



Inhalt

1.	Zu diesem Dokument	3
1.1.	Version	3
1.2.	Gültigkeit	3
1.3.	Zielgruppe	3
1.4.	Ergänzende Dokumente	3
1.5.	Hinweis	3
2.	Verwendete Bauteile / Module	4
2.1.	EUCHNER	4
2.2.	Andere	4
2.3.	Software	4
3.	Funktionsbeschreibung	5
3.1.	Allgemein	5
4.	PROFINET-Konfiguration	5
4.1.	GSDML-Datei einbinden	5
4.2.	Parametrierung der Steuerung	5
4.3.	Einfügen des PROFINET-Controllers	6
4.4.	Einfügen des EKS2 mit PROFINET-Schnittstelle	7
4.5.	Parametrierung des EKS2	8
4.5.1.	Gerätename und IP-Adresse	8
4.6.	Module und Submodule	9
4.7.	Datenstruktur EU001	11
4.7.1.	Konfiguration eines Moduls	11
4.7.2.	Parametrierung der Endianness	13
4.8.	Datenstruktur EU000	14
4.8.1.	Konfiguration der Module und Submodule	14
5.	Verwenden der BECKHOFF Bibliothek	16
5.1.	Installation der Bibliothek	16
5.2.	Datenstruktur EU001	18
5.2.1.	Beschreibung der Bausteinschnittstelle	18
5.2.2.	Aufruf des Bausteins FB_EKS2_Eu001Base	19
5.2.3.	Verknüpfung des Ein- und Ausgangsbereichs	19
5.2.4.	EKS2 Online Daten	22
5.3.	Datenstruktur EU000	23
5.3.1.	Beschreibung der Bausteinschnittstelle	23
5.3.2.	Aufruf des Bausteins FB_EKS2_Eu000Base	24
5.3.3.	Verknüpfung des Ein- und Ausgangsbereichs	24
5.3.4.	EKS2 Online Daten	27
6.	Wichtiger Hinweis – Bitte unbedingt sorgfältig beachten!	28

1. Zu diesem Dokument

1.1. Version

Version	Datum	Änderung/Erweiterung	Kapitel
01-05/26	05/2026	Erstellung	Alle

1.2. Gültigkeit

Diese Applikation beschreibt die Einbindung und Projektierung des Electronic-Key-Systems EKS2 (im nachfolgenden nur noch kurz *EKS2* genannt) mit PROFINET-IO-Schnittstelle in Beckhoff TwinCAT 3.

1.3. Zielgruppe

Konstrukteure und Anlagenplaner für Sicherheitseinrichtungen an Maschinen sowie Inbetriebnahme- und Servicefachkräfte, die über folgende Kenntnisse verfügen:

- spezielle Kenntnisse im Umgang mit Sicherheitsbauteilen
- Kenntnisse in Installation, Inbetriebnahme, Programmierung und Diagnose von speicherprogrammierbaren Steuerungen (SPS) und Bussystemen
- Kenntnis der geltenden EMV-Vorschriften
- Kenntnis der geltenden Vorschriften zur Arbeitssicherheit und Unfallverhütung

1.4. Ergänzende Dokumente

Die Gesamtdokumentation für diese Applikation besteht aus folgenden Dokumenten:

Dokumenttitel (Dokumentnummer)	Inhalt
Betriebsanleitung (MAN20001715)	Transpondercodiertes Schlüsselsystem Electronic-Key-System EKS2
Ggf. beiliegende Daten- blätter	Artikelspezifische Information zu Abweichungen oder Ergänzungen

1.5. Hinweis

Dieses Dokument basiert auf der Betriebsanleitung des EKS2. Die technischen Details sowie weitere Informationen können der Betriebsanleitung entnommen werden.

2. Verwendete Bauteile / Module

2.1. EUCHNER

Beschreibung	Bestellnummer / Artikel
Auswertegerät/Leseinheit EKS2	170904 / EKS2-E-PN-MH1-170904
	170915 / EKS2-R1A1-B1-170915
Schlüssel EKS2	168433 / EKS2-K-K-B-D2-BK-168433
	168434 / EKS2-K-K-B-D2-BU-168434
	168435 / EKS2-K-K-B-D2-GN-168435
	168438 / EKS2-K-K-B-D2-OG-168438
	168432 / EKS2-K-K-B-D2-RD-168432
	168437 / EKS2-K-K-B-D2-WH-168437
	168436 / EKS2-K-K-B-D2-YE-168436



TIPP!

Weitere Informationen und Downloads zu den o.g. EUCHNER-Produkten finden Sie unter www.euchner.com. Geben Sie einfach die Bestellnummer in die Suche ein.

2.2. Andere

Beschreibung	Bestellnummer / Artikel
CX2030-M930	CX2030-M930

2.3. Software

Beschreibung	Version
TwinCAT	v3.1.4024.62

3. Funktionsbeschreibung

3.1. Allgemein

Diese Applikation unterstützt die Integration des EKS2. Die Beschreibung umfasst die Hardware-Einbindung sowie die Integration und Nutzung der optionalen Softwarebibliothek zur Datenaufbereitung.

Das EKS2 arbeitet als IO-Device im PROFINET-Netzwerk.

Die Betriebsanleitung des EKS2 enthält die Kommunikationsdaten.

4. PROFINET-Konfiguration

Das EKS2 tauscht die Daten in der PROFINET-Umgebung über Datenblöcke mit den zugehörigen Kommunikationsdaten aus. Die Projektierungssoftware nutzt die GSDML-Datei zur Konfiguration und Parametrierung der benötigten Module.

4.1. GSDML-Datei einbinden

Für die Einbindung des EKS2 in TwinCAT 3 wird eine GSD-Datei im GSDML-Format benötigt. Die GSD-Dateien stehen im Downloadbereich unter www.euchner.com zur Verfügung. Für den Einsatz immer die aktuelle GSD-Datei verwenden. Die GSDML-Datei und die zugehörige Bitmap-Datei in das entsprechende TwinCAT-Verzeichnis kopieren.

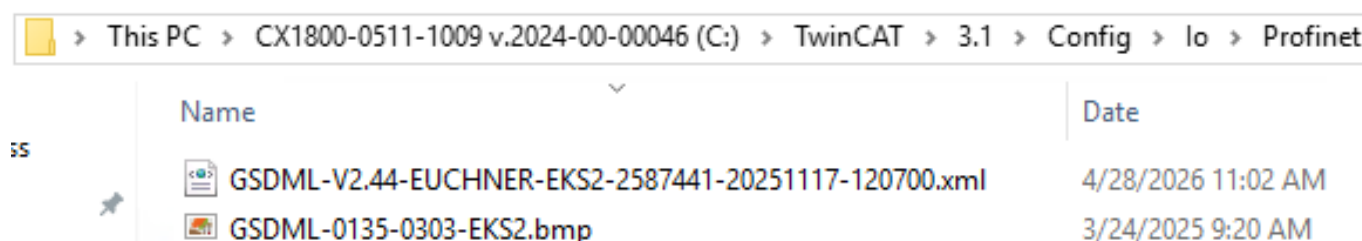


Bild 1: Einfügen GSD-Datei

4.2. Parametrierung der Steuerung

Die PROFINET-Zykluszeit des PLC-Task richtet sich nach den Anforderungen der Anwendung. Weitere Informationen enthält das [BECKHOFF Information System](http://www.beckhoff.com).

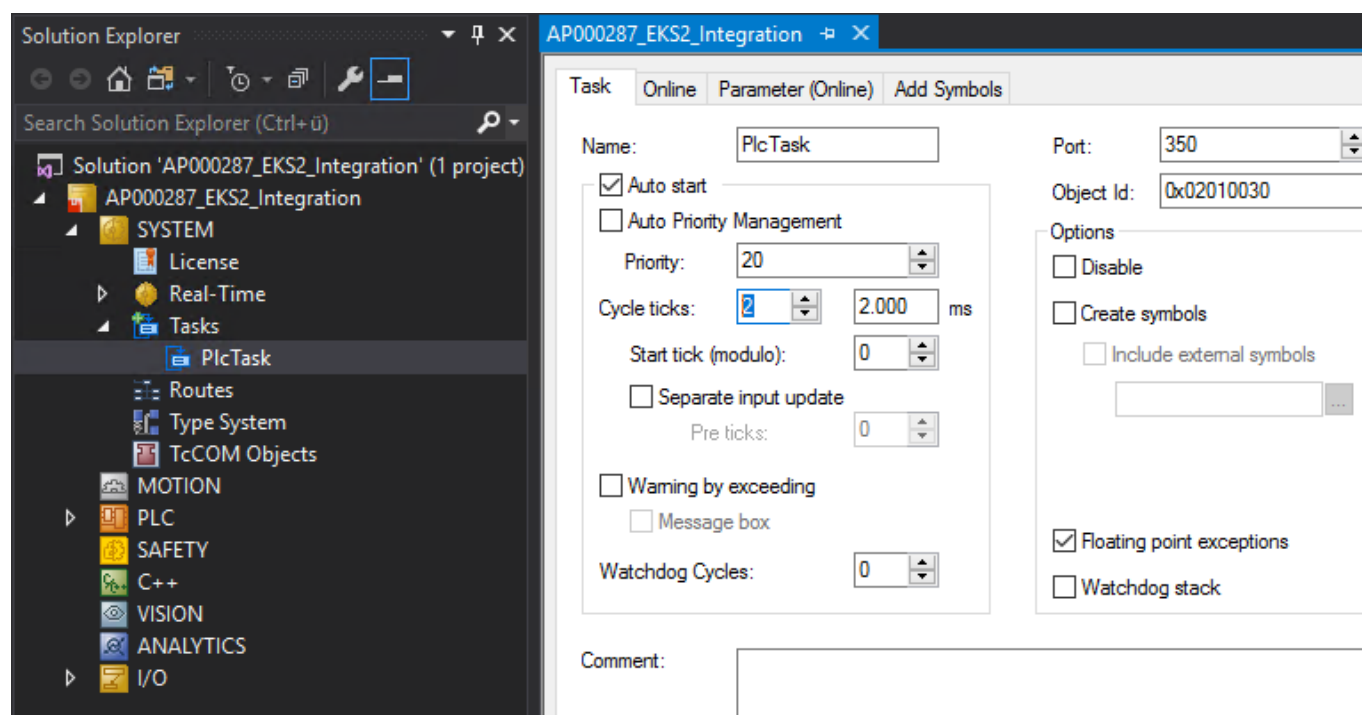


Bild 2: Parametrierung der Steuerung

4.3. Einfügen des PROFINET-Controllers

Einrichtung des PROFINET-Controllers

1. Im Solution Explorer unter I/O mit der rechten Maustaste auf Devices klicken und die Funktion Scan auswählen.

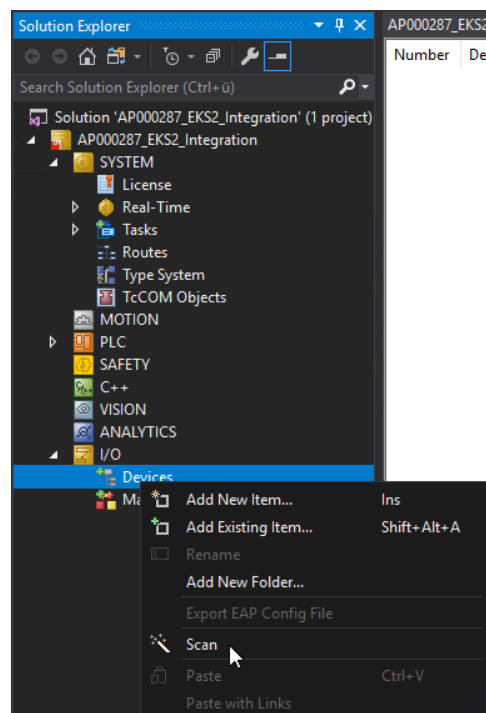


Bild 3: Einfügen PROFINET-Netzwerk



HINWEIS!

