

EKS-Profibus an Siemens S7-300 – EKS Schlüssel einlesen



Inhalt

Verwendete Bauteile / Module	2
EUCHNER	2
Andere.....	2
Funktionsbeschreibung.....	2
Allgemein	2
Beispiel einer Schlüsselstruktur	2
Einstellung der EKS Schlüsselaufnahme.....	3
Profibus Adresse	3
Schreibschutzeinstellung	3
Projektierung in der Steuerung	4
Hardware.....	4
Programmierung in der Steuerung	5
Globale Datenbausteine	5
AWL Programm zum Abruf des Schlüsselinhalts	6
Wichtiger Hinweis – Bitte unbedingt sorgfältig beachten!.....	10

Verwendete Bauteile / Module

EUCHNER

Beschreibung	Best.-Nr. / Artikelbezeichnung
EKS Profibus	084800 / EKS-A-IDX-G01-ST09/03
EKS Schlüssel	077859 / EKS-A-K1RDWT32-EU 084735 / EKS-A-K1BKWT32-EU 091045 / EKS-A-K1BUWT32-EU 094839 / EKS-A-K1GNWT32-EU 094840 / EKS-A-K1YEWT32-EU

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

Andere

Beschreibung	Artikel
S7-300, CPU 315F-2 PN/DP	6ES7315-2FJ14-0AB0

Funktionsbeschreibung

Allgemein

Das EKS wird über den Profibus an einer Siemens S7-300 SPS angeschlossen. Es sollen alle Daten entsprechend der folgenden Datenstruktur ausgelesen werden.

Beispiel einer Schlüsselstruktur

Die Daten auf dem Schlüssel sind wie folgt strukturiert:

Bytenr.	Beschreibung	Typ	Länge	Erläuterung
103 – 104	KEYCRC	CRC	2 Byte	Prüfsumme über einen bestimmten Teil des Schlüssels als Kopierschutz. Nähere Erläuterungen zur CRC siehe EKM Handbuch.
105 – 112	Verfallsdatum	Date	8 Byte	Verfallsdatum des Schlüssels.
113 – 114	Berechtigungsstufe	Word	2 Byte	Autorisierungsstufe für Zugriff auf die Maschine.
115	Abteilung	Byte	1 Byte	Nummer, die eine begrenzte Menge an Maschinen oder Anlagen beschreibt.
116 – 123	KeyID	KeyID	8 Byte	Die KeyID ist eine von EUCHNER fest programmierte Nummer auf dem Schlüssel. Diese Nummer ist bei jedem Schlüssel unterschiedlich. Diese Nummer kann zur Werkeridentifizierung herangezogen werden.

Die Struktur entspricht dem Applikationsbeispiel AP000169-2...

Einstellung der EKS Schlüsselaufnahme

Profibus Adresse

Das Gerät soll auf Adresse 75 eingestellt werden. Die Adresse 75 lautet binär geschrieben 1001011. Die DIP-Schalter 1 bis 7 werden dementsprechend eingestellt (DIP Schalter 1 ist das niederwertige Bit).

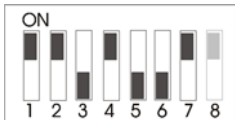


Bild 1

Schreibschutzeinstellung

Das Gerät wird nur zum Lesen konfiguriert. Dementsprechend wird der DIP Schalter 8 auf ON gestellt.

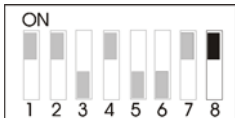


Bild 2

Projektierung in der Steuerung

Hardware

Für die Projektierung wird der Simatic Manager Version 5.5+SP1 eingesetzt. Um das EKS am Profibus zu parametrieren, ziehen Sie das Objekt „EKS-A-IDX-G01-ST09/03“ auf den Profibus und anschließend das Modul „Read/Write: 32 Byte“ in den ersten Slot. Der Adressbereich kann auf 256 bis 287 eingestellt bleiben. Die Größe von 32 Byte wird gewählt, da insgesamt 21 Byte an Nutzdaten inkl. der KeyID abgeholt werden sollen. Hierzu muss in der Steuerung ein ausreichend großer Eingangsbereich gehalten werden.

