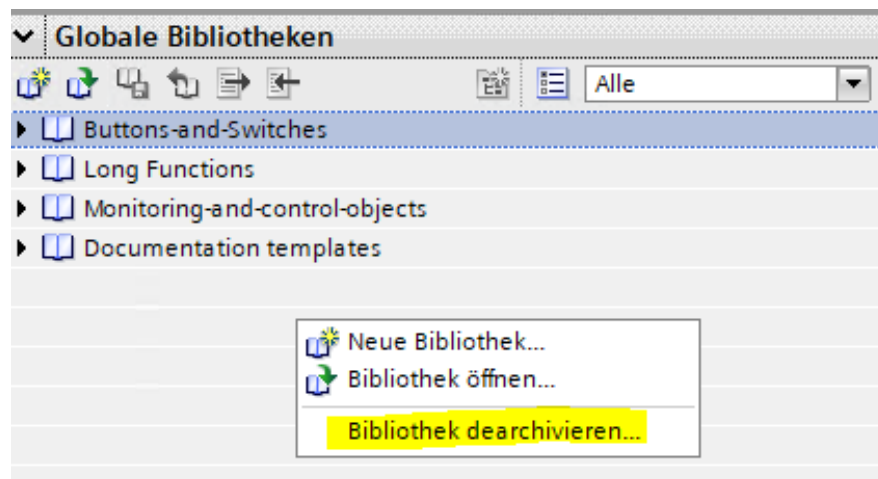
Speicherplatz im DB_EKSData1:

			EKS_Data_Temp_Profinet	"type_EKSDatabase"
			EKS_Data_Temp_Profibus	"type_EKSDatabase"

5. S7-300 / S7-400 TIA Portal

5.1. Bibliothek im TIA Portal einbinden

Um die Bibliothek in Ihrem Projekt verwenden zu können, müssen Sie die Bibliothek dearchivieren. Machen Sie im Bereich der Globalen Bibliotheken einen Rechtsklick und wählen Sie *Bibliothek dearchivieren...*



Navigieren Sie anschließend zur Bibliothek (PLC_Library\TIA_300_400\EKS_Data_Service_PLC_Library_TIA_300_400_20190503.zal14) und wählen Sie diese aus. Anschließend geben Sie den Speicherort der dearchivierten Bibliothek an. Falls Sie eine höhere Version als TIA Portal V14 SP1 haben werden Sie gefragt, ob Sie die Bibliothek hochrüsten möchten (dieser Vorgang ist mit TIA Portal V15 erfolgreich getestet worden). Anschließend befindet sich die Bibliothek zur Auswahl im Bereich *Globale Bibliotheken*.

5.2. Bibliothek EKSDatService

In der Bibliothek EKSDatService befinden sich im Ordner S7-300 alle Bausteine für die CPU 300 und im Ordner S7-400 alle Bausteine für die CPU 400.

5.2.1. Verwendete Standard Siemens Bausteine

Für die Realisierung wurden aus der Siemens Bibliothek verschiedene Bausteine verwendet.

- AG_SEND (AG_LSEND)
- AG_RECV (AG_LRECV)
- EQ_STRNG
- AG_CNTRL
- GT_DT
- MID
- NE_STRNG
- REPLACE
- STRNG_I

Diese Bausteine sind zwingend notwendig und sollten nicht in der Nummerierung geändert werden (Ausnahme bei symbolischem Vorrang). Die Bausteine der CPU 300 und CPU 400 unterscheiden sich von in ihrer Funktion. Es müssen die richtigen Bausteine der CPU verwendet werden.

Beispiel Bausteinfamilie:

CPU 300

Information	
Titel:	<input type="text"/>
Kommentar:	<input type="text"/>
Version:	1.4
Familie:	CP_300
Autor:	SIMATIC
Anwenderdefinierte ID:	AG_CNTRL

CPU 400

Information	
Titel:	<input type="text"/>
Kommentar:	<input type="text"/>
Version:	1.0
Familie:	CP_400
Autor:	SIMATIC
Anwenderdefinierte ID:	AG_CNTRL

5.2.2. Baustein FB_EKSDataService

Der Baustein (FB_EKSDataService) ist für den Datenaustausch mit dem EKS Data Service PC und der Bereitstellung der Daten in der SPS zuständig.

5.2.3. Baustein DB_EKSData1

Der Datenbaustein (DB_EKSData1) ist ein Beispiel für den Aufbau der notwendigen Daten für den FB_EKSDataService.

5.2.4. Datentypen type_EKSxxxx

Die Datentypen UDTs sind notwendig für die Deklaration von Bausteinparametern und Datentypen im DB.

- type_EKSDatabase
- type_EKSDataID
- type_EKSDescription
- type_EKSKeyID

5.2.5. Baustein FC_EKS_Euchner

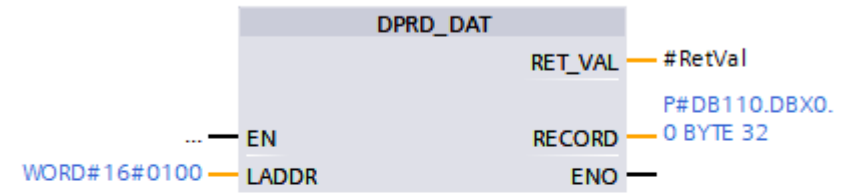
Der Baustein wird im Hauptprogramm Main (OB1) aufgerufen und verwaltet alle Euchner Bausteine im Zusammenhang mit dem EKS Data Service PC.

Im Beispiel werden Hardware-Adressen verwendet, die in der Gerätekonfiguration mit der Deklaration der EKS Lesegeräte erzeugt werden. Die richtigen Adressen können nach der Projektierung der Hardware im Konfigurationsteil entnommen werden.

Adressverweis EKS_Euchner (Profibus)

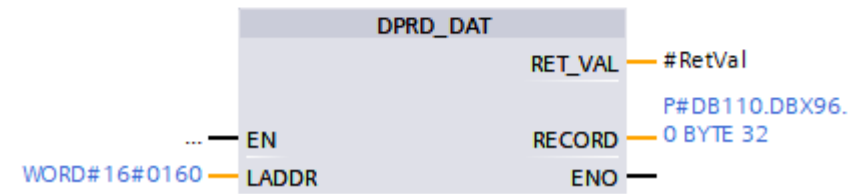
Netzwerk 1: Read EKS Profibus (Statusinformation)

Kommentar



Netzwerk 2: Read EKS Profibus (KeyID)

Kommentar



DX-G01-STC

EKS-Slave

Geräteübersicht

Modul	Baugr...	Steck...	E-Adresse	A-Adres...	Ty
EKS-Slave	0	0	2043*		E
Read/Write: 128/120 Byte I/...	0	1	256...287	256...287	R
Read/Write: 128/120 Byte I/...	0	2	288...319	288...319	R
Read/Write: 128/120 Byte I/...	0	3	320...351	320...351	R
Read/Write: 128/120 Byte I/...	0	4	352...383	352...375	R



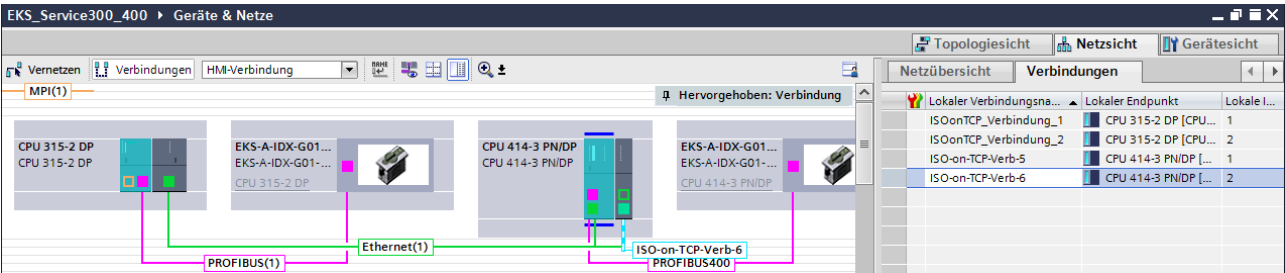
Hinweis

Die E-Adresse muss als HEX Wert an LADDR angegeben werden

5.3. Einstellungen (TCP/IP Konfiguration)

5.3.1. Verbindung in der CPU einrichten

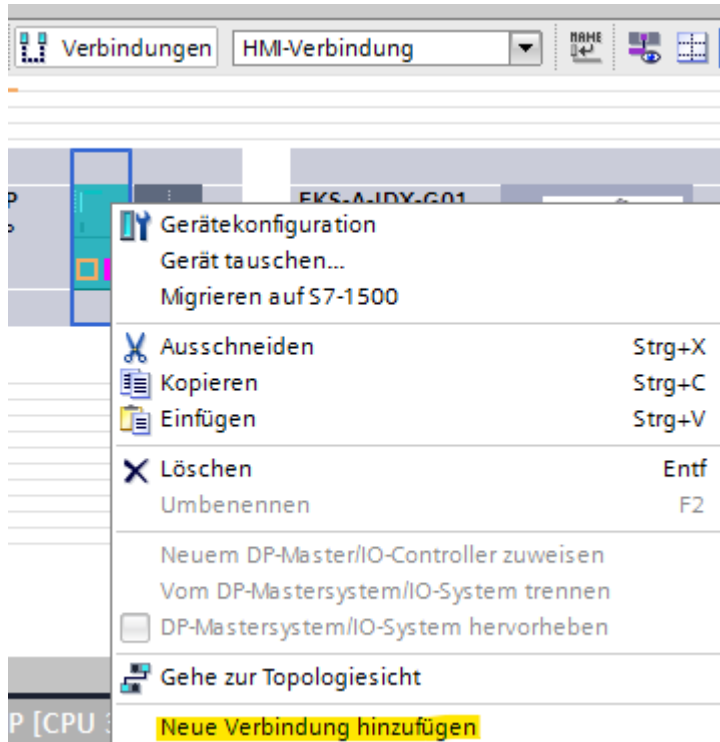
Für jede Instanz des Bausteins FB_EKSDataService muss eine Verbindung zum EKS Data Service PC in der CPU eingerichtet werden.