TwinCAT muss sich für Scan-Vorgänge im Config Mode befinden.

2. Den PROFINET-Controller auswählen und mit OK bestätigen.

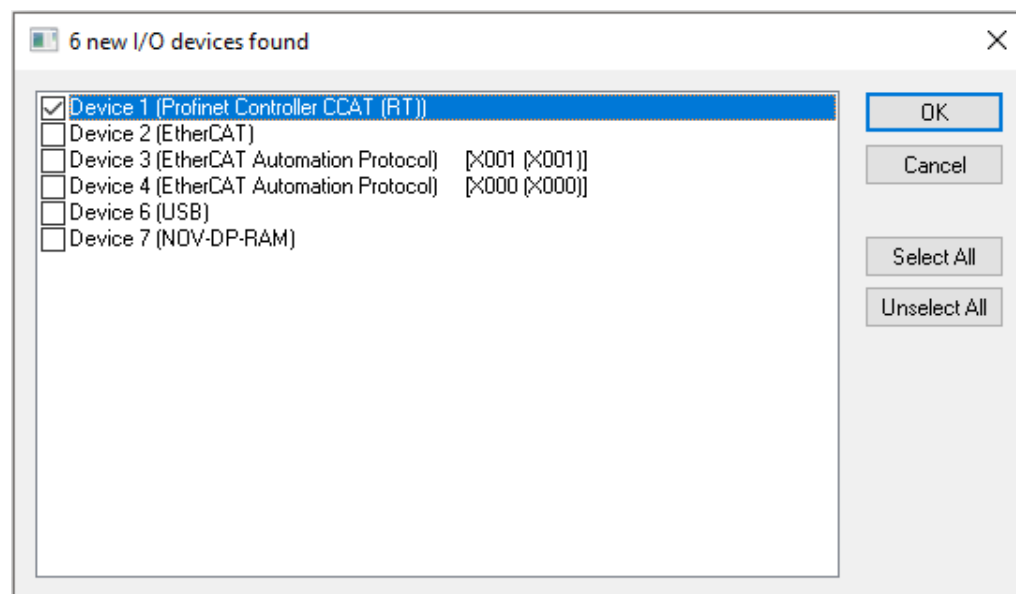


Bild 4: Auswahl PROFINET-Controller

- Die Abfrage zur automatischen Suche weiterer Geräte mit *Nein* beantworten. Eine eindeutige Zuordnung des EKS2 ist bei der automatischen Suche nicht sichergestellt.

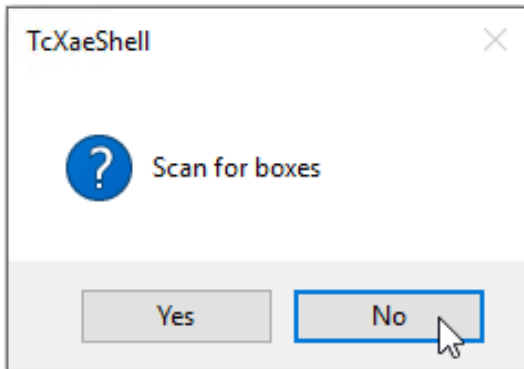


Bild 5: Nach Geräten suchen ablehnen

4.4. Einfügen des EKS2 mit PROFINET-Schnittstelle

- Mit der rechten Maustaste auf den PROFINET-Controller klicken und die Funktion *Add New Item...* auswählen..

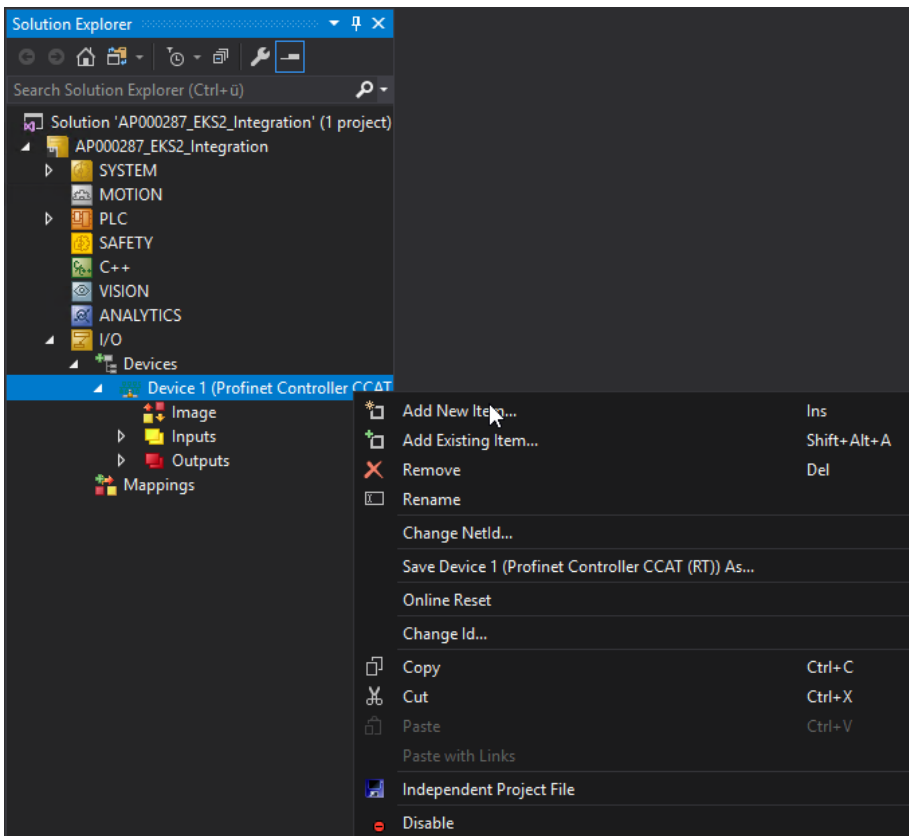


Bild 6: Hinzufügen eines Geräts

2. Die entsprechende GSDML-Datei auswählen.

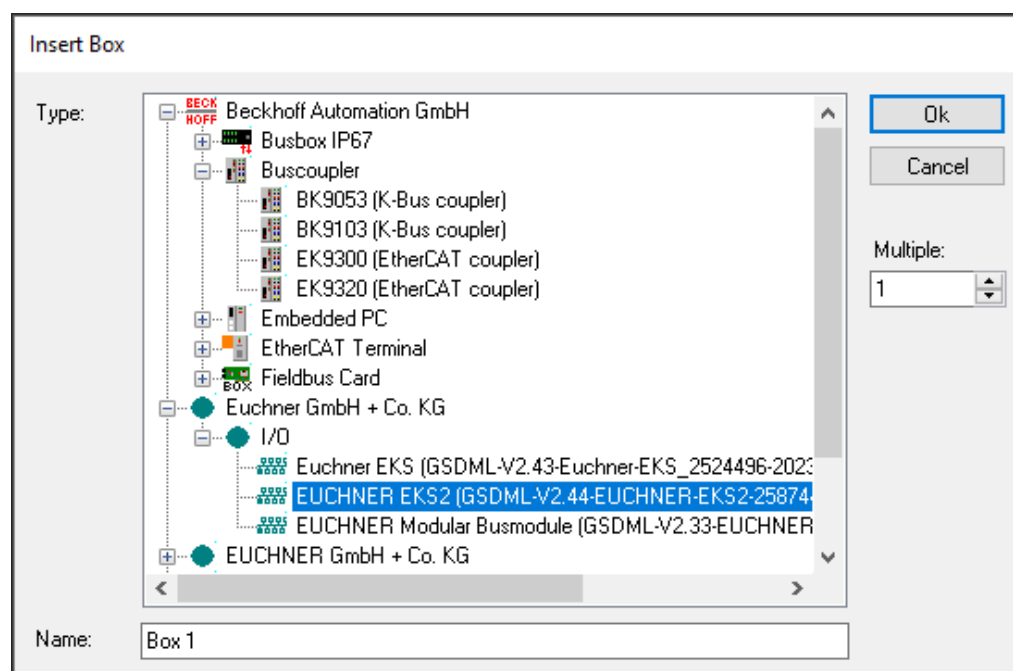


Bild 7: Auswahl der GSDML-Datei

4.5. Parametrierung des EKS2

4.5.1. Geräte- und IP-Adresse

Die folgende Konfiguration legt die PROFINET-Parameter fest. Hierzu die Eigenschaften des EKS2-Geräts im *Solution Explorer* per Doppelklick öffnen. Danach den Reiter *Device* auswählen.

- › Geräte- und IP-Adresse
- › IP-Adresse: fest

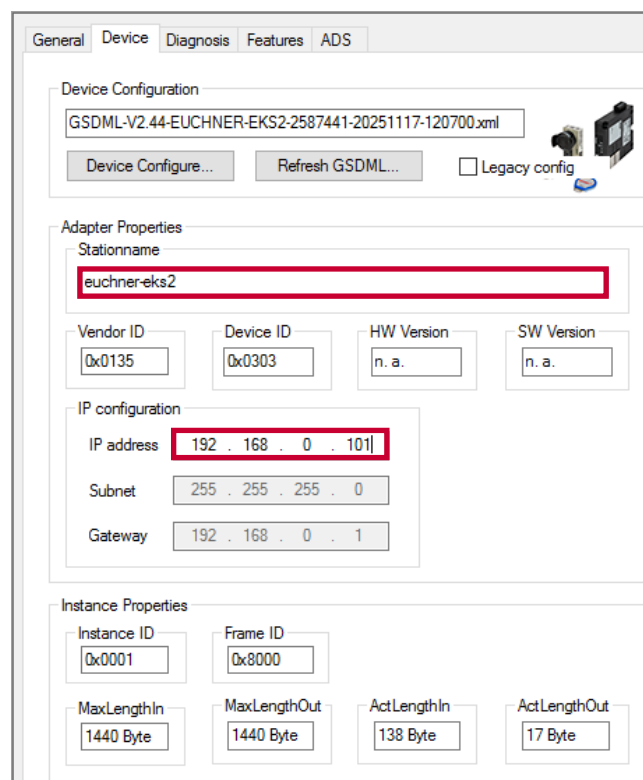


Bild 8: PROFINET-Parameter

4.6. Module und Submodule

Folgende Module stehen zur Auswahl und sind entweder als Standardkonfiguration bereits in der GSDML des Auswertegeräts enthalten oder können optional hinzugefügt werden:

Modul	Parameter [Default-Wert]	Beschreibung	Slot
EKS2 Diagnose Basic	Einstellung Alarme: Aktiviert/Deaktiviert [Deaktiviert] Einstellung Datenformat: Little/Big Endian [Big Endian]	▸ Lesen von Statusmeldungen ▸ Parametrieren der Alarme ▸ Parametrieren des Datenformats	1
EKS2 Diagnose Extended	Einstellung Alarme: Aktiviert/Deaktiviert [Deaktiviert] Einstellung Datenformat: Little/Big Endian [Big Endian]	▸ Lesen von Status- und Diagnosemeldungen ▸ Parametrieren der Alarme ▸ Parametrieren des Datenformats	1
Read: Transponder UID Standard mode with 8 bytes		▸ Aktivieren des Standard-Modus, siehe Betriebsanleitung ▸ Lesen der eindeutigen Seriennummer (UID) des Transponders	2
Read: Transponder UID-only mode with 8 bytes		▸ Aktivieren des UID-only Modus, siehe Betriebsanleitung ▸ Lesen der eindeutigen Seriennummer (UID) des Transponders ▸ Keine Betriebsartenwahl möglich	2
Read: EU001 Base Module	Automatische Datenauswertung: Ja/Nein [Ja]	Automatische Datenauswertung aktivieren	3.1
	Firma: [0] Werk: [0] Abteilung: [0]	Diese Felder werden hierarchisch nacheinander ausgewertet.	
	Kostenstelle: 1 - 5 [0]	Diese Felder sind mit einer ODER-Verknüpfung verbunden	
	Ausnahmewert: 1, 2 [0]	Die Werte in diesen Felder sind den anderen Felder in der Auswertung übergeordnet.	
	Ablaufdatum verwenden: Ja/Nein [Nein]	▸ Interne Auswertung des Ablaufdatums des Transponders ▸ Zusätzlich muss das aktuelle Datum in der Steuerung definiert und über einen azyklischen Befehl an das Auswertegerät gesendet werden.	
	Automatisches Schlüssel-Management: Aktiviert/Deaktiviert [Deaktiviert]	Automatisches Schlüssel-Management aktivieren	
Read: EU001 Header data with 18 bytes		Zusätzliches Lesen der Header-Daten in der Steuerung	3.2
Read: EU001 User data with 16 bytes		Lesen von Nutzdaten mit 16 Bytes ¹⁾	3.3
Read: EU001 User data with 32 bytes	Start-Adresse: [26]	Lesen von Nutzdaten mit 32 Bytes ¹⁾	3.3
Read: EU001 User data with 64 bytes	Anzahl der zu lesenden Bytes: [abhängig vom Modul]	Lesen von Nutzdaten mit 64 Bytes ¹⁾	3.3
Read: EU001 User data with 90 bytes		Lesen von Nutzdaten mit 90 Bytes ¹⁾	3.3
Read: EU000 Base Module	Ablaufdatum verwenden: Ja/Nein [Nein]	▸ Interne Auswertung des Ablaufdatums des Transponders ▸ Zusätzlich muss das aktuelle Datum in der Steuerung definiert und über einen azyklischen Befehl an das Auswertegerät gesendet werden.	3.1
	Start-Adresse des Ablaufdatums: [0]	Start-Adresse des Ablaufdatums	
Read: EU000 User data with 16 bytes		Lesen von Nutzdaten mit 16 Bytes	3.2
Read: EU000 User data with 32 bytes	Start-Adresse: [0]	Lesen von Nutzdaten mit 32 Bytes	
Read: EU000 User data with 64bytes	Anzahl der zu lesenden Bytes: [abhängig vom Modul]	Lesen von Nutzdaten mit 64 Bytes	
Read: EU000 User data with 116 bytes		Lesen von Nutzdaten mit 116 Bytes (Default)	
Read/Write: EU001 MO Module	Maschinengruppe: 1-4 [1] Betriebsart nach Systemstart: MO0-MO5 [MO0]	▸ Nur kompatibel mit EU001 Base Module ▸ Einstellen der Maschinengruppe ▸ Parametrieren der Betriebsartenwahl im Auswertegerät	4
Read/Write: EU000 MO Module	Start-Adresse des MO-Werts: [0] Betriebsart nach Systemstart: MO0-MO5 [MO0]	▸ Nur kompatibel mit EU000 Base Module ▸ Parametrieren der Betriebsartenwahl im Auswertegerät	

¹⁾ Die ersten vier Bytes enthalten die Personalnummer.
Die Standardmodule werden fett dargestellt.

Tabelle 1: Module EKS2

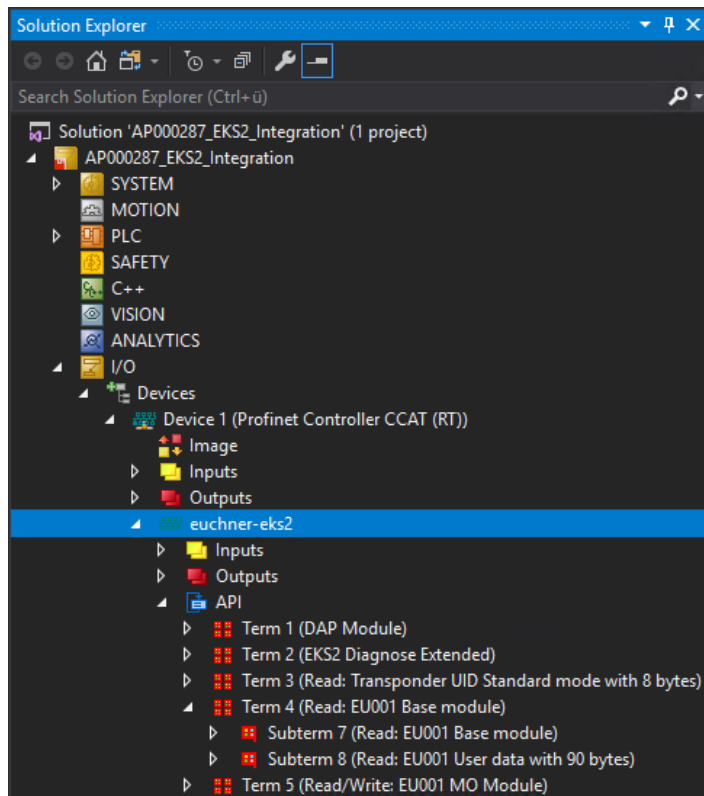


Bild 9: Standard-Konfiguration

4.7. Datenstruktur EU001

4.7.1. Konfiguration eines Moduls

Das optionale Submodul *Read: EU001 Header data with 18 bytes* ergänzt die Standardkonfiguration zur Veranschaulichung der Konfiguration. Die Steuerung liest damit die Header-Daten aus und stellt sie dar. Anwendungen ohne Header-Daten benötigen die folgenden Erläuterungen nicht.

1. Die Eigenschaften des EKS2 im *Solution Explorer* per Doppelklick öffnen. Anschließend den Reiter *Device* auswählen und die Funktion *Device Configure...* aufrufen.

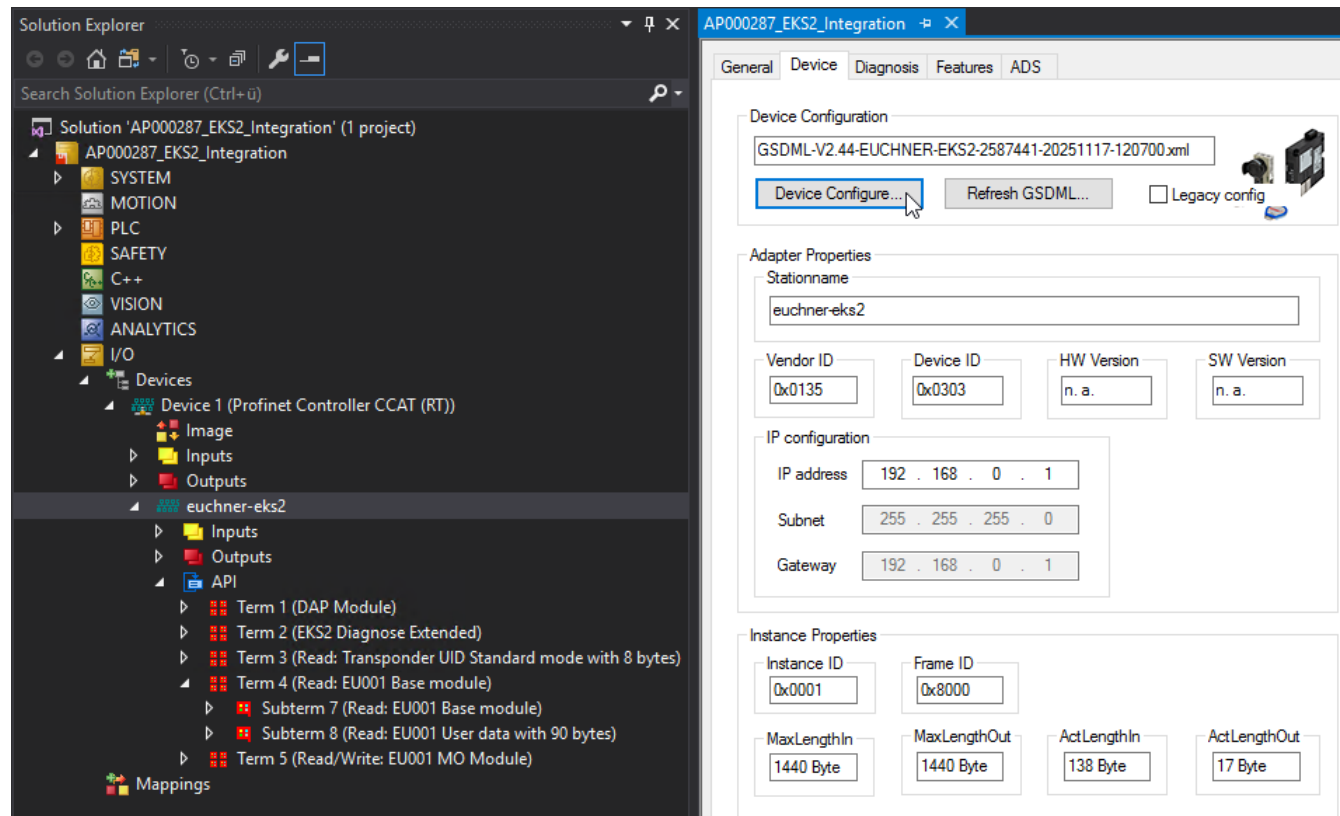


Bild 10: Gerätekonfiguration öffnen

2. Auf der linken Seite *Slot 3 (1)* auswählen. Danach das Modul *Read: EU001 Header data with 18 bytes* aus der Liste unten rechts (2) in den freien *Subslot 2* in der Mitte (3) ziehen.