Wenn ein neuer Schlüssel gesteckt wird, werden die Daten ab Byte 0 immer automatisch gelesen. Da in diesem Beispiel die Nutzdaten nicht am Anfang des Schlüssels sondern am Ende stehen, werden die eigentlichen Nutzdaten in einer Leseroutine im Ablaufprogramm geladen. Dabei wird die KeyID mit abgeholt.

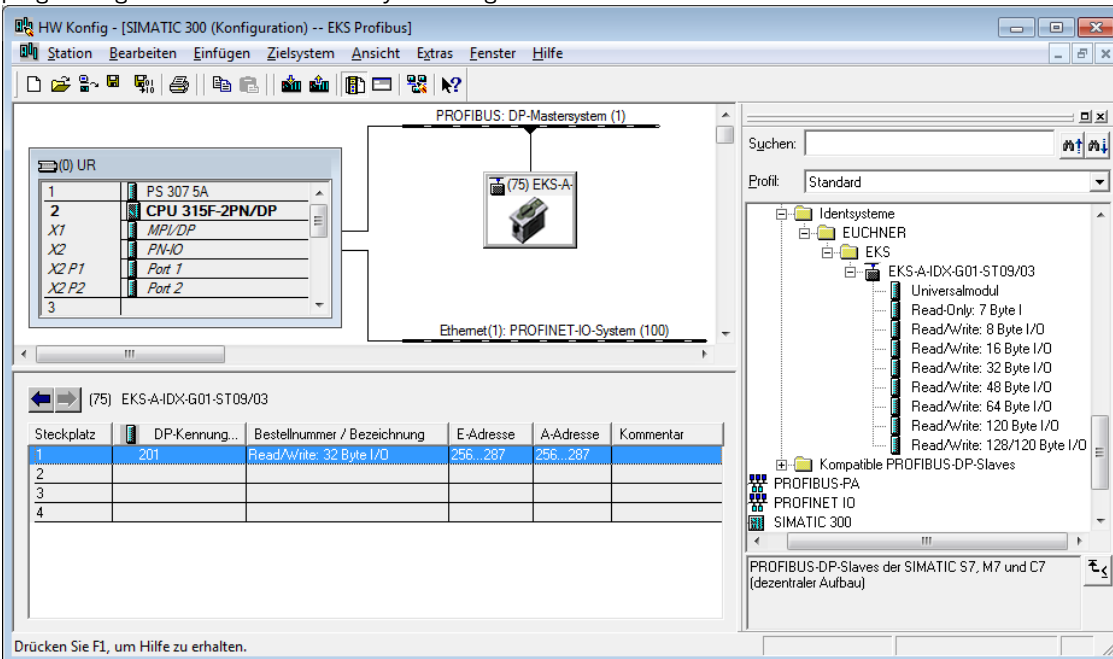


Bild 3

In den Eigenschaften zur Profibus Schnittstelle des DP-Slaves stellen Sie die Adresse 75, passend zur Einstellung an den DIP-Schaltern, ein. Die Subnetz Parameter stellen Sie entsprechend Ihres Bussystems ein.