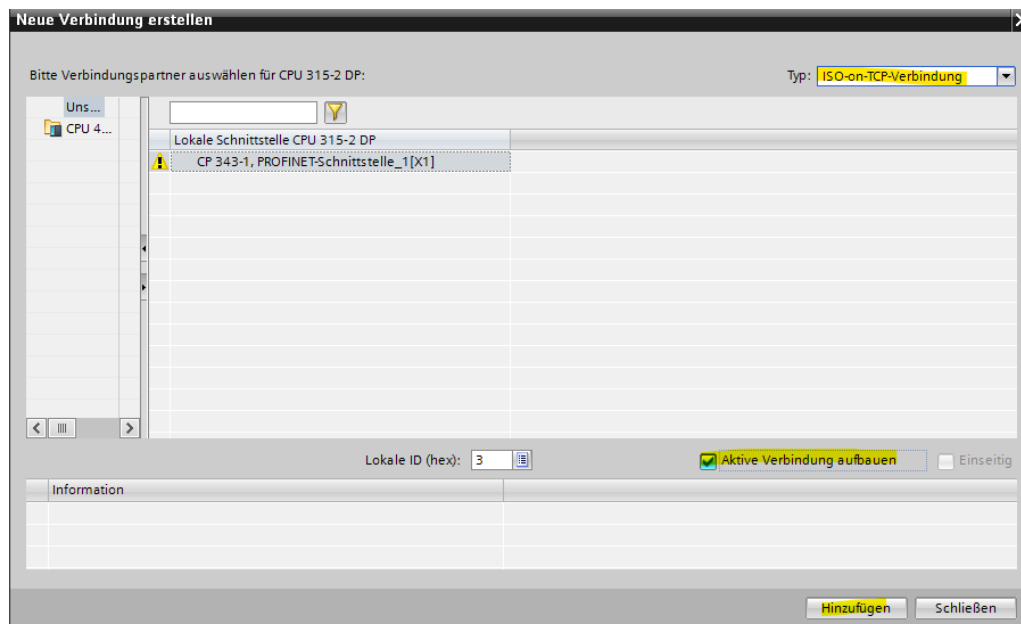
Topologiesicht Netzsicht Gerätesicht						
Netzübersicht		Verbindungen		E/A-Kommunikation		VPN TeleControl
	Lokaler Verbindungsna...	Lokaler Endpunkt	Lokale ID (...)	Partn..	Partner	Verbindungstyp
	ISOonTCP_Verbindung_1	CPU 315-2 DP [CPU 315-2 DP]	1		Unbekannt	ISO-on-TCP-Verbind...
	ISOonTCP_Verbindung_2	CPU 315-2 DP [CPU 315-2 DP]	2		Unbekannt	ISO-on-TCP-Verbind...
	ISO-on-TCP-Verb-5	CPU 414-3 PN/DP [CPU 414-3 PN/DP]	1		Unbekannt	ISO-on-TCP-Verbind...
	ISO-on-TCP-Verb-6	CPU 414-3 PN/DP [CPU 414-3 PN/DP]	2		Unbekannt	ISO-on-TCP-Verbind...

Neue Verbindung einfügen

Wählen Sie im Menü *Geräte & Netze* die Ansicht *Verbindungen* und machen Sie anschließend einen Rechtsklick auf die CPU und wählen Sie *Neue Verbindung hinzufügen*.



Typ *ISO-on-TCP-Verbindung* und *Aktive Verbindung aufbauen* auswählen und auf *Hinzufügen* drücken.



CPU 300 Verbindung und Bausteinparameter

- › Name der Verbindung kann geändert werden.
- › Aktiver Verbindungsaufbau muss markiert werden (besondere Verbindungseigenschaften).
- › Bausteinparameter sind für die Programmierung wichtig.

Adressen

Hier muss der TSAP eingestellt werden.

Der „Lokale TSAP“ wird für die Identifizierung der Kommunikation verwendet und wird im EKS Data Service PC angezeigt. Für mehrere Instanzen ist es notwendig unterschiedliche Beschreibungen zu verwenden, dies ist auch hilfreich für das Debugging im EKS Data Service PC, hier wird der Name für den Log-Eintrag verwendet.

Als Partner muss die IP Adresse des EKS Data Service PC eingetragen werden. Der TSAP des Partners muss *EKSSERV* sein.

	Lokal	Partner
IP (dez):	192.168.12.137	192.168.12.101
TSAP (ASCII):	CPU414_1	EKSSERV
TSAP (hex):	43.50.55.34.31.34.5F.31	45.4B.53.53.45.52.56
TSAP Länge:	8	7

5.4. EKSDatService Bausteinbeschreibung

Die Steuerung benötigt zwei Bausteine für das Arbeiten mit dem EKS Data Service PC.

Der erste Baustein ist ein FB (FB_EKSDatService) in welchem alle Funktionalitäten programmiert sind. Der zweite Baustein ist ein DB (DB_EKSDat1) in welchem die Daten für die Notlaufstrategie, die aktuellen Daten für den platzierten Schlüssel und die Deklaration der Daten liegen.

5.4.1. DB_EKSDat1

Alle Daten, die hier abgelegt sind, können auch auf andere Datenbausteine verteilt werden. In der Vorlage sind alle notwendigen Daten in einem DB vereint.

Daten platzierter Schlüssel (type_EKSDatabase)

Der Datentyp *type_EKSDatabase* beschreibt die Struktur der abgefragten Daten von der Datenbank.

Alle Daten die in der SPS von der Datenbank benötigt werden, können hier in einer Struktur festgelegt werden. Es ist notwendig, dass die Struktur mit der Beschreibung *type_EKSDescription* übereinstimmt (siehe hierzu auch das Software Handbuch EKS Data Service PC Kapitel 7.2.2).

Beispiel aus der Bibliothek:

type_EKSDatabase		
Name	Datentyp	Defaultwert
KeyID	String[16]	''
LOCKED	Bool	false
Key_Bit	Bool	false
Key_ShortInt	Byte	16#0
Key_Byte	Byte	16#0
Key_SmallInt	Int	0
Key_Word	Word	16#0
Key_Integer	Dint	0
Key_Float	Real	0.0
Key_String	String[12]	''
Key_StringBlankFilled	String[10]	''
Key_Time	Time	T#0ms
Key_TimeAscii	String[8]	''
Key_Date	Date	D#1990-01-01
Key_DateAscii	String[8]	''
Key_BitString	Word	16#0
KEYCRC	Word	16#0
DB_Bit	Bool	false
DB_ShortInt	Byte	16#0
DB_Byte	Byte	16#0
DB_SmallInt	Int	0
DB_Word	Word	16#0
DB_Integer	Dint	0
DB_Float	Real	0.0
DB_String	String[14]	''
DB_StringBlankFilled	String[12]	''
DB_Time	Time	T#0ms
DB_Date	Date	D#1990-01-01
DB_BitString	Byte	16#0

Im DB kann für jedes EKS ein eigenes Datenfeld erzeugt werden, in welchem die Daten des aktuell platzierten Schlüssels abgelegt werden.

DB_EKSDData1	
Name	Datentyp
Static	
Data_Key_Reader_Profibus	*type_EKSDatabase*

Beschreibung der angefragten Daten (type_EKSDescription)

Die Daten werden beim EKS Data Service PC angefragt. Diese Anfrage wird über eine Struktur definiert, die der Struktur vom Typ `type_EKSDatabase` entspricht (siehe hierzu auch das Software Handbuch EKS Data Service PC Kapitel 7.2.2).

Beispiel aus der Bibliothek:

type_EKSDescription		
Name	Datentyp	Defaultwert
KeyID	Int	1
LOCKED	Int	2
Key_Bit	Int	3
Key_ShortInt	Int	4
Key_Byte	Int	5
Key_SmallInt	Int	6
Key_Word	Int	7
Key_Integert	Int	8
Key_Float	Int	9
Key_String	Int	10
Key_StringBlankFilled	Int	11
Key_Time	Int	12
Key_TimeAscii	Int	13
Key_Date	Int	14
Key_DateAscii	Int	15
Key_BitString	Int	16
KEYCRC	Int	17
DB_Bit	Int	18
DB_ShortInt	Int	19
DB_Byte	Int	20
DB_SmallInt	Int	21
DB_Word	Int	22
DB_Integert	Int	23
DB_Float	Int	24
DB_String	Int	25
DB_StringBlankFilled	Int	26
DB_Time	Int	27
DB_Date	Int	28
DB_BitString	Int	29

Die Defaultwerte sind dabei wichtig und können im EKS Data Service PC eingestellt werden.

Im DB wird die Beschreibung des Typs *EKS Description* einmal einer Variablen zugeordnet:

DB_EKSDData1	
Name	Datentyp
Static	
Data_Key_Reader_Profibus	*type_EKSDatabase*
Description	*type_EKSDescription*

ID Array für die Verwaltung der Daten (type_EKSDataID)

Die ID wird benötigt, um zusätzliche Informationen, wie den Zeitstempel für das Platzieren des Schlüssels oder den Zeitstempel der letzten Aktualisierung, die KeyID und die Checksumme zu speichern. Diese zusätzlichen Informationen werden für jeden platzierten Schlüssel eingetragen und im DB abgelegt.

Dieser Datentyp darf nicht verändert werden und wird als Array in den DB abgelegt.

Für jede Datenspeicherung (Data Array) benötigt man ein ID Array mit der gleichen Größe.

Da die Daten auch nach Spannungsausfall/Neustart benötigt werden, ist es notwendig diese als remanent zu markieren.



Hinweis zur Checksumme

Der EKS Data Service berechnet über die angefragten Daten eine Checksumme (im weiteren Text auch kurz „CRC“ genannt) zur Gewährleistung der Integrität der angefragten Daten. Diese Daten werden vom Anwender nicht verarbeitet. Diese Checksumme ist nicht mit der Checksumme zu verwechseln, welche die Electronic-Key-Manager EKM Software berechnet. Die Checksumme des EKM (im weiteren Text auch kurz „Key CRC“ genannt) bezieht sich immer auf einen bestimmten Datenbereich, welcher im EKM konfiguriert werden kann. Die Key CRC des EKM wird auf dem Schlüssel mit abgespeichert und kann ausschließlich auf „On-Key“ Felder angewendet werden. Im Gegensatz dazu berechnet der EKS Data Service über alle angefragten Daten die CRC (sowohl „On-Key“ als auch EKM Datenbank-Werte).

type_EKSDataID

	Name	Datentyp	Defaultwert
	timestampPlaced	Date_And_Time	DT#1990-01-01-0
	timestampRefresh	Date_And_Time	DT#1990-01-01-0
	KeyID	"type_EKSKeyID"	
	CRC	Word	16#0

Beispiel aus der Bibliothek:







DB_EKSData1

	Name	Datentyp
	Static	
	▶ Data_Key_Reader_Profibus	"type_EKSDatabase"
	▶ Description	"type_EKSDescription"
	▶ ID	Array[0..20] of "type_EKSDataID"

Data Array für die Speicherung der Notlauf-Daten (type_EKSDatabase)

Der gleiche Datentyp für Daten der platzierten Schlüssel, wird auch für die Speicherung der Daten für die Notlaufstrategie verwendet. Alle Daten der platzierten Schlüssel werden in das Data Array eingetragen. Die Daten werden erst überschrieben, wenn die Anzahl der platzierten Schlüssel größer ist, als die Anzahl der Array Einträge. Sollten die Daten überschrieben werden, dann wird immer der älteste Schlüsseleintrag überschrieben. Wird ein Schlüssel bei der Anfrage beim EKS Data Service PC nicht gelistet, dann wird der Schlüssel auch im Data Array gelöscht. Die Instanzen des FB_EKSDataService können alle auf die gleichen Daten (Description, ID, Data) im DB_EKSData1 zugreifen. Sollten die Daten auch nach einem Spannungsausfall zu Verfügung stehen, ist es notwendig diese als remanent zu markieren.

Beispiel aus der Bibliothek:

DB_EKSData1		
	Name	Datentyp
	Static	
	▶ Data_Key_Reader_Profibus	"type_EKSDatabase"
	▶ Description	"type_EKSDescription" 
	▶ ID	Array[0..20] of "type_EKSDataID"
	▶ Data	Array[0..20] of "type_EKSDatabase"

5.4.2. FB_EKSDataService

Der FB_EKSDataService für die S7-300 / S7-400 besitzt die folgende Schnittstellenbeschreibung.