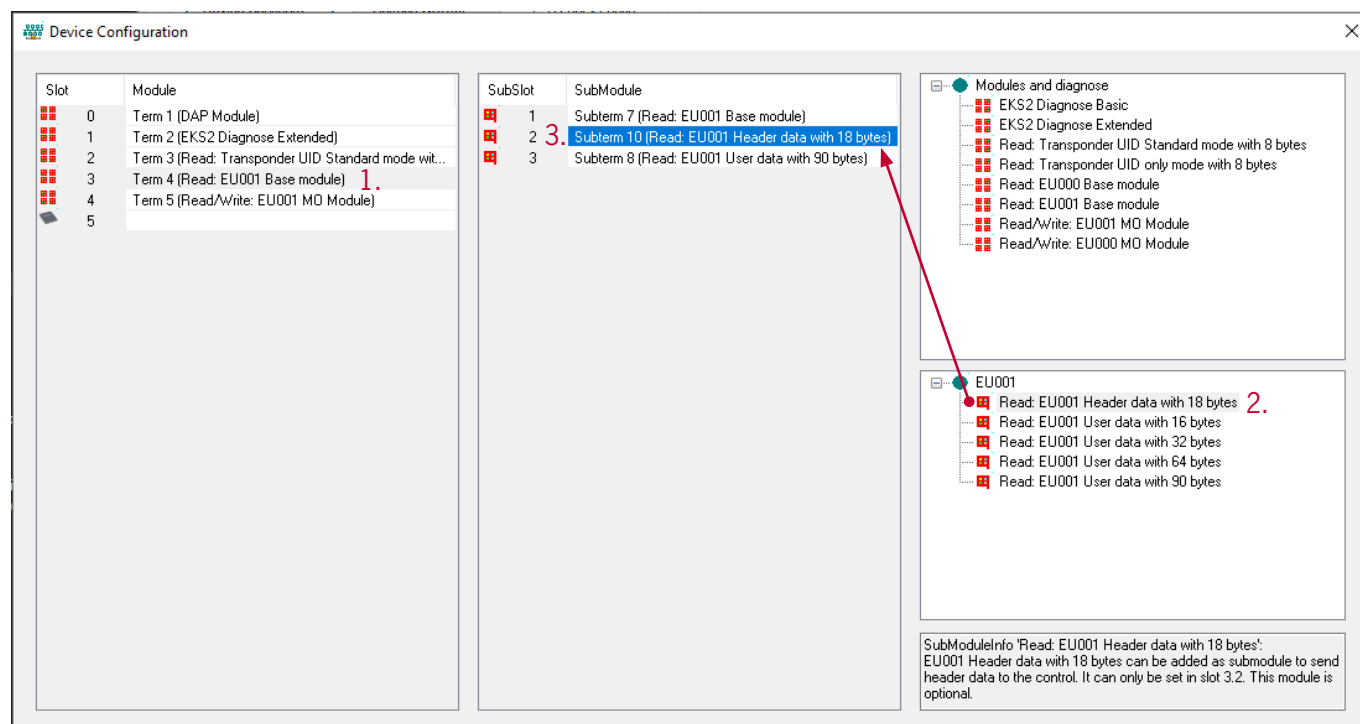


Bild 11: Hinzufügen eines Submoduls

4.7.2. Parametrierung der Endianness

Die Endianness muss zur korrekten Interpretation der im EKS2-Schlüssel gespeicherten Daten zur verwendeten Steuerung passen. Diese Applikation verwendet eine BECKHOFF-Steuerung mit Little-Endian-Format. Daher die Einstellung *Little Endian* auswählen.

Im Solution Explorer das projektierte EKS2-Gerät auswählen.

Anschließend die Struktur über *API -> Term x (EKS2 Diagnose Extended) -> Subterm x (EKS2 Diagnose Extended)* erweitern. Den Subterm per Doppelklick öffnen. Danach den Reiter *Parameterize Module* auswählen.

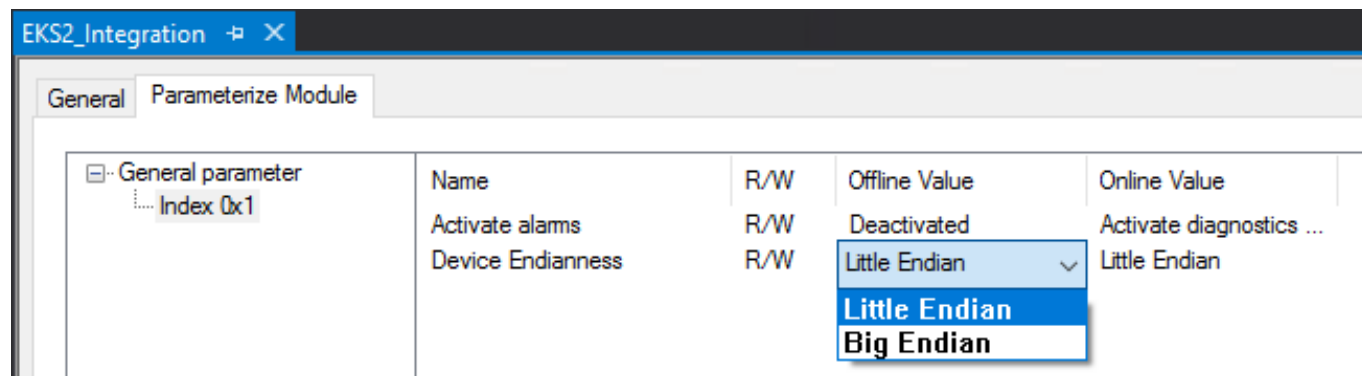


Bild 12: Parametrierung Endianness



HINWEIS

Die Übertragung von Parametern erfolgt ausschließlich über die Schaltfläche *Activate Configuration*.

4.8. Datenstruktur EU000

4.8.1. Konfiguration der Module und Submodule

Für die Verwendung der Datenstruktur EU000 müssen zwei vorkonfigurierte Module durch die zugehörigen EU000-Module ersetzt werden.

1. Die Eigenschaften des EKS2 im *Solution Explorer* per Doppelklick öffnen. Anschließend den Reiter *Device* auswählen und die Funktion *Device Configure...* aufrufen.

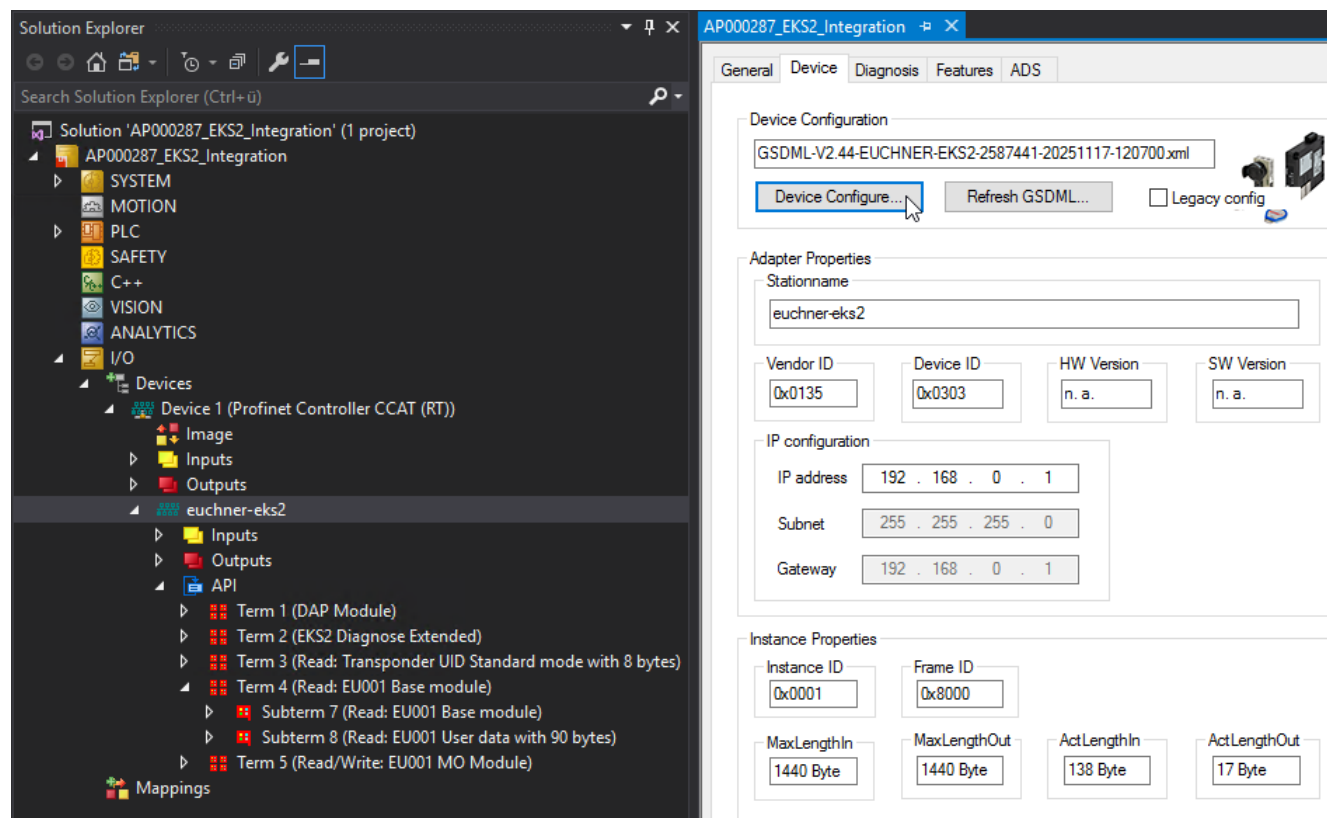


Bild 13: Gerätekonfiguration öffnen

- Auf der linken Seite alle vorkonfigurierten Module mit der Bezeichnung *EU001* löschen. Anschließend die entsprechenden *EU000*-Module im rechten oberen Fenster auswählen und den freien Slots auf der linken Seite zuweisen.



Bild 14: Konfigurierte EU000 Module



HINWEIS

Das Submodul *Read: EU000 User data with 116 bytes* im Modul *Read: EU000 Base module* ist bereits vorkonfiguriert. Werden weniger frei programmierbare Bytes benötigt, kann das Submodul durch kleinere Submodule ersetzt werden.



HINWEIS

Die Parametrierung der konfigurierten Submodule richtet sich nach den Anforderungen der Anwendung. *Tabelle 1* enthält die zugehörigen Parameter.

5. Verwenden der BECKHOFF Bibliothek

Die Bibliothek stellt Vorlagen für die einfache Integration des EKS2 in die Anwendung bereit.

Der Downloadbereich unter www.euchner.com im Bereich Applikationen enthält die EKS2-Bibliothek.

5.1. Installation der Bibliothek

1. Im Reiter *PLC* die *Library Repository* öffnen.
2. Anschließend die Bibliothek über die Funktion *Install...* aus dem entsprechenden Verzeichnis laden.

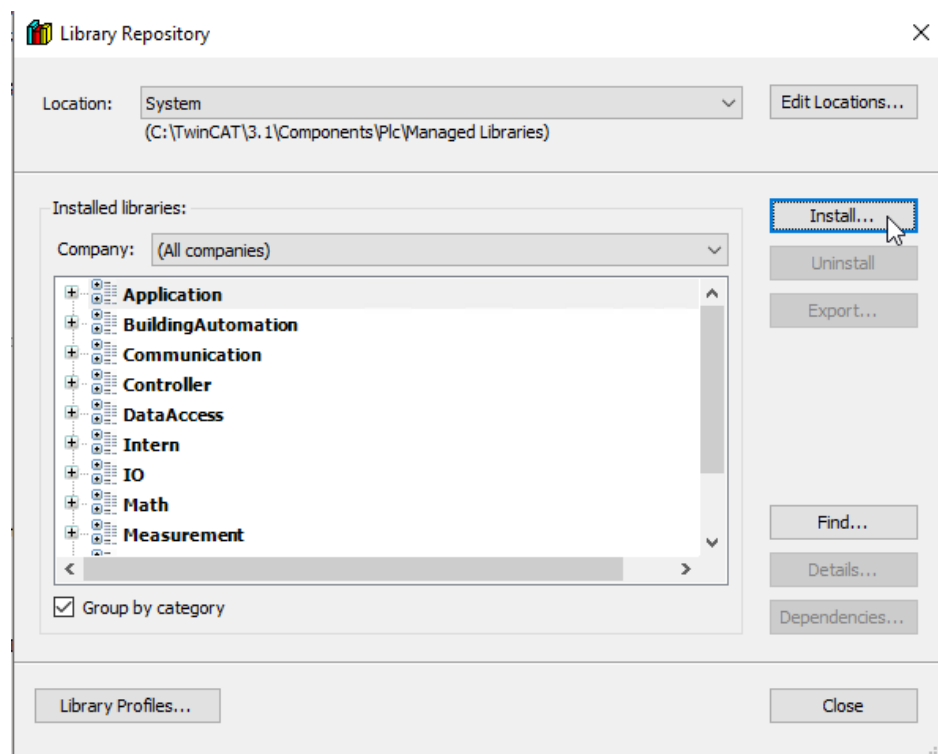


Bild 15: Bibliothek installieren



WICHTIG!