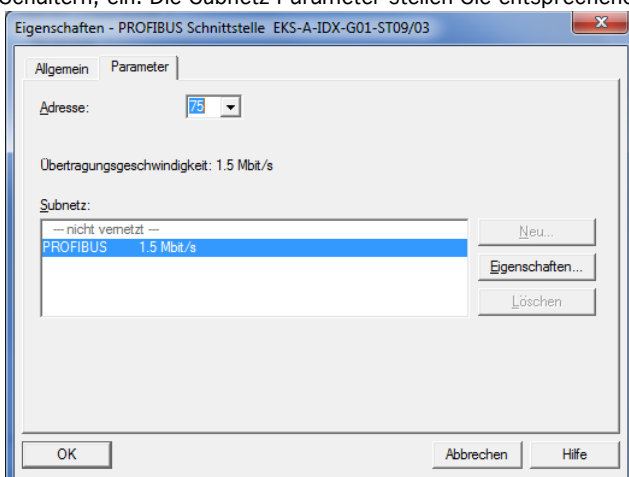


Bild 3

Programmierung in der Steuerung

Globale Datenbausteine

Es werden Datenbausteine angelegt, in denen die Sende- und Empfangsdaten für das EKS abgelegt werden. Im Datenbaustein DB1 für das Lesen sind die Daten strukturiert angelegt, wobei alle Daten mit mehr als einem Byte als Einzelbytes angelegt sind, um das geradzahlige Alignment in der Steuerung zu umgehen. Der Datenbaustein muss dieselbe Länge wie der Eingangsbereich des EKS haben, da andernfalls die Systemfunktion zum Lesen nicht funktioniert.

DB1, ReadBufferEKS

Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+0.0	ReadEKSStatus	BYTE	B#16#0	Statusbyte vom EKS
+1.0	ReadKeyCount	BYTE	B#16#0	Zähler für gesteckte Schlüssel
+2.0	ReadStartAddress	BYTE	B#16#0	Erstes gelesenes Byte
+3.0	ReadNumberBytes	BYTE	B#16#0	Anzahl der gelesenen Bytes
+4.0	ReadCRC_00	BYTE	B#16#0	CRC Byte 0
+5.0	ReadCRC_01	BYTE	B#16#0	CRC Byte 1
+6.0	ReadDate_00	BYTE	B#16#0	Datum Byte 0
+7.0	ReadDate_01	BYTE	B#16#0	Datum Byte 1
+8.0	ReadDate_02	BYTE	B#16#0	Datum Byte 2
+9.0	ReadDate_03	BYTE	B#16#0	Datum Byte 3
+10.0	ReadDate_04	BYTE	B#16#0	Datum Byte 4
+11.0	ReadDate_05	BYTE	B#16#0	Datum Byte 5
+12.0	ReadDate_06	BYTE	B#16#0	Datum Byte 6
+13.0	ReadDate_07	BYTE	B#16#0	Datum Byte 7
+14.0	ReadAuthorization_00	BYTE	B#16#0	Berechtigungsstufe Byte 0
+15.0	ReadAuthorization_01	BYTE	B#16#0	Berechtigungsstufe Byte 1
+16.0	ReadDepartment	BYTE	B#16#0	Abteilungsnummer
+17.0	ReadKeyID_00	BYTE	B#16#0	KeyID Byte 0
+18.0	ReadKeyID_01	BYTE	B#16#0	KeyID Byte 1
+19.0	ReadKeyID_02	BYTE	B#16#0	KeyID Byte 2
+20.0	ReadKeyID_03	BYTE	B#16#0	KeyID Byte 3
+21.0	ReadKeyID_04	BYTE	B#16#0	KeyID Byte 4
+22.0	ReadKeyID_05	BYTE	B#16#0	KeyID Byte 5
+23.0	ReadKeyID_06	BYTE	B#16#0	KeyID Byte 6
+24.0	ReadKeyID_07	BYTE	B#16#0	KeyID Byte 7
+26.0	Buffer	ARRAY[0..5]		Empfangspuffer auf 32 Byte füllen
+1.0		BYTE		
=32.0		END_STRUCT		

Bild 4

Da ein Kommando an das EKS gesendet werden muss, damit die Daten ab Byte Nr. 103 inklusive der KeyID gelesen werden können, stehen am Anfang des Datenbausteins DB2 die hierzu notwendigen Daten. Es wird immer derselbe Datenbereich vom EKS Schlüssel gelesen. In der Initialisierung wird der Datenbaustein passend vorbelegt. Dies sind die Bytes WriteCommand (64(dez)), WriteStartAddress (103(dez)) und WriteNumberBytes (21(dez)). Der Datenbaustein muss dieselbe Länge wie der Ausgangsbereich des EKS haben, da andernfalls die Systemfunktion zum Schreiben nicht funktioniert.