INPUT		
Name	Datentyp	Beschreibung
Key_Present	Bool	Schlüssel ist platziert
Key_Data_updated	Bool	Schlüsseldaten werden bei Platzierung erneuert
Interface_ID	Word	Profinet Schnittstelle Hardware Kennung
Con_ID	Int	Eindeutige Kennung der Verbindung
Timeout_Time	Time	Zeitüberschreitung Kommunikation
Refresh_Time	Time	Aktualisierungszeit für Schlüsseldaten

OUTPUT		
Name	Datentyp	Beschreibung
Con_Parametrization_Error	Word	Parametrierungsfehler der Kommunikation
Con_Communication_Error	Word	Kommunikationsfehler Server Dienst
DB_Error	Bool	Database Länge nicht richtig
Data_Error	Bool	Datenfehler vom Server Dienst
Emergency_Level_1	Bool	Notlaufstrategie 1 aktiv
Emergency_Level_2	Bool	Notlaufstrategie 2 aktiv
KEY_Not_Found	Bool	Schlüssel ID nicht gefunden
KEY_Checksum_Error	Bool	CRC Fehler (der angefragten Daten)
KEY_Valid	Bool	Schlüssel platziert und Daten verfügbar
Error	Bool	Bausteinfehler
Status	Word	Status des Bausteins
serviceStatus	Word	Status des Server Dienstes

INOUT		
Name	Datentyp	Beschreibung
Key_ID	type_EKSkeyID	Seriennummer des platzierten Schlüssels
EKS_Data_Key_Reader	Any	Datenablage für den platzierten Schlüssel
EKS_Data_Description	Any	Daten der Beschreibung
EKS_Data	Any	Daten Array der SPS Daten
EKS_KEY_ID	Any	ID Daten für die Datenstruktur



Hinweis

Der FB_EKSDataService ist durch ein Kennwort geschützt um ungewollte Manipulation zu verhindern. In manchen Fällen ist es notwendig, den FB_EKSDataService erneut zu übersetzen, wenn Sie ihn in Ihr Projekt mit eingebunden haben. Um den Baustein übersetzen zu können, benötigen Sie ein Kennwort. Bitte führen Sie keine Änderungen an der Implementierung des FB_EKSDataService durch.

Kennwort: EKS1234

Key_Present

Beim Lesen des platzierten Schlüssels vom EKS Lesegerät wird ein Signal erzeugt, dass die gelesene Schlüssel Seriennummer bestätigt. Dieses Signal ist für die Anfrage der Schlüsseldaten erforderlich. Wechselt Key_Present von FALSE auf TRUE, wird eine Datenanfrage an den EKS Data Service PC gesendet.

Key_Data_updated

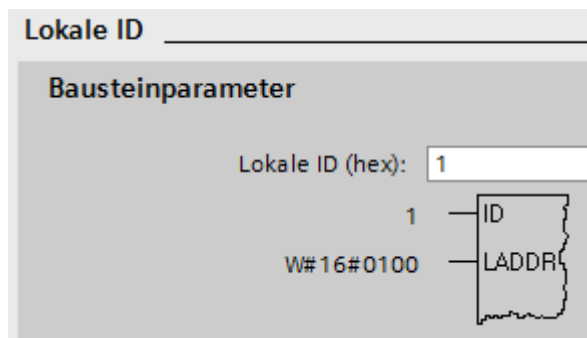
Wird der Parameter mit TRUE beschaltet, dann werden die Schlüsseldaten des platzierten Schlüssels immer mit der Datenbank aktualisiert. Dies bedeutet, dass sich die Daten während der Platzierung ändern können.

Beispiel:

Ein Schlüssel ist im EKS platziert, die angefragten Daten sind bereits in der SPS hinterlegt (DB_EKSData1.Data_Key_Reader_Profinet) und die Datenaktualisierung (Refresh_Time) ist aktiviert. D.h. die Daten der Notlaufstrategie werden im Hintergrund zyklisch aktualisiert. Sollten sich der aktualisierte Wert des platzierten Schlüssels und der bereits hinterlegte Wert unterscheiden, so wird bei Key_Data_updated = TRUE der Wert des aktuell platzierten Schlüssels (DB_EKSData1.Data_Key_Reader_Profinet) überschrieben. Andernfalls wird der aktualisierte Wert erst bei erneutem Schlüssel platzieren geladen.

Interface_ID

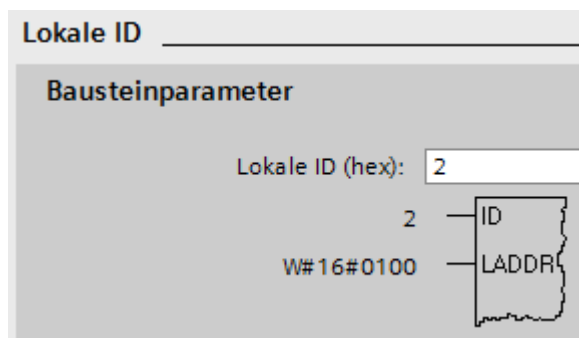
Die Hardware-Adresse wird aus dem Eigenschaftsfenster entnommen und muss am Baustein angegeben werden.



WORD#16#0100 — Interface_ID

Con_ID

Die Verbindungs_ID wird aus dem Eigenschaftsfenster entnommen und muss am Baustein angegeben werden.



WORD#16#0100 — Interface_ID
2 — Con_ID

Timeout_Time

Bei einer Zeitüberschreitung während der Kommunikation mit dem EKS Data Service PC, wird ein Fehler erzeugt, der die Kommunikation abbricht und einen Error mit Status ausgibt. Im Fehlerfall wird die Notlaufstrategie 2 ausgelöst und die angefragten Daten werden aus dem Speicher der Notlaufstrategie (DB_EKSData1.Data) geladen.

T#5S — Timeout_Time

Refresh_Time

In diesem Zeittakt werden die Daten der gelisteten Schlüssel im DB_EKSData1.Data vom EKS Data Service PC abgeholt und geprüft, ob diese aktuell sind. Andernfalls werden die Werte im DB_EKSData1.Data aktualisiert. D.h. bereits angefragte Daten werden im Hintergrund ohne Platzierung des Schlüssels aktualisiert. Der Baustein geht nach jedem Takt zum nächsten Eintrag und fragt die Daten beim EKS Data Service PC an. Bei einer Einstellung von 10 s benötigt der Dienst 100 s für eine Abfrage von 10 gespeicherten Schlüsseldaten.

Die Aktualisierung der Daten kann über den Wert 0 ms abgeschaltet werden. Werden mehrere Instanzen des FB_EKSDataService benötigt und greifen alle Instanzen auf die gleichen Daten zu, dann muss nur eine Instanz die Aktualisierung vornehmen.

Beispiel:

Instanz A: Refresh_Time = 10 s (sorgt für die Aktualisierung)

Instanz B: Refresh_Time = 0 ms (Aktualisierung ausgeschaltet)

Instanz C: Refresh_Time = 0 ms (Aktualisierung ausgeschaltet)

t#10s — Refresh_Time

Con_Parametrization_Errorcode

Wird beim Verbindungsaufbau ein Fehler erzeugt, wird der entsprechende Fehlercode hier eingetragen. Eine Liste der Fehlercodes finden Sie im Kapitel 7.1. *Verbindungsaufbau (Con_Parametrization_Errorcode)*.

Con_Communication_Errorcode

Wird beim Empfangen der Daten ein Fehler erzeugt, wird der entsprechende Fehlercode hier eingetragen. Eine Liste der Fehlercodes finden Sie im Kapitel 7.2. *Datenempfang (Con_Communication_Errorcode)*.

DB_Error

Das Fehler-Bit wird TRUE, wenn die Länge der Struktur *type_EKSDatabase* nicht mit der gesendeten Datengröße vom EKS Data Service PC übereinstimmt.

Data_Error

Der Fehler *Data_Error* wird durch den EKS Data Service PC ausgegeben, wenn die vorgegebene Datenstruktur nicht mit der Struktur vom EKS Data Service PC übereinstimmt. Meldet der EKS Data Service PC diesen Fehler, dann wird *Data_Error* auf TRUE gesetzt.

Emergency_Level_1

Notlaufstrategie 1, der EKS Data Service PC kann die original EKM CSV Exportdatei nicht erreichen und verwendet die lokale Backupdatei der EKM CSV Exportdatei.

Emergency_Level_2

Notlaufstrategie 2, der EKS Data Service PC kann keine Daten zur Verfügung stellen, weil die Kommunikation gestört ist und es werden die Daten des internen Datenbausteins verwendet (*DB_EKSData1.Daten*).

KEY_Not_Found

Der platzierte Schlüssel wurde nicht gefunden. Diese Information kann vom EKS Data Service PC kommen oder bei Notlaufstrategie 2 von der SPS, wenn der angefragte Schlüssel nicht im Speicher (*DB_EKSData1.Data*) hinterlegt ist.

KEY_Checksum_Error

Die CRC welche über die angefragten Daten berechnet wurde, stimmt nicht überein. Dieser Fehler wird ausgegeben wenn die CRC der gesendeten Daten nicht übereinstimmt oder im Falle der Notlaufstrategie 2, wenn die CRC der gespeicherten Daten nicht korrekt ist.

KEY_Valid

Die Daten des platzierten Schlüssels stehen zur Verfügung, dies gilt auch bei den Notlaufstrategien.

Error

Ein Fehler bei der Bausteinbearbeitung ist aufgetreten. Der Fehler wird bei der nächsten Bearbeitung gelöscht und ggf. wieder ausgegeben.

Status

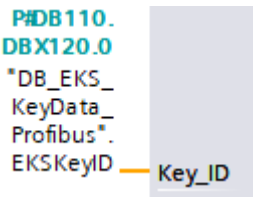
Statusmeldungen welche innerhalb der Verarbeitung im *FB_EKSDataService* generiert werden. Eine Liste der Statusmeldungen finden Sie im Kapitel 7.3. *Status des FB_EKSDataService (Status)*.

serviceStatus

Statuscodes welche vom EKS Data Service PC an den *FB_EKSDataService* übergeben werden. Eine Liste der Statusmeldungen finden Sie im Kapitel 7.4. *Status der Datenanfrage (serviceStatus)*.

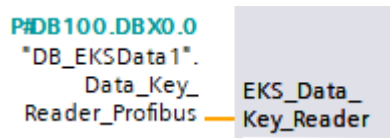
Key_ID

Die Seriennummer des platzierten Schlüssels muss hier angegeben werden, um die Daten vom EKS Data Service PC anzufragen. Der Datentyp `type_EKSKeyID` muss hierzu verwendet werden.



EKS_Data_Key_Reader

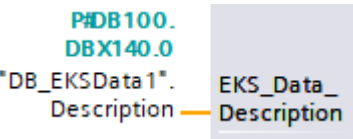
Die Daten des platzierten Schlüssels werden an diese Adresse kopiert. Es muss der Datentyp `type_EKSDatabase` verwendet werden.