Auswertegeräte EKS2 ab **Version 1.2.0** unterstützen die Bibliothek.



HINWEIS

Sobald die Bibliothek installiert ist, wird sie unter (Miscellaneous) angezeigt.

3. Die Bibliothek im *Solution Explorer* über *References* -> *Add library...* per Rechtsklick zum Projekt hinzufügen.

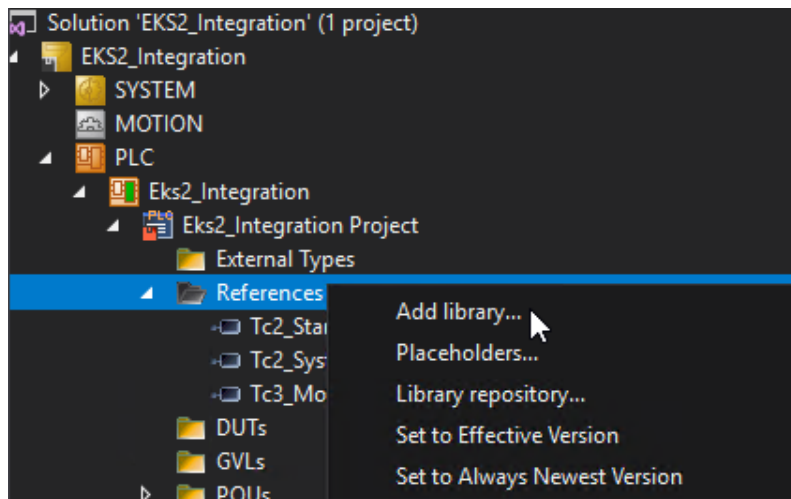


Bild 16: Bibliothek dem Projekt hinzufügen

4. Die von EUCHNER bereitgestellte Bibliothek auswählen.

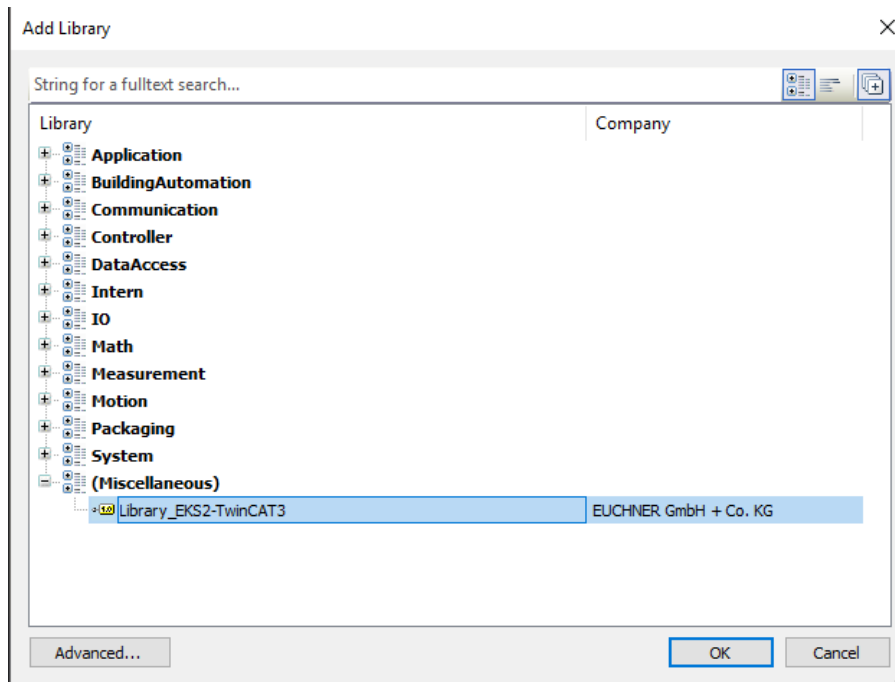


Bild 17: Bibliothek der Bibliothek

5.2. Datenstruktur EU001

5.2.1. Beschreibung der Bausteinschnittstelle

Die nachfolgende Tabelle beschreibt die Parameter des Bausteins *FB_EKS2_Eu001Base*.

	Variable	Verwendung	Datentyp	Beschreibung
<pre> 322 FB_EKS2_Eu001Base 322 ITF_Eu001DiagnoseBasic eu001DiagnoseBasic 322 ITF_Eu001DiagnoseExt eu001DiagnoseExt 322 eu001DiagnoseToEks2 eu001Uid 322 ITF_Eu001Uid eu001HeaderData 322 ITF_Eu001HeaderData eu001UserData16Bytes 322 ITF_Eu001UserData16Bytes eu001UserData32Bytes 322 ITF_Eu001UserData32Bytes eu001UserData64Bytes 322 ITF_Eu001UserData64Bytes eu001UserData90Bytes 322 ITF_Eu001Mo eu001MoFromEks2 322 eu001MoToEks2 </pre>	ITF_Eu001DiagnoseBasic	INPUT	ITF_EKS2_Eu001DiagnoseBasic	Schnittstelle für Diagnose Basic Modul
	ITF_Eu001DiagnoseExt	INPUT	ITF_EKS2_Eu001DiagnoseExt	Schnittstelle für Diagnose Extended Modul
	eu001DiagnoseToEks2	INPUT	ST_EKS2_Structure_Output_Eu001Diagnose	Ausgangsdaten für Diagnose (Base und Extended) Modul
	ITF_Eu001Uid	INPUT	ITF_EKS2_Eu001Uid	Schnittstelle für UID Modul
	ITF_Eu001HeaderData	INPUT	ITF_EKS2_Eu001HeaderData	Schnittstelle für Header Data Modul
	ITF_Eu001UserData16Bytes	INPUT	ITF_EKS2_Eu001UserData16Bytes	Schnittstelle für User Data 16 Bytes Modul
	ITF_Eu001UserData32Bytes	INPUT	ITF_EKS2_Eu001UserData32Bytes	Schnittstelle für User Data 32 Bytes Modul
	ITF_Eu001UserData64Bytes	INPUT	ITF_EKS2_Eu001UserData64Bytes	Schnittstelle für User Data 64 Bytes Modul
	ITF_Eu001UserData90Bytes	INPUT	ITF_EKS2_Eu001UserData90Bytes	Schnittstelle für User Data 90 Bytes Modul
	ITF_Eu001Mo	INPUT	ITF_EKS2_Eu001Mo	Schnittstelle für MO Modul
	eu001MoToEks2	INPUT	ST_EKS2_Structure_Output_Eu001Mo	Ausgangsdaten für MO Modul
	eu001DiagnoseBasic	OUTPUT	ST_EKS2_Structure_Input_Eu001Diagnose-Basic	Strukturierte Daten Diagnose Basic
	eu001DiagnoseExt	OUTPUT	ST_EKS2_Structure_Input_Eu001Diagnose-Ext	Strukturierte Daten Diagnose Extended
	eu001Uid	OUTPUT	ST_EKS2_Structure_Input_Eu001Uid	Strukturierte Daten UID
	eu001HeaderData	OUTPUT	ST_EKS2_Structure_Input_Eu001HeaderData	Strukturierte Daten Header Data
	eu001UserData16Bytes	OUTPUT	ST_EKS2_Structure_Input_Eu001UserData-16Bytes	Strukturierte Daten User Data 16 Bytes
	eu001UserData32Bytes	OUTPUT	ST_EKS2_Structure_Input_Eu001UserData-32Bytes	Strukturierte Daten User Data 32 Bytes
	eu001UserData64Bytes	OUTPUT	ST_EKS2_Structure_Input_Eu001UserData-64Bytes	Strukturierte Daten User Data 64 Bytes
	eu001UserData90Bytes	OUTPUT	ST_EKS2_Structure_Input_Eu001UserData-90Bytes	Strukturierte Daten User Data 90 Bytes
	eu001MoFromEks2	OUTPUT	ST_EKS2_Structure_Input_Eu001Mo	Strukturierte Daten MO

Tabelle 2: Parameter des Bausteins

5.2.2. Aufruf des Bausteins FB_EKS2_Eu001Base

Den Funktionsbaustein *FB_EKS2_Eu001Base* aufrufen und entsprechend der Anwendung parametrieren. Für jedes EKS2-Gerät eine separate Instanz des Bausteins verwenden.

Die Bibliothek stellt die vorbereitete Struktur *ST_EKS2_Eu001Integration* bereit. Die Struktur enthält die Ein- und Ausgänge bereits mit den zugehörigen Datentypen. Die Verwendung der Struktur wird empfohlen.

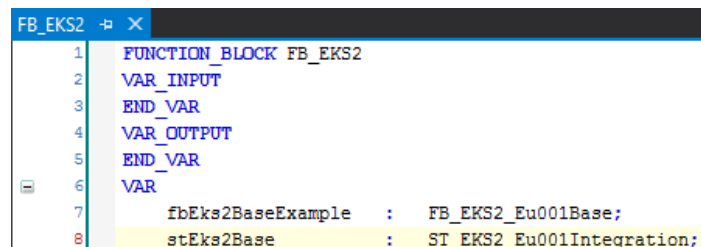


Bild 18: Bausteinschnittstelle

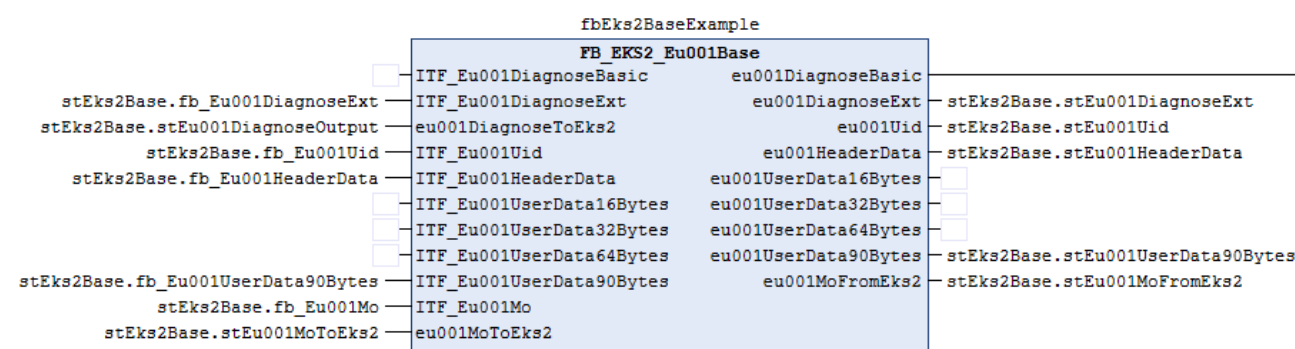


Bild 19: Parametrierter Funktionsbaustein

5.2.3. Verknüpfung des Ein- und Ausgangsbereichs

Der erste Kompilervorgang erzeugt die verknüpfbaren Prozessdaten. Das I/O-Mapping verknüpft diese Prozessdaten mit den projizierten Ein- und Ausgängen der EKS2-Hardware.