DB_EKSData1	
Name	Datentyp
Static	
Data_Key_Reader_Profibus	"type_EKSDatabase"
Data	Array[0..20] of "type_EKSDatabase"

EKS_Data_Description

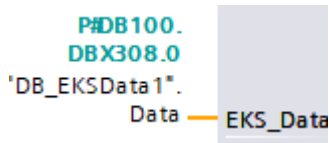
Die Beschreibung der Daten wird als Datentyp Variant übergeben. Diese Datenstruktur wird benutzt um die geforderten Daten vom EKS Data Service PC anzufragen. Es muss die Datenstruktur vom Datentyp `type_EKSDescription` verwendet werden.



DB_EKSData1	
Name	Datentyp
Static	
Data_Key_Reader_Profibus	"type_EKSDatabase"
Description	"type_EKSDescription"

EKS_Data

Das Datenarray für die Datensicherung auf SPS Seite wird als Variant angelegt (siehe Data DB_EKSDData1).

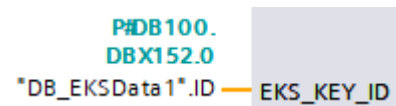


DB_EKSDData1	
Name	Datentyp
Static	
▶ Data_Key_Reader_Profibus	"type_EKSDatabase"
▶ Description	"type_EKSDescription"
▶ ID	Array[0..20] of "type_EKSDDataID"
▶ Data	Array[0..20] of "type_EKSDatabase"

Das Datenarray muss die gleiche Größe wie das EKS_KEY_ID Array haben.

EKS_KEY_ID

Für die Verwaltung der Datensicherung gibt es den ID Bereich (Siehe ID DB_EKSDData1).



DB_EKSDData1	
Name	Datentyp
Static	
▶ Data_Key_Reader_Profibus	"type_EKSDatabase"
▶ Description	"type_EKSDescription"
▶ ID	Array[0..20] of "type_EKSDDataID"

Das Datenarray muss die gleiche Größe wie das EKS_Data Array haben.

6. S7-300 / S7-400 SIMATIC STEP7

6.1. Bibliothek in SIMATIC STEP7 einbinden

Um die Bibliothek im SIMATIC STEP7 verwenden zu können, gehen Sie im SIMATIC Manager auf *Datei | Öffnen....* Es öffnet sich ein weiteres Fenster. Wechseln Sie zum Reiter *Bibliotheken* und klicken Sie anschließend auf *Durchsuchen....* Navigieren Sie zur Bibliothek und wählen Sie diese aus.

Die Bibliothek befindet sich in \PLC_Library\STEP_7_300_400\EKS_Data_Service_PLC_Library_STEP_7_190503.zip

Objektname	Symbolischer Name	Erstellsprache	Größe im Arbeitsspeicher	Typ	Version (Header)	Name (Header)
Systemdaten	---	---	---	---	---	---
OB1	Main	LAD	54	Organisationsbaustein	1.0	---
FB100	FB_EKSDatService	SCL	10638	Funktionsbaustein	1.3	EKSServ
FC1	FC_EKS_Euchner	LAD	702	Funktion	0.1	---
FC5	AG_SEND	AWL	1664	Funktion	4.2	AG_SEND
FC6	AG_RECV	AWL	1206	Funktion	4.7	AG_RECV
FC10	EQ_STRNG	AWL	152	Funktion	1.1	EQ_STRNG
FC11	AG_CNTRL	AWL	1152	Funktion	1.4	AG_CNTRL
FC14	GT_DT	AWL	338	Funktion	1.3	GT_DT
FC26	MID	AWL	302	Funktion	1.2	MID
FC29	NE_STRNG	AWL	188	Funktion	1.1	NE_STRNG
FC31	REPLACE	AWL	600	Funktion	1.2	REPLACE
FC38	STRNG_I	AWL	330	Funktion	1.1	STRNG_I
DB100	DB_EKSDat1	DB	3808	Datenbaustein	1.0	---
DB103	InstEKReadProfibus	DB	4750	Instanzdatenbaustei...	0.0	---
DB104	InstEKReadProfibus2	DB	4750	Instanzdatenbaustei...	0.0	---
DB110	DB_EKS_KeyData	DB	164	Datenbaustein abgel...	0.1	---
UDT100	type_EKSDatabase	AWL	---	Datentyp	1.0	---
UDT101	type_EKSDescription	AWL	---	Datentyp	1.0	---
UDT102	type_EKSKeyID	AWL	---	Datentyp	1.0	---
UDT103	type_EKSDatID	AWL	---	Datentyp	1.0	---
UDT110	type_EKSRead	AWL	---	Datentyp	1.0	---
SFB4	TON	AWL	---	Systemfunktionsbau...	1.0	TON
SFC1	READ_CLK	AWL	---	Systemfunktion	1.0	READ_CLK
SFC14	DPRD_DAT	AWL	---	Systemfunktion	1.0	DPRD_DAT
SFC20	BLKMOV	AWL	---	Systemfunktion	1.0	BLKMOV
SFC24	TEST_DB	AWL	---	Systemfunktion	1.0	TEST_DB
SFC58	WR_REC	AWL	---	Systemfunktion	1.0	WR_REC
SFC59	RD_REC	AWL	---	Systemfunktion	1.0	RD_REC

6.2. Bibliothek EKSDatService

In der Bibliothek (EKSDatService) befinden sich im Ordner S7-300 alle Bausteine für die CPU 300 und im Ordner S7-400 alle Bausteine für die CPU 400.

6.2.1. Verwendete Standard Siemens Bausteine

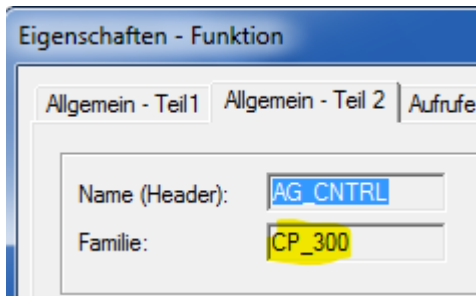
Für die Realisierung wurden aus der Siemens Bibliothek verschiedene Bausteine verwendet.

- AG_SEND (AG_LSEND)
- AG_RECV (AG_LRECV)
- EQ_STRNG
- AG_CNTRL
- GT_DT
- MID
- NE_STRNG
- REPLACE
- STRNG_I

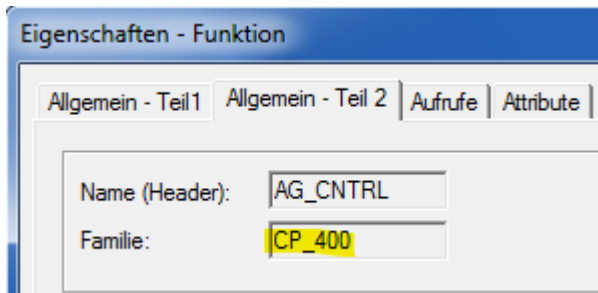
Diese Bausteine sind zwingend notwendig und sollten nicht in der Nummerierung geändert werden (Ausnahme bei symbolischem Vorrang). Die Bausteine der CPU 300 und CPU 400 unterscheiden sich von in ihrer Funktion. Es müssen die richtigen Bausteine der CPU genutzt werden.

Beispiel Bausteinfamilie:

CPU 300



CPU 400



6.2.2. Verwendete Systembausteine

Zu den Bibliotheksbausteinen werden auch Systembausteine verwendet.

- SFB4 TON
- SFC1 READ_CLK
- SFC14 DPRD_DAT
- SFC20 BLKMOV
- SFC24 TEST_DB
- SFC58 WR_REC
- SFC59 RD_REC

6.2.3. Baustein FB_EKSDataService

Der Baustein (FB_EKSDataService) ist für den Datenaustausch mit dem EKS Data Service PC und der Bereitstellung der Daten in der SPS zuständig.

6.2.4. Baustein DB_EKSData1

Der Datenbaustein (DB_EKSData1) ist ein Beispiel für den Aufbau der notwendigen Daten für den FB_EKSDataService.

6.2.5. Datentypen type_EKSxxxx

Die Datentypen sind notwendig für die Deklaration von Bausteinparametern und Datentypen im DB.

- type_EKSDatabase
- type_EKSDataID
- type_EKSDescription
- type_EKSKeyID

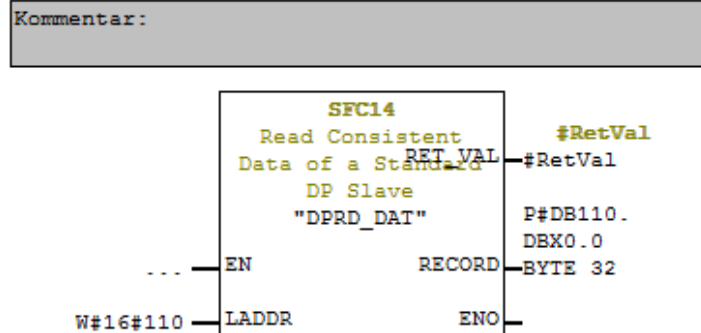
6.2.6. Baustein FC_EKS_Euchner

Der Baustein wird im Hauptprogramm Main (OB1) aufgerufen und verwaltet alle Euchner Bausteine im Zusammenhang mit dem EKS Data Service.

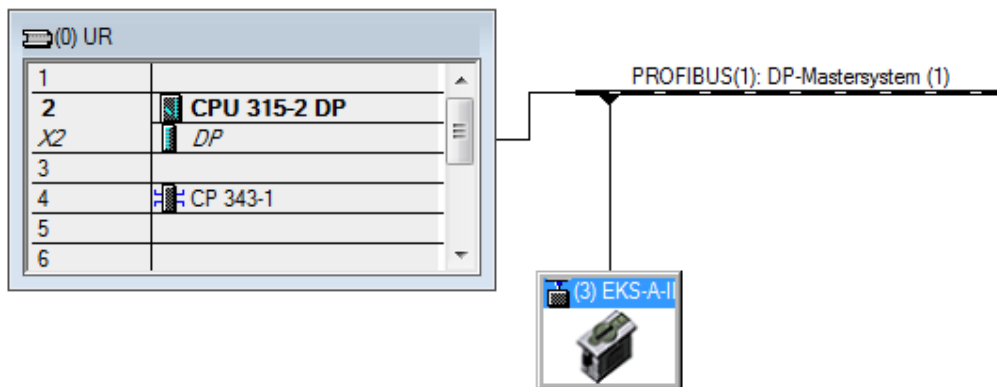
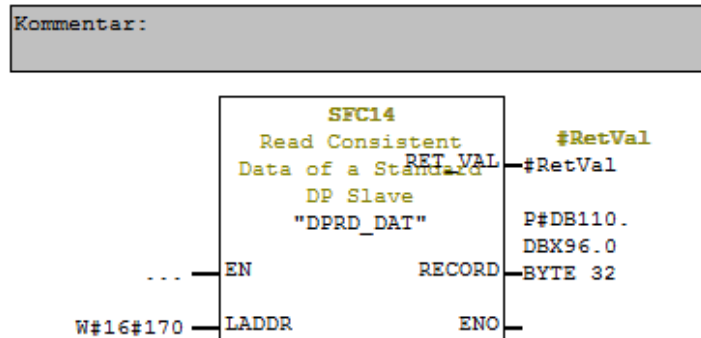
Im Beispiel werden Hardware-Adressen verwendet, die in der Gerätekonfiguration mit der Deklaration der Lesegeräte erzeugt werden. Die richtigen Adressen können nach der Deklaration der Hardware im Konfigurationsteil entnommen werden.