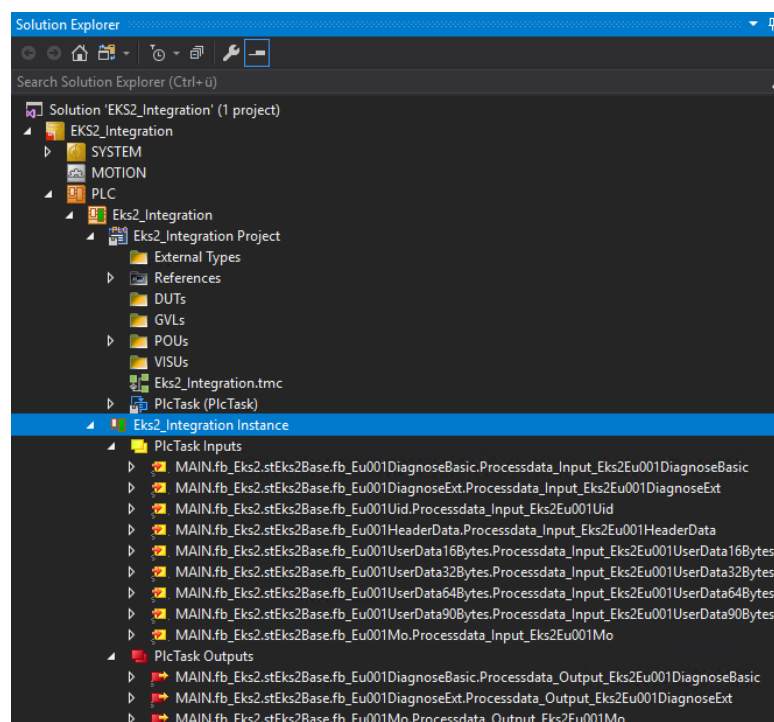


Bild 20: Erzeugte Verknüpfungen

Die folgende Applikation verwendet die nachstehend projektierten Hardware-Module. Abweichungen bei der Nummerierung der Subterms sind projektorungsbedingt.

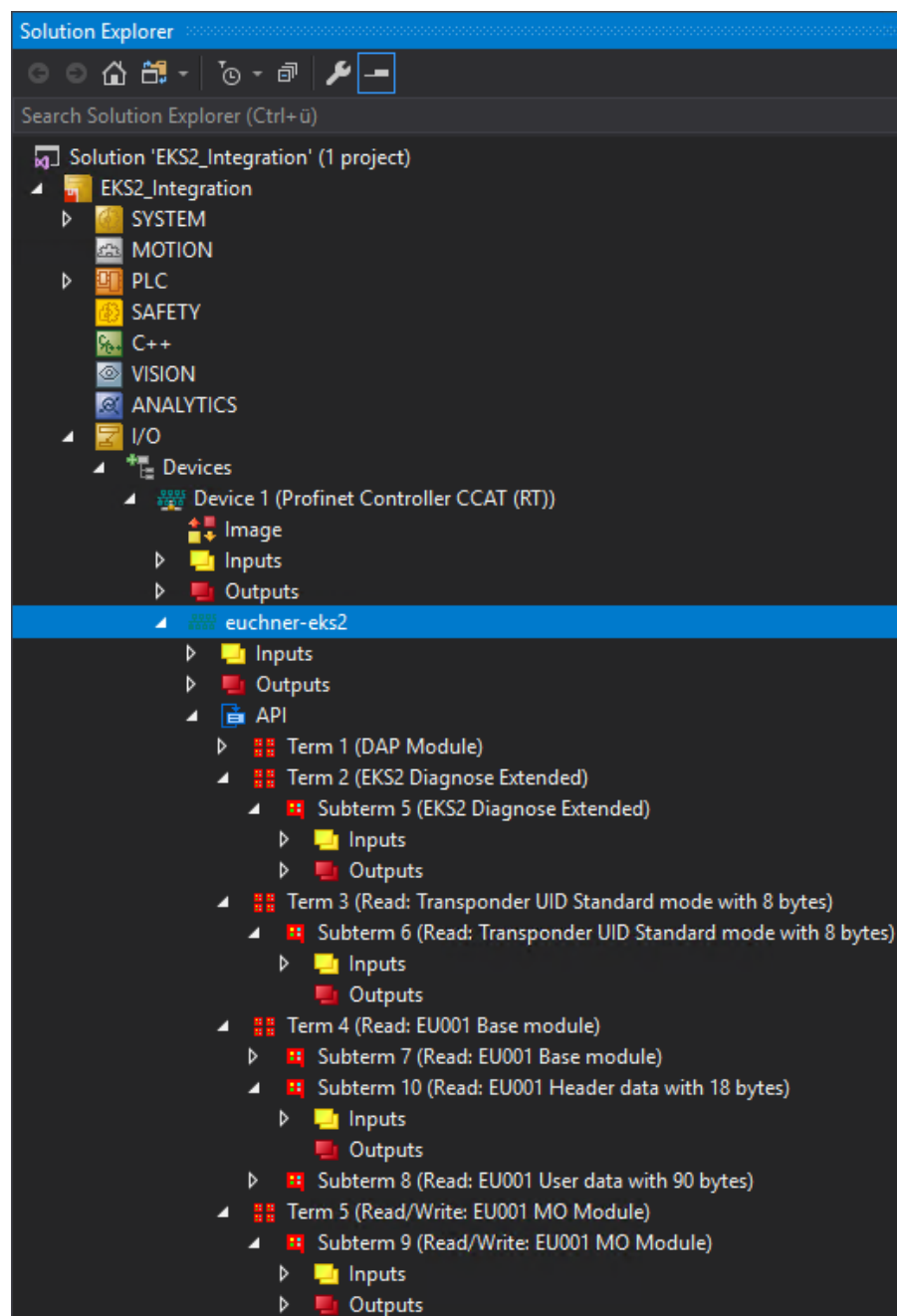


Bild 21: Projektierte Hardware

Prozessdaten	Verwendung	Projektierte Hardware
MAIN.fb_Eks2.stEks2Base.fb_Eu001DiagnoseExt.Processdata_Input_Eks2Eu001DiagnoseExt	INPUT	Subterm 5 (EKS2 Diagnose Extended)
MAIN.fb_Eks2.stEks2Base.fb_Eu001Uid.Processdata_Input_Eks2Eu001Uid	INPUT	Subterm 6 (Read: Transponder UID Standard mode with 8 bytes)
MAIN.fb_Eks2.stEks2Base.fb_Eu001HeaderData.Processdata_Input_Eks2Eu001HeaderData	INPUT	Subterm 10 (Read: EU001 Header data with 18 bytes)
MAIN.fb_Eks2.stEks2Base.fb_Eu001UserData90Bytes.Processdata_Input_Eks2Eu001UserData90Bytes	INPUT	Subterm 8 (Read: EU001 User data with 90 bytes)
MAIN.fb_Eks2.stEks2Base.fb_Eu001Mo.Processdata_Input_Eks2Eu001Mo	INPUT	Subterm 9 (Read/Write: EU001 MO Module)
MAIN.fb_Eks2.stEks2Base.fb_Eu001DiagnoseExt.Processdata_Output_Eks2Eu001DiagnoseExt	OUTPUT	Subterm 5 (EKS2 Diagnose Extended)
MAIN.fb_Eks2.stEks2Base.fb_Eu001Mo.Processdata_Output_Eks2Eu001Mo	OUTPUT	Subterm 9 (Read/Write: EU001 MO Module)

Tabelle 3: I/O-Mapping zwischen Ein- und Ausgangsprozessdaten und projektierte EKS2 Hardware

Das folgende Beispiel zeigt das I/O-Mapping für Diagnose Extended.

Den Prozessdaten-Eingang (PDI) *MAIN...Eks2Eu001DiagnoseExt* öffnen, siehe *Bild 20*. Anschließend die Funktion *Linked to...* aufrufen.

Bild 22: I/O-Mapping

Anschließend öffnet sich ein weiteres Fenster.

Dort den entsprechenden Eingang des EKS2 für das Modul *EKS2 Diagnose Extended* auswählen.

Bild 23: Auswahl des projektierten EKS2 Eingangsmoduls

Das I/O-Mapping für alle Ein- und Ausgangsprozessdaten entsprechend der projektierten Hardware durchführen (*Tabelle 3*).

5.2.4. EKS2 Online Daten

Nach der Aktivierung der Konfiguration eine Online-Verbindung zur Steuerung herstellen und die Bausteinschnittstelle öffnen. Das nachfolgende Bild zeigt die aufbereiteten Daten am Beispiel der erweiterten Diagnosedaten und der UID des Schlüssels.

FB_EKS2 [Online] - X					
EKS2_Integration.Eks2_Integration.MAIN.fb_Eks2					
Expression	Type	Value	Prepared value	Address	Comment
fbEks2BaseExample	FB_EKS2_Eu001Base				
stEks2Base	ST_EKS2_Eu001Integration				
fb_Eu001DiagnoseBasic	FB_EKS2_Eu001DiagnoseBasic				Input: Function block associated with the API
fb_Eu001DiagnoseExt	FB_EKS2_Eu001DiagnoseExt				Input: Function block associated with the API
fb_Eu001Uid	FB_EKS2_Eu001Uid				Input: Function block associated with the API
fb_Eu001HeaderData	FB_EKS2_Eu001HeaderData				Input: Function block associated with the API
fb_Eu001UserData16Bytes	FB_EKS2_Eu001UserData16Bytes				Input: Function block associated with the API
fb_Eu001UserData32Bytes	FB_EKS2_Eu001UserData32Bytes				Input: Function block associated with the API
fb_Eu001UserData64Bytes	FB_EKS2_Eu001UserData64Bytes				Input: Function block associated with the API
fb_Eu001UserData90Bytes	FB_EKS2_Eu001UserData90Bytes				Input: Function block associated with the API
fb_Eu001Mo	FB_EKS2_Eu001Mo				Input: Function block associated with the API
stEu001DiagnoseOutput	ST_EKS2_Structure_Output_Eu001Diagnose				Input: Data to be written to EKS2
stEu001MoToEks2	ST_EKS2_Structure_Output_Eu001Mo				Input: Data to be written to EKS2
stEu001DiagnoseBasic	ST_EKS2_Structure_Input_Eu001DiagnoseBasic				Output: Available data
stEu001DiagnoseExt	ST_EKS2_Structure_Input_Eu001DiagnoseExt				Output: Available data
deviceReadyForOperation	BIT	TRUE			
electronicKeyDetected	BIT	TRUE			
acyclicalJobError	BIT	FALSE			
readyForAcyclicalJob	BIT	TRUE			
deviceError	BIT	FALSE			
cyclicalJobError	BIT	FALSE			
readyForCyclicalJob	BIT	TRUE			
cyclicalJobInProgress	BIT	FALSE			
noUserAreaIncluded	BIT	FALSE			
selectionOfSafeOperatingModeError	BIT	FALSE			
nc_1	BIT	FALSE			
nc_2	BIT	FALSE			
nc_3	BIT	FALSE			
electronicKeyInvalid	BIT	FALSE			
electronicKeyValid	BIT	TRUE			
nc_4	BIT	FALSE			
extDiagnosticByte	WORD	0			
stEu001Uid	ST_EKS2_Structure_Input_Eu001Uid				Output: Available data
uniqueId	ARRAY [0..7] OF BYTE				
uniqueId[0]	BYTE	16#04			
uniqueId[1]	BYTE	16#1F			
uniqueId[2]	BYTE	16#39			
uniqueId[3]	BYTE	16#2A			
uniqueId[4]	BYTE	16#C7			
uniqueId[5]	BYTE	16#10			
uniqueId[6]	BYTE	16#90			
uniqueId[7]	BYTE	16#00			

Bild 24: EKS2 Online Daten

5.3. Datenstruktur EU000

5.3.1. Beschreibung der Bausteinschnittstelle

Die nachfolgende Tabelle beschreibt die Parameter des Bausteins *FB_EKS2_Eu000Base*.

	Variable	Verwendung	Datentyp	Beschreibung
<pre> 222 FB_EKS2_Eu000Base 222 ITF_Eu000DiagnoseBasic eu000DiagnoseBasic 222 ITF_Eu000DiagnoseExt eu000DiagnoseExt 222 eu000DiagnoseToEks2 eu000UId 222 ITF_Eu000UId eu000UserData16Bytes 222 ITF_Eu000UserData16Bytes eu000UserData32Bytes 222 ITF_Eu000UserData32Bytes eu000UserData64Bytes 222 ITF_Eu000UserData64Bytes eu000UserData116Bytes 222 ITF_Eu000Mo eu000MoFromEks2 222 eu000MoToEks2 </pre>	ITF_Eu000DiagnoseBasic	INPUT	ITF_EKS2_Eu000DiagnoseBasic	Schnittstelle für Diagnose Basic Modul
	ITF_Eu000DiagnoseExt	INPUT	ITF_EKS2_Eu000DiagnoseExt	Schnittstelle für Diagnose Extended Modul
	eu000DiagnoseToEks2	INPUT	ST_EKS2_Structure_Output_Eu000Diagnose	Ausgangsdaten für Diagnose (Base und Extended) Modul
	ITF_Eu000UId	INPUT	ITF_EKS2_Eu000UId	Schnittstelle für UID Modul
	ITF_Eu000UserData16Bytes	INPUT	ITF_EKS2_Eu000UserData16Bytes	Schnittstelle für User Data 16 Bytes Modul
	ITF_Eu000UserData32Bytes	INPUT	ITF_EKS2_Eu000UserData32Bytes	Schnittstelle für User Data 32 Bytes Modul
	ITF_Eu000UserData64Bytes	INPUT	ITF_EKS2_Eu000UserData64Bytes	Schnittstelle für User Data 64 Bytes Modul
	ITF_Eu000UserData116Bytes	INPUT	ITF_EKS2_Eu000UserData116Bytes	Schnittstelle für User Data 116 Bytes Modul
	ITF_Eu000Mo	INPUT	ITF_EKS2_Eu000Mo	Schnittstelle für MO Modul
	eu000MoToEks2	INPUT	ST_EKS2_Structure_Output_Eu000Mo	Ausgangsdaten für MO Modul
	eu000DiagnoseBasic	OUTPUT	ST_EKS2_Structure_Input_Eu000Diagnose-Basic	Strukturierte Daten Diagnose Basic
	eu000DiagnoseExt	OUTPUT	ST_EKS2_Structure_Input_Eu000Diagnose-Ext	Strukturierte Daten Diagnose Extended
	eu000UId	OUTPUT	ST_EKS2_Structure_Input_Eu000UId	Strukturierte Daten UID
	eu000UserData16Bytes	OUTPUT	ST_EKS2_Structure_Input_Eu000UserData-16Bytes	Strukturierte Daten User Data 16 Bytes
	eu000UserData32Bytes	OUTPUT	ST_EKS2_Structure_Input_Eu000UserData-32Bytes	Strukturierte Daten User Data 32 Bytes
	eu000UserData64Bytes	OUTPUT	ST_EKS2_Structure_Input_Eu000UserData-64Bytes	Strukturierte Daten User Data 64 Bytes
	eu000UserData116Bytes	OUTPUT	ST_EKS2_Structure_Input_Eu000UserData-116Bytes	Strukturierte Daten User Data 116 Bytes
	eu000MoFromEks2	OUTPUT	ST_EKS2_Structure_Input_Eu000Mo	Strukturierte Daten MO

Tabelle 4: Parameter des Bausteins

5.3.2. Aufruf des Bausteins FB_EKS2_Eu000Base

Den Funktionsbaustein *FB_EKS2_Eu000Base* aufrufen und entsprechend der Anwendung parametrieren. Für jedes EKS2-Gerät eine separate Instanz des Bausteins verwenden.

Die Bibliothek stellt die vorbereitete Struktur *ST_EKS2_Eu000Integration* bereit. Die Struktur enthält die Ein- und Ausgänge bereits mit den zugehörigen Datentypen. Die Verwendung der Struktur wird empfohlen.

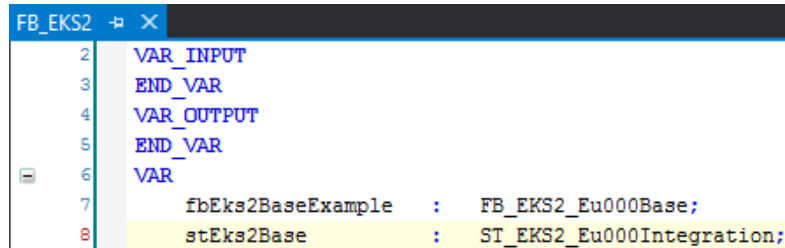


Bild 25: Bausteinschnittstelle

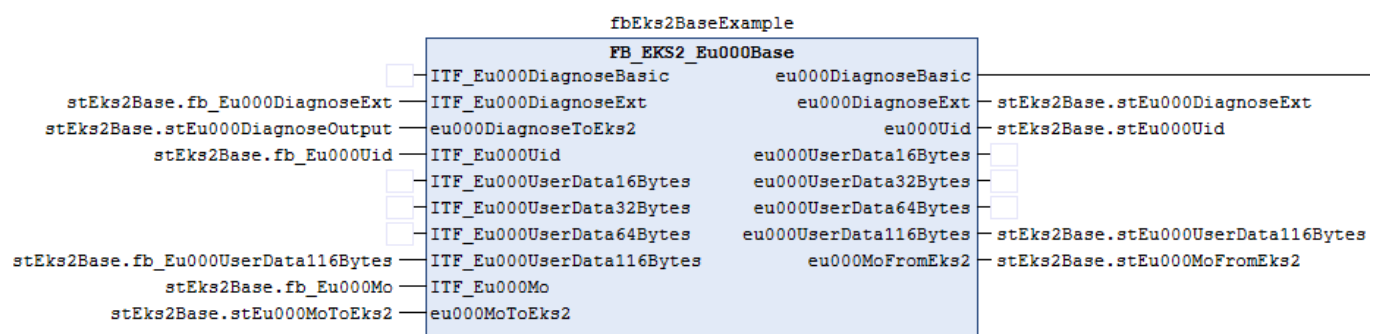


Bild 26: Parametrierter Funktionsbaustein

5.3.3. Verknüpfung des Ein- und Ausgangsbereichs

Der erste Kompiliervorgang erzeugt die verknüpfbaren Prozessdaten. Das I/O-Mapping verknüpft diese Prozessdaten mit den projizierten Ein- und Ausgängen der EKS2-Hardware.

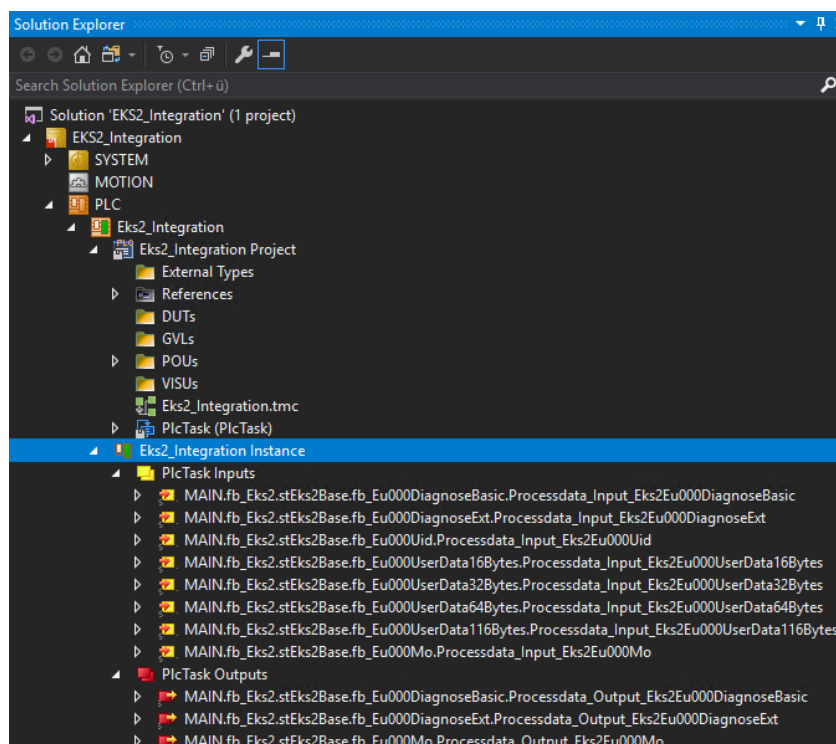


Bild 27: Erzeugte Verknüpfungen

Die folgende Applikation verwendet die nachstehend projektierten Hardware-Module. Abweichungen bei der Nummerierung der Subterms sind projektorungsbedingt.