Adressverweis EKS_Euchner

Netzwerk 1 : Read EKS



Netzwerk 2 : Read EKS



[3] EKS-A-IDX-G01-ST09/03						
Steckplatz	DP-Kennung	...	Bestellnummer / Bezeichnung	E-Adresse	A-Adresse	Kommentar
1	192		Read/Write: 128/120 Byte I/O	272...303	272...303	
2	192		-> Read/Write: 128/120 Byte I/O	304...335	304...335	
3	192		-> Read/Write: 128/120 Byte I/O	336...367	336...367	
4	192		-> Read/Write: 128/120 Byte I/O	368...399	368...399	



Hinweis

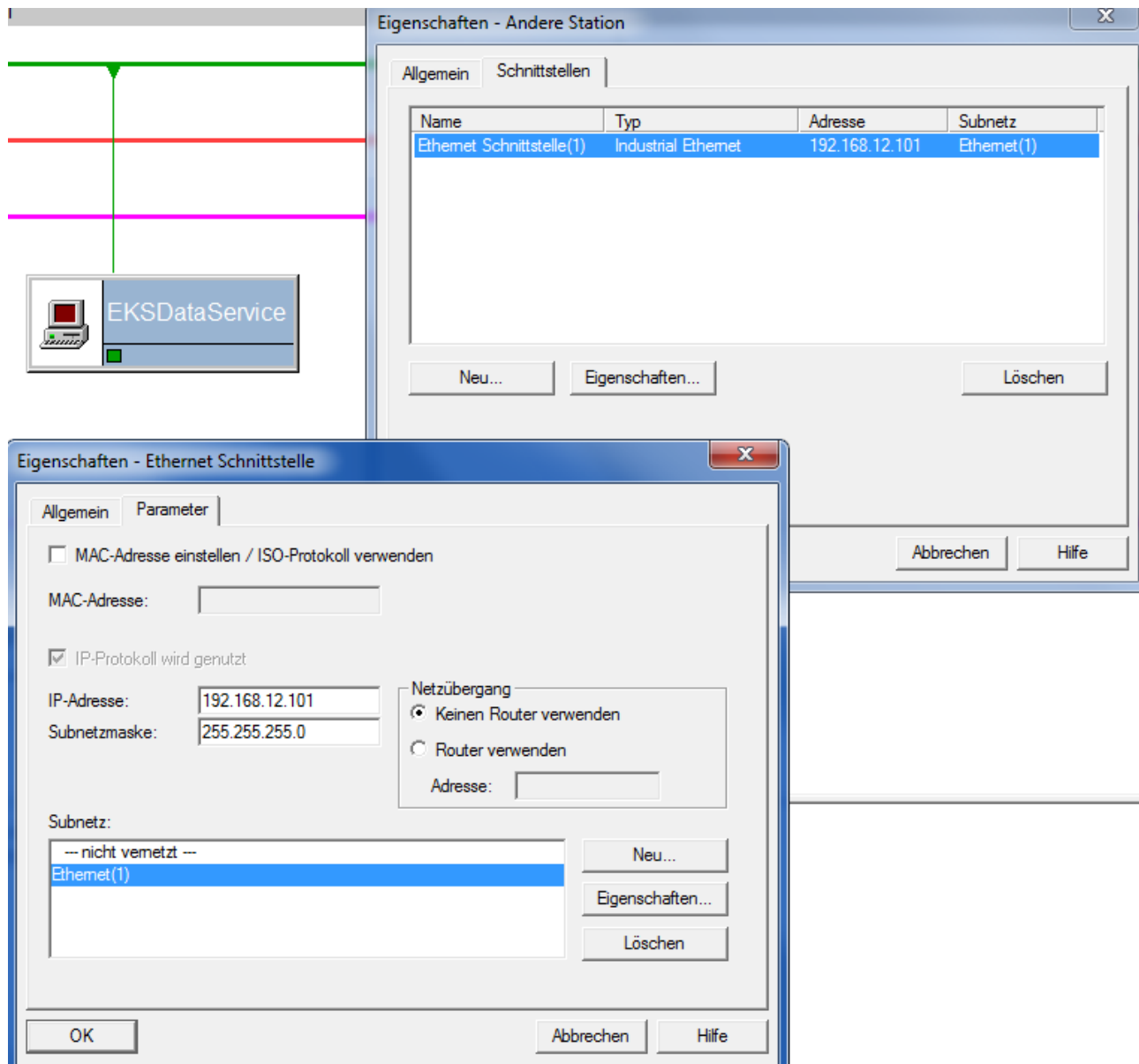
Die E-Adresse muss als HEX Wert an LADDR angegeben werden

6.3. Einstellungen NetPro (TCP/IP Konfiguration)

6.3.1. Einrichtung EKS Data Service PC

Der EKS Data Service PC kann in NetPro als *Andere Station* eingerichtet werden. Wird dies nicht gemacht, dann muss die Verbindung unspezifisch eingetragen werden.

Die *Andere Station* benötigt eine IP Adresse und muss einem Ethernet Subnetz zugewiesen werden.



Verbindung in der CPU einrichten

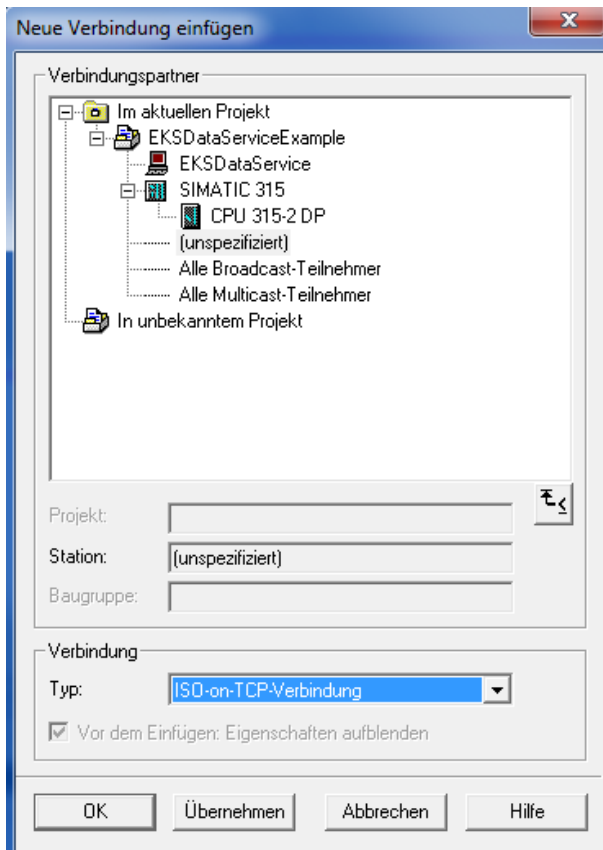
Für jede Instanz des Bausteins FB_EKSDataService muss eine Verbindung in der CPU eingerichtet werden. Machen Sie dafür in NetPro einen Rechtsklick auf die gewünschte CPU und fügen eine Verbindung ein.

Beispiel CPU 300:

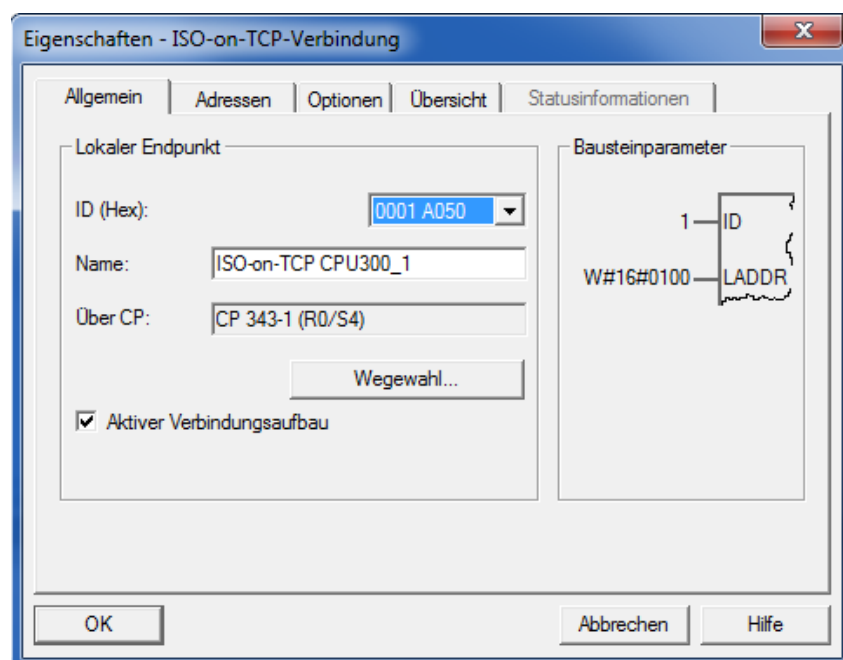
Lokale ID	Partner	Typ	Aktive	Betriebszust	Subnetz	Lokale Schnittstelle	Partner Schnittstelle	Lokale Adresse	Partner Adresse
0001 A050	EKSDataService	ISO-on-TCP-Verbindung	ja	-	Ethernet(1) [IE]	CP 343-1	Ethernet Schnittstelle(1)	192.168.12.145	192.168.12.101
0002 A050	EKSDataService	ISO-on-TCP-Verbindung	ja	-	Ethernet(1) [IE]	CP 343-1	Ethernet Schnittstelle(1)	192.168.12.145	192.168.12.101

EKS DataServicePC als unspezifizierte Verbindung einfügen

Unspezifizierte Verbindung auswählen und als Typ ISO-on-TCP-Verbindung verwenden.

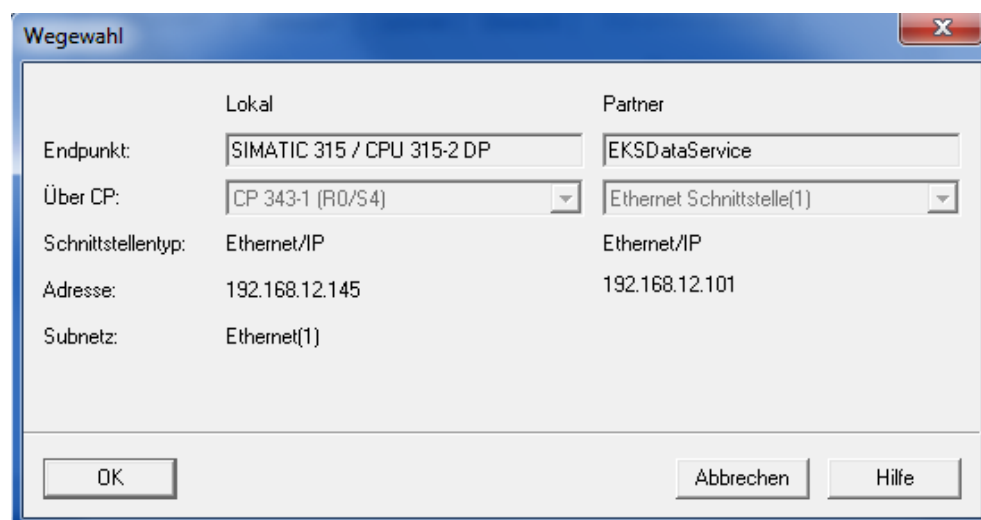


CPU 300 Verbindung und Bausteinparameter



- › Name der Verbindung kann geändert werden.
- › Aktiver Verbindungsaufbau muss markiert werden.
- › Bausteinparameter sind für die Programmierung wichtig.