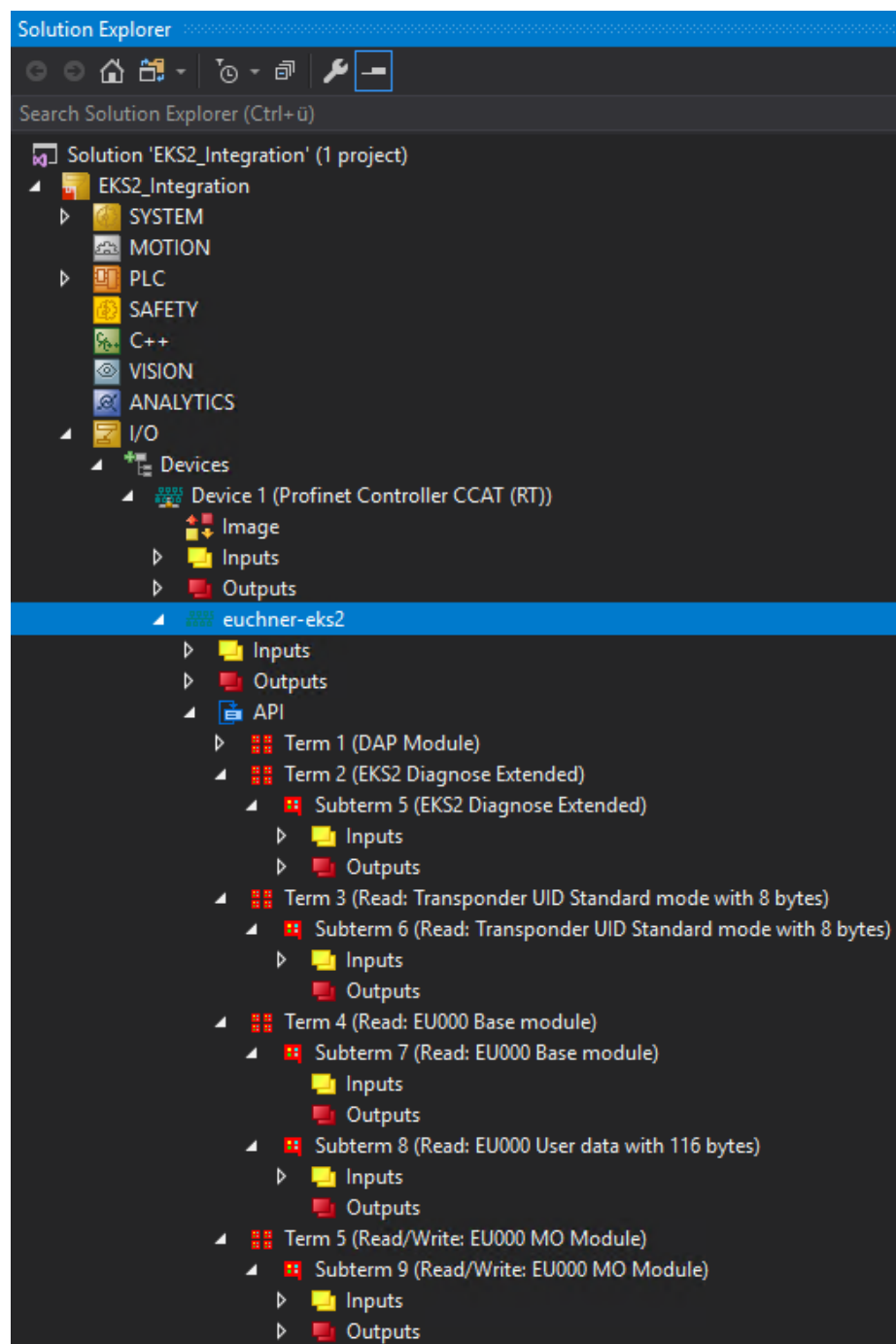


Bild 28: Projektierte Hardware

Prozessdaten	Verwendung	Projektierte Hardware
MAIN.fb_Eks2.stEks2Base.fb_Eu000DiagnoseExt.Processdata_Input_Eks2Eu000DiagnoseExt	INPUT	Subterm 5 (EKS2 Diagnose Extended)
MAIN.fb_Eks2.stEks2Base.fb_Eu000Uid.Processdata_Input_Eks2Eu000Uid	INPUT	Subterm 6 (Read: Transponder UID Standard mode with 8 bytes)
MAIN.fb_Eks2.stEks2Base.fb_Eu000UserData116Bytes.Processdata_Input_Eks2Eu000UserData116Bytes	INPUT	Subterm 8 (Read: EU000 User data with 116 bytes)
MAIN.fb_Eks2.stEks2Base.fb_Eu000Mo.Processdata_Input_Eks2Eu000Mo	INPUT	Subterm 9 (Read/Write: EU000 MO Module)
MAIN.fb_Eks2.stEks2Base.fb_Eu000DiagnoseExt.Processdata_Output_Eks2Eu000DiagnoseExt	OUTPUT	Subterm 5 (EKS2 Diagnose Extended)
MAIN.fb_Eks2.stEks2Base.fb_Eu000Mo.Processdata_Output_Eks2Eu000Mo	OUTPUT	Subterm 9 (Read/Write: EU000 MO Module)

Tabelle 5: I/O-Mapping zwischen Ein- und Ausgangsprozessdaten und projektierte EKS2 Hardware

Das folgende Beispiel zeigt das I/O-Mapping für Diagnose Extended.

Den Prozessdaten-Eingang (PDI) *MAIN...Eks2Eu000DiagnoseExt* öffnen, siehe *Bild 27*. Anschließend die Funktion *Linked to...* aufrufen.

Bild 29: I/O-Mapping

Anschließend öffnet sich ein weiteres Fenster.

Dort den entsprechenden Eingang des EKS2 für das Modul *EKS2 Diagnose Extended* auswählen.

Bild 30: Auswahl des projektierten EKS2 Eingangmoduls

Das I/O-Mapping für alle Ein- und Ausgangsprozessdaten entsprechend der projektierten Hardware durchführen. (Tabelle 5).

5.3.4. EKS2 Online Daten

Nach der Aktivierung der Konfiguration eine Online-Verbindung zur Steuerung herstellen und die Bausteinschnittstelle öffnen. Das nachfolgende Bild zeigt die aufbereiteten Daten am Beispiel der erweiterten Diagnosedaten und der UID des Schlüssels.

EKS2_Integration.Eks2_Integration.MAIN.fb_Eks2					
Expression	Type	Value	Prepared value	Address	Comment
fbEks2BaseExample	FB_EKS2_Eu000Base				
stEks2Base	ST_EKS2_Eu000Integration				
fb_Eu000DiagnoseBasic	FB_EKS2_Eu000DiagnoseBasic				Input: Function block associated with the API
fb_Eu000DiagnoseExt	FB_EKS2_Eu000DiagnoseExt				Input: Function block associated with the API
fb_Eu000Uid	FB_EKS2_Eu000Uid				Input: Function block associated with the API
fb_Eu000UserData16Bytes	FB_EKS2_Eu000UserData16Bytes				Input: Function block associated with the API
fb_Eu000UserData32Bytes	FB_EKS2_Eu000UserData32Bytes				Input: Function block associated with the API
fb_Eu000UserData64Bytes	FB_EKS2_Eu000UserData64Bytes				Input: Function block associated with the API
fb_Eu000UserData116Bytes	FB_EKS2_Eu000UserData116Bytes				Input: Function block associated with the API
fb_Eu000Mo	FB_EKS2_Eu000Mo				Input: Function block associated with the API
stEu000DiagnoseOutput	ST_EKS2_Structure_Output_Eu000Diagnose				Input: Data to be written to EKS2
stEu000MoToEks2	ST_EKS2_Structure_Output_Eu000Mo				Input: Data to be written to EKS2
stEu000DiagnoseBasic	ST_EKS2_Structure_Input_Eu000DiagnoseBasic				Output: Available data
stEu000DiagnoseExt	ST_EKS2_Structure_Input_Eu000DiagnoseExt				Output: Available data
deviceReadyForOperation	BIT	TRUE			
electronicKeyDetected	BIT	TRUE			
acyclicalJobError	BIT	FALSE			
readyForAcyclicalJob	BIT	TRUE			
deviceError	BIT	FALSE			
cyclicalJobError	BIT	FALSE			
readyForCyclicalJob	BIT	TRUE			
cyclicalJobInProgress	BIT	FALSE			
noUserAreaIncluded	BIT	FALSE			
selectionOfSafeOperatingModeError	BIT	FALSE			
nc_1	BIT	FALSE			
nc_2	BIT	FALSE			
nc_3	BIT	FALSE			
electronicKeyInvalid	BIT	FALSE			
electronicKeyValid	BIT	TRUE			
nc_4	BIT	FALSE			
extDiagnosticByte	ARRAY [0..1] OF BYTE				
stEu000Uid	ST_EKS2_Structure_Input_Eu000Uid				Output: Available data
uniqueId	ARRAY [0..7] OF BYTE				
uniqueId[0]	BYTE	16#04			
uniqueId[1]	BYTE	16#1E			
uniqueId[2]	BYTE	16#2F			
uniqueId[3]	BYTE	16#2A			
uniqueId[4]	BYTE	16#C7			
uniqueId[5]	BYTE	16#10			
uniqueId[6]	BYTE	16#90			

Bild 31: EKS2 Online Daten

6. Wichtiger Hinweis – Bitte unbedingt sorgfältig beachten!

Dieses Dokument richtet sich an einen Konstrukteur, der die entsprechenden Kenntnisse in der Sicherheitstechnik hat und die Kenntnis der einschlägigen Normen besitzt, z.B. durch eine Ausbildung zum Sicherheitsingenieur. Nur mit entsprechender Qualifikation kann das vorgestellte Beispiel in eine vollständige Sicherheitskette integriert werden.

Das Beispiel stellt nur einen Ausschnitt aus einer vollständigen Sicherheitskette dar und erfüllt für sich allein genommen keine Sicherheitsfunktion. Zur Erfüllung einer Sicherheitsfunktion muss beispielsweise zusätzlich die Abschaltung der Energie der Gefährdungsstelle sowie auch die Software innerhalb der Sicherheitsauswertung betrachtet werden.

Die vorgestellten Applikationen stellen lediglich Beispiele zur Lösung bestimmter Sicherheitsaufgaben zur Absicherung von Schutztüren dar. Bedingt durch applikationsabhängige und individuelle Schutzziele innerhalb einer Maschine/Anlage können die Beispiele nicht erschöpfend sein.

Falls Fragen zu diesem Beispiel offen bleiben, wenden Sie sich bitte direkt an uns.

Nach der Maschinenrichtlinie 2006/42/EG ist der Konstrukteur einer Maschine bzw. Anlage verpflichtet, eine Risikobeurteilung durchzuführen und Maßnahmen zur Minderung des Risikos zu ergreifen. Er muss sich hierbei an die einschlägigen nationalen und internationalen Sicherheitsnormen halten. Normen stellen in der Regel den aktuellen Stand der Technik dar. Der Konstrukteur sollte sich daher laufend über Änderungen in den Normen informieren und seine Überlegungen darauf abstimmen, relevant für die funktionale Sicherheit sind u.a. die EN ISO 13849 und EN 62061. Diese Applikation ist immer nur als Unterstützung für die Überlegungen zu Sicherheitsmaßnahmen zu sehen.

Der Konstrukteur einer Maschine/Anlage ist verpflichtet die Sicherheitstechnik selbst zu beurteilen. Die Beispiele dürfen nicht zu einer Beurteilung herangezogen werden, da hier nur ein kleiner Ausschnitt einer vollständigen Sicherheitsfunktion sicherheitstechnisch betrachtet wurde.

Um die Applikationen der Sicherheitsschalter an Schutztüren richtig einsetzen zu können, ist es unerlässlich, dass die Normen EN ISO 13849-1, EN ISO 14119 und alle relevanten C-Normen für den jeweiligen Maschinentyp beachtet werden. Dieses Dokument ersetzt keinesfalls eine eigene Risikobeurteilung und kann auch nicht als Basis für eine Fehlerbeurteilung herangezogen werden.

Insbesondere bei einem Fehlerausschluss ist zu beachten, dass dieser nur vom Konstrukteur einer Maschine bzw. Anlage durchgeführt werden kann und dass hierzu eine Begründung notwendig ist. Ein genereller Fehlerausschluss ist nicht möglich. Nähere Auskünfte zum Fehlerausschluss gibt die EN ISO 13849-2.

Änderungen an Produkten oder innerhalb der Baugruppen von dritten Anbietern, die in diesem Beispiel verwendet werden, können dazu führen, dass die Funktion nicht mehr gewährleistet ist oder die sicherheitstechnische Beurteilung angepasst werden muss. In jedem Fall sind die Angaben in den Betriebsanleitungen sowohl seitens EUCHNER, als auch seitens der dritten Anbieter zugrunde zu legen, bevor diese Applikation in eine gesamte Sicherheitsfunktion integriert wird. Sollten hierbei Widersprüche zwischen Betriebsanleitungen und diesem Dokument auftreten, setzen Sie sich bitte mit uns direkt in Verbindung.

Verwendung von Marken- und Firmennamen

Alle aufgeführten Marken- und Firmennamen sind Eigentum des jeweiligen Herstellers. Deren Verwendung dient ausschließlich zur eindeutigen Identifikation kompatibler Peripheriegeräte und Betriebsumgebungen im Zusammenhang mit unseren Produkten.

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

Ausgabe:
AP000287-01-05/26
Titel:
Applikation EKS2
EKS2 - Einbindung EKS2 mit PROFINET-Schnittstelle in
BECKHOFF TwinCAT 3

Copyright:
© EUCHNER GmbH + Co. KG, 05/2026

Technische Änderungen vorbehalten,
alle Angaben ohne Gewähr.