CPU 300 Wegewahl...



Adressen

Hier muss der TSAP eingestellt werden. Für den Partner *EKSSERV*, Lokal kann eine beliebiger TSAP verwendet werden (wird beim Log Eintrag im EKS Data Service PC eingetragen).

Eigenschaften - ISO-on-TCP-Verbindung

Allgemein

Adressen

Optionen

Übersicht

Statusinformationen

	Lokal	Partner
IP (DEZ):	192.168.12.145	192.168.12.101
TSAP (ASC):	CPU300_1	EKSSERV
TSAP (HEX):	43.50.55.33.30.30.5F.31	45.4B.53.53.45.52.56
TSAP-Länge:	8	7

OK

Abbrechen

Hilfe

6.4. EKS Data Service Bausteinbeschreibung

Die Steuerung benötigt zwei Bausteine für das Arbeiten mit dem EKS Data Service PC.

Der erste Baustein ist ein FB (FB_EKSDataService) in welchem alle Funktionalitäten programmiert sind. Der zweite Baustein ist ein DB (DB_EKSData1) in welchem die Daten für die Notlaufstrategie, die aktuellen Daten für den platzierten Schlüssel und die Deklaration der Daten liegen.

6.4.1. DB_EKSData1

Alle Daten, die hier abgelegt sind, können auch auf andere Datenbausteine verteilt werden. In der Vorlage sind alle notwendigen Daten in einem DB vereint.

Daten platzierter Schlüssel *type_EKSDatabase*

Der Datentyp *type_EKSDatabase* beschreibt die Struktur der abgefragten Daten von der Datenbank.

Alle Daten die in der SPS von der Datenbank benötigt werden, können hier in einer Struktur festgelegt werden. Es ist notwendig, dass die Struktur mit der Beschreibung *type_EKSDescription* übereinstimmt (siehe hierzu auch das Software Handbuch EKS Data Service PC Kapitel 7.2.2).

Beispiel aus der Bibliothek:

UDT100 -- "type_EKSDatabase" -- EKSDataServiceExample\SIMATIC 315\CPU 315-2 DP\...\UDT100

Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+0.0	KeyID	STRING[16]	''	
+18.0	LOCKED	BOOL	FALSE	
+18.1	Key_Bit	BOOL	FALSE	
+19.0	Key_ShortInt	BYTE	B#16#0	
+20.0	Key_Byte	BYTE	B#16#0	
+22.0	Key_SmalInt	INT	0	
+24.0	Key_Word	WORD	W#16#0	
+26.0	Key_Integer	DINT	L#0	
+30.0	Key_Float	REAL	0.000000e+00	
+34.0	Key_String	STRING[12]	''	
+48.0	Key_StringBlankFilled	STRING[10]	''	
+60.0	Key_Time	TIME	T#0MS	
+64.0	Key_TimeAscii	STRING[8]	''	
+74.0	Key_Date	DATE	D#1990-1-1	
+76.0	Key_DateAscii	STRING[8]	''	
+86.0	Key_BitString	WORD	W#16#0	
+88.0	KEYCRC	WORD	W#16#0	
+90.0	DB_Bit	BOOL	FALSE	
+91.0	DB_ShortInt	BYTE	B#16#0	
+92.0	DB_Byte	BYTE	B#16#0	
+94.0	DB_SmalInt	INT	0	
+96.0	DB_Word	WORD	W#16#0	
+98.0	DB_Integer	DINT	L#0	
+102.0	DB_Float	REAL	0.000000e+00	
+106.0	DB_String	STRING[14]	''	
+122.0	DB_StringBlankFilled	STRING[12]	''	
+136.0	DB_Time	TIME	T#0MS	
+140.0	DB_Date	DATE	D#1990-1-1	
+142.0	DB_BitString	BYTE	B#16#0	
=144.0		END_STRUCT		

Im DB kann für jedes EKS ein eigenes Datenfeld erzeugt werden, in welchem die Daten des aktuell platzierten Schlüssels abgelegt werden.

DB100 -- "DB_EKSDat1" -- EKSDatServiceExample\SIMATIC 315\CPU 315-2 DP\...\DB1

Adresse	Name	Typ
0.0		STRUCT
+0.0	Data_Key_Reader_Profibus	"type_EKSDatabase"

Beschreibung der angefragten Daten (type_EKSDescription)

Die Daten werden beim EKS Data Service PC angefragt. Diese Anfrage wird über eine Struktur definiert, die der Struktur vom Typ type_EKSDatabase entspricht (siehe hierzu auch das Software Handbuch EKS Data Service PC Kapitel 7.2.2).

Beispiel aus der Bibliothek:

UDT101 -- "type_EKSDescription" -- EKSDatServiceExample\SIMATIC 315\CPU 315-2 DP\...\UDT101

Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+0.0	KeyID	INT	1	
+2.0	LOCKED	INT	2	
+4.0	Key_Bit	INT	3	
+6.0	Key_ShortInt	INT	4	
+8.0	Key_Byte	INT	5	
+10.0	Key_SmalInt	INT	6	
+12.0	Key_Word	INT	7	
+14.0	Key_Integer	INT	8	
+16.0	Key_Float	INT	9	
+18.0	Key_String	INT	10	
+20.0	Key_StringBlankFilled	INT	11	
+22.0	Key_Time	INT	12	
+24.0	Key_TimeAscii	INT	13	
+26.0	Key_Date	INT	14	
+28.0	Key_DateAscii	INT	15	
+30.0	Key_BitString	INT	16	
+32.0	KEYCRC	INT	17	
+34.0	DB_Bit	INT	18	
+36.0	DB_ShortInt	INT	19	
+38.0	DB_Byte	INT	20	
+40.0	DB_SmalInt	INT	21	
+42.0	DB_Word	INT	22	
+44.0	DB_Integer	INT	23	
+46.0	DB_Float	INT	24	
+48.0	DB_String	INT	25	
+50.0	DB_StringBlankFilled	INT	26	
+52.0	DB_Time	INT	27	
+54.0	DB_Date	INT	28	
+56.0	DB_BitString	INT	29	
+58.0		END_STRUCT		

Die Defaultwerte sind dabei wichtig und können im EKS Data Service PC eingestellt werden.

Im DB wird die *Description* einmal eingetragen:

DB100 -- "DB_EKSDat1" -- EKSDatServiceExample\SIMATIC 315\CPU 315-2 DP\...\DB100

Adresse	Name	Typ	Anfa
0.0		STRUCT	
+0.0	Data_Key_Reader_Profibus	"type_EKSDatabase"	
+144.0	Description	"type_EKSDescription"	
+202.0	ID	ARRAY[0..201]	

ID Array für die Verwaltung der Daten (type_EKSDataID)

Die ID wird benötigt, um zusätzliche Informationen wie den Zeitstempel für das Platzieren des Schlüssels oder den Zeitstempel der letzten Aktualisierung, die KeyID und die CRC zu speichern. Diese zusätzlichen Informationen werden für jeden platzierten Schlüssel eingetragen und im DB abgelegt.

Dieser Datentyp darf nicht verändert werden und wird als Array in den DB abgelegt.

Für jede Datenspeicherung (Data Array) benötigt man ein ID Array mit der gleichen Größe.

Da die Daten auch nach Spannungsausfall/Neustart benötigt werden, ist es notwendig diese als remanent zu markieren.



Hinweis zur Checksumme

Der EKS Data Service berechnet über die angefragten Daten eine Checksumme (im weiteren Text auch kurz „CRC“ genannt) zur Gewährleistung der Integrität der angefragten Daten. Diese Daten werden vom Anwender nicht verarbeitet. Diese Checksumme ist nicht mit der Checksumme zu verwechseln, welche die Electronic-Key-Manager EKM Software berechnet. Die Checksumme des EKM (im weiteren Text auch kurz „Key CRC“ genannt) bezieht sich immer auf einen bestimmten Datenbereich, welcher im EKM konfiguriert werden kann. Die Key CRC des EKM wird auf dem Schlüssel mit abgespeichert und kann ausschließlich auf „On-Key“ Felder angewendet werden. Im Gegensatz dazu berechnet der EKS Data Service über alle angefragten Daten die CRC (sowohl „On-Key“ als auch EKM Datenbank-Werte).

UDT103 -- "type_EKSDataID" -- EKSDatServiceExample\SIMATIC 315\CPU 315-2 DP\...\UDT103

Adresse	Name	Typ	Anfangswert
0.0		STRUCT	
+0.0	timestampPlaced	DATE_AND_TIME	DT#90-1-1-0:0:0.000
+8.0	timestampRefresh	DATE_AND_TIME	DT#90-1-1-0:0:0.000
+16.0	KeyID	"type_EKSKeyID"	
+24.0	CRC	WORD	W#16#0
=26.0		END_STRUCT	

Beispiel aus der Bibliothek:

DB100 -- "DB_EKSData1" -- EKSDatServiceExample\SIMATIC 315\CPU 315-2 DP\...\DB100

Adresse	Name	Typ	Anf
0.0		STRUCT	
+0.0	Data_Key_Reader_Profibus	"type_EKSDatabase"	
+144.0	Description	"type_EKSDescription"	
+202.0	ID	ARRAY[0..20]	
*26.0		"type_EKSDataID"	
+748.0	Data	ARRAY[0..20]	
*144.0		"type_EKSDatabase"	
=3772.0		END_STRUCT	

Data Array für die Speicherung der Notlauf-Daten (type_EKSDatabase)

Der gleiche Datentyp für Daten der platzierten Schlüssel, wird auch für die Speicherung der Daten für die Notlaufstrategie verwendet. Alle Daten der platzierten Schlüssel werden in das Data Array eingetragen. Die Daten werden erst überschrieben, wenn die Anzahl der platzierten Schlüssel größer ist, als die Anzahl der Array Einträge. Sollten die Daten überschrieben werden, dann wird immer der älteste Schlüsseleintrag überschrieben. Wird ein Schlüssel bei der Anfrage beim EKS Data Service PC nicht gelistet, dann wird der Schlüssel auch im Data Array gelöscht. Die Instanzen des FB_EKSDataService können alle auf die gleichen Daten (Description, ID, Data) im DB_EKSData1 zugreifen. Sollten die Daten auch nach einem Spannungsausfall zu Verfügung stehen, ist es notwendig diese als remanent zu markieren.

Beispiel aus der Bibliothek:

DB100 -- "DB_EKSData1" -- EKSDatabaseExample\SIMATIC 315\CPU 315-2 DP\...\DB100

Adresse	Name	Typ	Anfang
0.0		STRUCT	
+0.0	Data_Key_Reader_Profibus	"type_EKSDatabase"	
+144.0	Description	"type_EKSDescription"	
+202.0	ID	ARRAY[0..20]	
*26.0		"type_EKSDataID"	
+748.0	Data	ARRAY[0..20]	
*144.0		"type_EKSDatabase"	
=3772.0		END_STRUCT	

6.4.2. FB_EKSDataService

Der FB_EKSDataService für die S7-300 / S7-400 besitzt die folgende Schnittstellenbeschreibung.

INPUT		
Name	Datentyp	Beschreibung
Key_Present	Bool	Schlüssel ist platziert
Key_Data_updated	Bool	Schlüsseldaten werden bei Platzierung erneuert
Interface_ID	Word	Profinet Schnittstelle Hardware Kennung
Con_ID	Int	Eindeutige Kennung der Verbindung
Timeout_Time	Time	Zeitüberschreitung Kommunikation
Refresh_Time	Time	Aktualisierungszeit für Schlüsseldaten

OUTPUT		
Name	Datentyp	Beschreibung
Con_Parametrization_Err	Word	Parametrierungsfehler der Kommunikation
Con_Communication_Error	Word	Kommunikationsfehler Server Dienst
DB_Error	Bool	Database Länge nicht richtig
Data_Error	Bool	Datenfehler vom Server Dienst
Emergency_Level_1	Bool	Notlaufstrategie 1 aktiv
Emergency_Level_2	Bool	Notlaufstrategie 2 aktiv
KEY_Not_Found	Bool	Schlüssel ID nicht gefunden
KEY_Checksum_Error	Bool	CRC Fehler
KEY_Valid	Bool	Schlüssel platziert und Daten verfügbar
Error	Bool	Bausteinfehler
Status	Word	Status des Bausteins
serviceStatus	Word	Status des Server Dienstes

INOUT		
Name	Datentyp	Beschreibung
Key_ID	type_EKSKeyID	Seriennummer des platzierten Schlüssels
EKS_Data_Key_Reader	Any	Datenablage für den platzierten Schlüssel
EKS_Data_Description	Any	Daten der Beschreibung
EKS_Data	Any	Daten Array der SPS Daten
EKS_KEY_ID	Any	ID Daten für die Datenstruktur



Hinweis

Der FB_EKSDataService ist durch ein Kennwort geschützt um ungewollte Manipulation zu verhindern. In manchen Fällen ist es notwendig, den FB_EKSDataService erneut zu übersetzen, wenn Sie ihn in Ihr Projekt mit eingebunden haben. Um den Baustein übersetzen zu können, benötigen Sie ein Kennwort. Bitte führen Sie keine Änderungen an der Implementierung des FB_EKSDataService durch.

Kennwort: EKS1234

Key_Present

Beim Lesen des platzierten Schlüssels vom EKS Lesegerät, wird ein Signal erzeugt, dass die gelesene Schlüssel Seriennummer bestätigt. Dieses Signal ist für die Anfrage der Schlüsseldaten erforderlich. Wechselt Key_Present von FALSE auf TRUE, wird eine Datenanfrage an den EKS Data Service PC gesendet.

Key_Data_updated

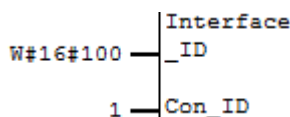
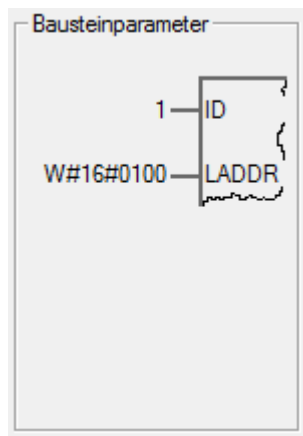
Wird der Parameter mit TRUE beschaltet, dann werden die Schlüsseldaten des platzierten Schlüssels immer mit der Datenbank aktualisiert. Dies bedeutet, dass sich die Daten während der Platzierung ändern können.

Beispiel:

Ein Schlüssel ist im EKS platziert und die angefragten Daten sind bereits in der SPS hinterlegt (DB_EKSData1.Data_Key_Reader_Profinet) und die Datenaktualisierung (Refresh_Time) ist aktiviert. D.h. die Daten der Notlaufstrategie werden im Hintergrund zyklisch aktualisiert. Sollten sich der aktualisierte Wert des platzierten Schlüssels und der bereits hinterlegte Wert unterscheiden, so wird bei Key_Data_updated = TRUE der Wert des aktuell platzierten Schlüssels (DB_EKSData1.Data_Key_Reader_Profinet) überschrieben. Andernfalls wird der aktualisierte Wert erst bei erneutem Schlüssel platzieren geladen.

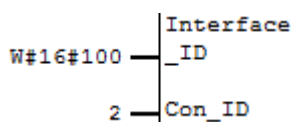
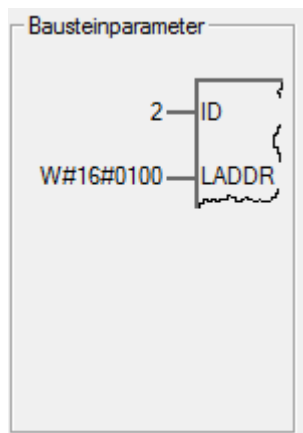
Interface_ID

Hardware Kennung der lokalen Schnittstelle über welche mit dem EKS Data Service PC kommuniziert werden soll.



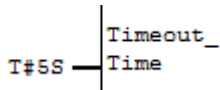
Con_ID

Die Verbindungs_ID wird aus dem Eigenschaftsfenster entnommen und muss am Baustein angegeben werden.



Timeout_Time

Bei einer Zeitüberschreitung während der Kommunikation mit dem EKS Data Service PC, wird ein Fehler erzeugt, der die Kommunikation abbricht und einen Error mit Status ausgibt. Im Fehlerfall wird die Notlaufstrategie 2 ausgelöst und die angefragten Daten werden aus dem Speicher der Notlaufstrategie (DB_EKSData1.Data) geladen.



Refresh_Time

In diesem Zeittakt werden die Daten der gelisteten Schlüssel im DB_EKSData1.Data vom EKS Data Service PC abgeholt und geprüft, ob diese aktuell sind. Andernfalls werden die Werte im DB_EKSData1.Data aktualisiert. D.h. bereits angefragte Daten werden im Hintergrund ohne Platzierung des Schlüssels aktualisiert. Der Baustein geht nach jedem Takt zum nächsten Eintrag und fragt die Daten beim EKS Data Service PC an. Bei einer Einstellung von 10 s benötigt der Dienst 100 s für eine Abfrage von 10 gespeicherten Schlüsseldaten.

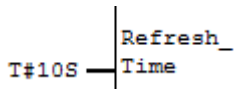
Die Aktualisierung der Daten kann über den Wert 0 ms abgeschaltet werden. Werden mehrere Instanzen des FB_EKSDataService benötigt und greifen alle Instanzen auf die gleichen Daten zu, dann muss nur eine Instanz die Aktualisierung vornehmen.

Beispiel:

Instanz A: Refresh_Time = 10 s (sorgt für die Aktualisierung)

Instanz B: Refresh_Time = 0 ms (Aktualisierung ausgeschaltet)

Instanz C: Refresh_Time = 0 ms (Aktualisierung ausgeschaltet)



Con_Parametrization_Errorcode

Wird beim Verbindungsaufbau ein Fehler erzeugt, wird der entsprechende Fehlercode hier eingetragen. Eine Liste der Fehlercodes finden Sie im Kapitel 7.1. *Verbindungsaufbau (Con_Parametrization_Errorcode)*.

Con_Communication_Errorcode

Wird beim Empfangen der Daten ein Fehler erzeugt, wird der entsprechende Fehlercode hier eingetragen. Eine Liste der Fehlercodes finden Sie im Kapitel 7.2. *Datenempfang (Con_Communication_Errorcode)*.

DB_Error

Das Fehler-Bit wird TRUE, wenn die Länge der Struktur *type_EKSDatabase* nicht mit der gesendeten Datengröße vom EKS Data Service PC übereinstimmt.

Data_Error

Der Fehler Data_Error wird durch den EKS Data Service PC ausgegeben, wenn die vorgegebene Datenstruktur nicht mit der Struktur vom EKS Data Service PC übereinstimmt. Meldet der EKS Data Service PC diesen Fehler, dann wird Data_Error auf TRUE gesetzt.

Emergency_Level_1

Notlaufstrategie 1, der EKS Data Service PC kann die original EKM CSV Exportdatei nicht erreichen und verwendet die lokale Backupdatei der EKM CSV Exportdatei.

Emergency_Level_2

Notlaufstrategie 2, der EKS Data Service PC kann keine Daten zur Verfügung stellen, weil die Kommunikation gestört ist und es werden die Daten des internen Datenbausteins verwendet (DB_EKSData1.Daten).

KEY_Not_Found

Der platzierte Schlüssel wurde nicht gefunden. Diese Information kann vom EKS Data Service PC kommen oder bei Notlaufstrategie 2 von der SPS, wenn der angefragte Schlüssel nicht im Speicher (DB_EKSData1.Data) hinterlegt ist.

KEY_Checksum_Error

Die CRC welche über die angefragten Daten berechnet wurde, stimmt nicht überein. Dieser Fehler wird ausgegeben wenn die CRC der gesendeten Daten nicht übereinstimmt oder im Falle der Notlaufstrategie 2, wenn die CRC der gespeicherten Daten nicht korrekt ist.

KEY_Valid

Die Daten des platzierten Schlüssels stehen zur Verfügung, dies geht auch bei den Notlaufstrategien.

Error

Ein Fehler bei der Bausteinbearbeitung ist aufgetreten. Der Fehler wird bei der nächsten Bearbeitung gelöscht und ggf. wieder ausgegeben.

Status

Statusmeldungen welche innerhalb der Verarbeitung im FB_EKSDataService generiert werden. Eine Liste der Statusmeldungen finden Sie im Kapitel 7.3. *Status des FB_EKSDataService (Status)*.

serviceStatus

Statuscodes welche vom EKS Data Service PC an den FB_EKSDataService übergeben werden. Eine Liste der Statusmeldungen finden Sie im Kapitel 7.4. *Status der Datenanfrage (serviceStatus)*.

Key_ID

Die Seriennummer des platzierten Schlüssels muss hier angegeben werden, um die Daten vom EKS Data Service PC anzufragen. Der Datentyp `type_EKSKeyID` muss hierzu benutzt werden.

```
P#DB110.DB
X120.0
"DB_EKS_
KeyData".
EKSKeyID — Key_ID
```

EKS_Data_Key_Reader

Die Daten des platzierten Schlüssels werden an diese Adresse kopiert. Es muss der Datentyp `type_EKSDatabase` verwendet werden.

```
P#DB100.DB
X0.0
"DB_
EKSData1".
Data_Key_
Reader_
Profibus — EKS_Data_
Key_
Reader
```

DB100 -- "DB_EKSData1" -- EKSDataServiceExample\SIMATIC 315\CPU 315-2 DP\...\DB100

Adresse	Name	Typ	Anfa
0.0		STRUCT	
+0.0	Data_Key_Reader_Profibus	"type_EKSDatabase"	
-			
+748.0	Data	ARRAY[0..20]	
*144.0		"type_EKSDatabase"	

EKS_Data_Description

Die Beschreibung der Daten wird als Datentyp Variant übergeben. Diese Datenstruktur wird benutzt um die geforderten Daten vom EKS Data Service PC anzufragen. Es muss die Datenstruktur vom Datentyp `type_EKSDescription` verwendet werden.

```
P#DB100.DB
X70.0
"DB_
EKSDat1".
Description
n      on
```

DB100 -- "DB_EKSDat1" -- EKSDataServiceExample\SIMATIC 315\CPU 315-2 DP\...\DB100

Adresse	Name	Typ	Anf
0.0		STRUCT	
+0.0	Data_Key_Reader_Profibus	"type_EKSDatabase"	
+144.0	Description	"type_EKSDescription"	

EKS_Data

Das Datenarray für die Datensicherung auf SPS Seite wird als Variant angelegt (siehe DB_EKSDat1).

```
P#DB100.DB
X238.0
"DB_
EKSDat1".
Data      EKS_Data
```

DB100 -- "DB_EKSDat1" -- EKSDataServiceExample\SIMATIC 315\CPU 315-2 DP\...\DB100

Adresse	Name	Typ	Anf
0.0		STRUCT	
+0.0	Data_Key_Reader_Profibus	"type_EKSDatabase"	
+144.0	Description	"type_EKSDescription"	
+202.0	ID	ARRAY[0..20]	
*26.0		"type_EKSDataID"	
+748.0	Data	ARRAY[0..20]	
*144.0		"type_EKSDatabase"	

Das Datenarray muss die gleiche Größe wie EKS_KEY_ID Array haben.

EKS_KEY_ID

Für die Verwaltung der Datensicherung gibt es den ID Bereich (siehe ID DB_EKSDat1).

```
P#DB100.DB
X82.0
"DB_
EKSDat1".
ID      ID
```

DB100 -- "DB_EKSDat1" -- EKSDataServiceExample\SIMATIC 315\CPU 315-2 DP\...\DB100

Adresse	Name	Typ	Anf
0.0		STRUCT	
+0.0	Data_Key_Reader_Profibus	"type_EKSDatabase"	
+144.0	Description	"type_EKSDescription"	
+202.0	ID	ARRAY[0..20]	
*26.0		"type_EKSDataID"	

Das Datenarray muss die gleiche Größe wie das EKS_Data Array haben.

7. Statusmeldungen

7.1. Verbindungsaufbau (Con_Parametrization_Errorcode)

Auszug aus der Siemens Beschreibung:

STATUS* (W#16#...)	Erläuterung
0	Verbindung wurde erfolgreich aufgebaut.
7000	Keine Auftragsbearbeitung aktiv
7001	Auftragsbearbeitung starten, Verbindung aufbauen.
7002	Verbindung wird aufgebaut (REQ irrelevant).
8085	Die Verbindungs-ID (Parameter ID) wird bereits von einer konfigurierten Verbindung verwendet.
8086	Der Parameter ID liegt außerhalb des zulässigen Bereichs.
8087	Maximale Anzahl von Verbindungen erreicht, keine weitere Verbindung möglich
8089	Der Parameter CONNECT zeigt nicht auf eine Verbindungsbeschreibung oder die Verbindungsbeschreibung wurde manuell erstellt.
809A	Die Struktur am Parameter CONNECT wird an einer integrierten Schnittstelle nicht unterstützt oder die Länge ist ungültig.
809B	Das Element Interfaceld innerhalb der TCON_xxx Struktur referenziert keine Hardware-Kennung einer CPU- oder CM/CP Schnittstelle oder hat den Wert „0“.
80A1	Die angegebene Verbindung oder der Port sind bereits belegt.
80A2	Lokaler oder remoter Port wird vom System verwendet. Lokal sind folgende Ports reserviert: 20, 21, 80, 102, 135, 161, 162, 443, 34962, 34963, 34964 sowie der Bereich 49152 bis 65535.
80A3	ID wird von einer durch das Anwenderprogramm erstellten Verbindung genutzt, die auch die gleiche Verbindungsbeschreibung am Parameter CONNECT verwendet.
80A4	IP-Adresse des remoten Endpunkts der Verbindung ist ungültig oder sie entspricht der IP-Adresse des lokalen Partners.
80A7	Kommunikationsfehler: Sie haben „TDISCON“ ausgeführt, bevor „TCON“ beendet war.
80B4	Nur bei TCON_IP_RFC: Der lokale T-Selektor wurde nicht angegeben oder das erste Byte enthält nicht den Wert 0x0E (nur bei einer Länge von T-Selector = 2) oder der lokale T-Selektor beginnt mit „SIMATIC“.
80B5	Bei Verbindungstyp 13 = UDP ist nur ein passiver Verbindungsaufbau zulässig (Parameter ActiveEstablished der Struktur TCON_IP_v4 / TCON_PARAM hat den Wert TRUE).
80B6	Parametrierungsfehler im Parameter ConnectionType des Datenbausteins für Verbindungsbeschreibung.
	Nur gültig bei TCON_IP_v4: 0x11, 0x0B und 0x13.
	Nur gültig bei TCON_IP_RFC: 0x0C und 0x12
80B7	Bei TCON_IP_v4:
	TCP (aktiver Verbindungsaufbau): Entfernter Port ist „0“.
	TCP (passiver Verbindungsaufbau): Lokaler Port ist „0“.
	UDP: Lokaler Port ist „0“.
	IP-Adresse des Partner-Endpunkts wurde auf 0.0.0.0 gesetzt.
	Bei TCON_IP_RFC:
	Lokaler (LocalTSelector) oder entfernter (RemoteTSelector) T-Selektor wurde mit einer Länge von mehr als 32 Byte angegeben.
	Bei TSELLength des T-Selektors (lokal oder entfernt) wurde eine Länge größer als 32 eingegeben.

	Fehler bei der Länge der IP-Adresse des jeweiligen Verbindungspartners.
	IP-Adresse des Partner-Endpunkts wurde auf 0.0.0.0 gesetzt.
80B8	Parameter ID in der lokalen Verbindungsbeschreibung (Struktur am Parameter CONNECT) und Parameter ID der Anweisung sind unterschiedlich.
80C3	Alle Verbindungsressourcen sind belegt, oder Ports werden möglicherweise von anderen Anwendungen oder Verbindungen dynamisch verwendet.
80C4	Temporärer Kommunikationsfehler:
	Die Verbindung kann derzeit nicht aufgebaut werden.
	Die Verbindung kann nicht aufgebaut werden, weil auf dem Verbindungsweg liegende Firewalls für die benötigten Ports nicht freigeschaltet sind.
	Die Schnittstelle empfängt gerade neue Parameter.
	Die projektierte Verbindung wird gerade von einer Anweisung „TDISCON“ entfernt.
80C5	Der Verbindungspartner verweigert den Verbindungsaufbau, hat die Verbindung abgebaut oder aktiv beendet.
80C6	Der Verbindungspartner kann nicht erreicht werden (Netzwerkfehler).
80C7	Zeitüberschreitung der Ausführung.
80C8	Wert am Parameter ID wird bereits von einer Verbindung genutzt, die über das Anwenderprogramm erstellt wurde. Die Verbindung verwendet die gleiche ID, aber unterschiedliche Verbindungseinstellungen am Parameter CONNECT.
80C9	Validierung des Verbindungspartners fehlgeschlagen. Der Verbindungspartner, der eine Verbindung aufbauen will, entspricht nicht dem definierten Partner der Struktur am Parameter CONNECT.
80CE	Die IP-Adresse des lokalen Interfaces ist 0.0.0.0.
80D0	Im Zusammenhang mit TCP und dem aktiven Verbindungs-Endpunkt: Der Parameter remote_qdn ist eine leere Zeichenkette. In diesem Fall kann keine Verbindung aufgebaut werden.

7.2. Datenempfang (Con_Communication_Errorcode)

Auszug aus der Siemens Beschreibung:

STATUS* (W#16#...)	Erläuterung
0	Sendeauftrag fehlerfrei beendet.
7000	Keine Auftragsbearbeitung aktiv.
7001	Beginn der Auftragsbearbeitung, Daten werden gesendet.
	Während dieser Bearbeitung greift das Betriebssystem auf die Daten im Sendebereich DATA zu.
7002	Auftrag wird bearbeitet (REQ irrelevant).
	Während dieser Bearbeitung greift das Betriebssystem auf die Daten im Sendebereich DATA zu.
8085	Parameter LEN ist größer als der größte zulässige Wert (65536).
	Parameter DATA und LEN haben beide den Wert „0“.
8086	Der Parameter ID liegt außerhalb des zulässigen Adressbereichs (1..0xFFFF).
8088	Parameter LEN ist größer als der in DATA angegebene Bereich.
80A1	Kommunikationsfehler:
	Die angegebene Verbindung wurde noch nicht aufgebaut.
	Die angegebene Verbindung wird gerade beendet. Übertragung über diese Verbindung ist nicht möglich.
	Die Schnittstelle wird neu initialisiert.
80B3	Die parametrisierte Protokollvariante (Parameter ConnectionType in der Verbindungsbeschreibung) ist UDP. Bitte verwenden Sie bei einer UDP-Verbindung die Anweisung „TUSEND“.
80C3	Ein Baustein mit dieser ID wird bereits in einer anderen Prioritätsgruppe bearbeitet.
	Interner Mangel an Ressourcen.
80C4	Temporärer Kommunikationsfehler:
	Die Verbindung zum Partner kann derzeit nicht aufgebaut werden.
	Die Schnittstelle empfängt gerade neue Parametereinstellungen oder die Verbindung wird gerade aufgebaut.
80C5	Verbindung durch den Kommunikationspartner abgebaut.
80C6	Netzwerkfehler. Kommunikationspartner kann nicht erreicht werden.
80C7	Zeitüberschreitung bei Ausführung.

7.3. Status des FB_EKSDataService (Status)

STATUS* (W#16#...)	Erläuterung
5001	Sendefehler
5002	Empfangsfehler
5003	TimeOut Empfang
5004	Längenfehler Empfangsdaten
5005	Telegramm DATA_ERROR
5006	CRC Fehler bei Datenempfang
5007	TimeOut Senden
5008	KeyID der Empfangsdaten unterscheiden sich mit KeyID der Anfrage
6001	Array Länge für ID und Daten sind nicht gleich
6002	Emergency Level 2 ist aktiv und es gibt keine Key Daten
6003	CRC Fehler bei Datenherstellung über DB Daten

7.4. Status der Datenanfrage (serviceStatus)

STATUS* (W#16#...)	Erläuterung
E001	Schlüssel unbekannt
E002	Feld bei der Beschreibung unbekannt
E003	Telegrammgröße überschritten
E004	Format Fehler
E005	EKM CSV Datei Fehler
F001	Notlaufstrategie 1 EKM CSV Datei nicht verwendbar

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

Ausgabe:
2528410-02-08/23
Titel:
Software Handbuch
EKS Data Service PLC
(Originalbetriebsanleitung)
Copyright:
© EUCHNER GmbH + Co. KG, 08/2023

Technische Änderungen vorbehalten,
alle Angaben ohne Gewähr.