DB2, WriteBufferEKS

Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+0.0	WriteCommand	BYTE	B#16#40	Sendepuffer
+1.0	WriteStartAddress	BYTE	B#16#67	Startadresse auf dem Schlüssel (103)
+2.0	WriteNumberBytes	BYTE	B#16#15	Anzahl Bytes, die vom Schlüssel gelesen werden
+4.0	Buffer	ARRAY[0..27]		Puffer für die restliche Sendung
+1.0		BYTE		
=32.0		END_STRUCT		

Bild 5

DB10, Instanzenbaustein für FB1

Da der Funktionsbaustein FB1 mit statischen Variablen arbeitet, muss ein DB als Instanzenbaustein verwendet werden. Im Beispiel wird dazu DB10 angelegt.

AWL Programm zum Abruf des Schlüsselinhalts

Das Leseprogramm ist in diesem Beispiel im FB1 programmiert. Das Programm liest nur dann, wenn ein Schlüssel gesteckt ist und neue Daten bereit stehen. Ein bereits gelesener Schlüssel wird kein zweites Mal eingelesen. Es werden die Daten ab Byte 103 (KeyCRC) inklusive der KeyID gelesen und im Datenbaustein DB1 ab Byte 4 zur weiteren Bearbeitung zur Verfügung gestellt. Insgesamt werden 21 Bytes Nutzdaten vom EKS Schlüssel abgeholt.

Im den Bytes 0 bis 3 des DB1 werden die Statusbytes vom EKS gespeichert.

Beschreibung der Schnittstelle

Eingabedaten

Keine.

Ausgabedaten

Fehlermeldung, Neuer Schlüssel und Status des DP-Slaves.

Ein- Ausgabedaten

Keine.

Statische Daten

Der Zählerstand des EKS wird statisch angelegt. Dieser Wert wird gegen den Wert, der vom EKS gelesen wird, verglichen. Es werden nur dann Daten abgeholt, wenn die beiden Werte unterschiedlich sind.

Temporäre Daten

Keine

Name	Datentyp	Adresse	Anfangswert	Kommentar
IN		0.0		
OUT		0.0		
Error	Bool	0.0	FALSE	Fehlermeldung
NewKey	Bool	0.1	FALSE	Neuer Schlüssel eingelesen
DPStatus	Word	2.0	W#16#0	Status des DP Slaves
IN_OUT		0.0		
STAT		0.0		
KeyCount	Byte	4.0	B#16#0	Zählerstand des gesteckten EKS Schlüssels
CycleMarker	Int	6.0	0	Ablaufzähler Vorgang lesen
TEMP		0.0		

Bild 6

Veränderte Register

A1, A2, SW

Unveränderte Register

AR1, AR2, DBR1, DBR2

Verwendete Systemfunktionen

SFC14, DPRD_DAT – DP-Normslaves/PROFINET IO-Devices lesen

SFC15, DPWR_DAT – DP-Normslaves/PROFINET IO-Devices schreiben

Globale Daten

Die Datenbausteine DB1 und DB2 werden mit einer Mindestgröße von jeweils 32 Byte vorausgesetzt.

Der Inhalt des Datenbausteins DB1 wird vollständig überschrieben.

Im Datenbaustein DB2 werden die erste 3 Bytes (Kommandobyte, EKS Adresse und Länge der Nutzdaten) überschrieben.

Symboltabelle

	Status	Symbol ▲	Adresse	Datentyp	Kommentar
1		Calculate CRC	FB 2	FB 2	
2		COMPLETE RESTART	OB 100	OB 100	Complete Restart
3		Data FB1	DB 10	FB 1	
4		Data FB2	DB 11	FB 2	
5		DIS_AIRT	SFC 41	SFC 41	Delay the Higher Priority Interrupts and Asynchronous Errors
6		DPNRM_DG	SFC 13	SFC 13	Read Diagnostic Data of a DP Slave
7		DPRD_DAT	SFC 14	SFC 14	Read Consistent Data of a Standard DP Slave
8		DPWR_DAT	SFC 15	SFC 15	Write Consistent Data to a Standard DP Slave
9		EKSactive	E 256.0	BOOL	EKS aktiv
10		EKSIn	E 256.1	BOOL	Kenntung, ob ein Schlüssel gesteckt ist
11		EKSInCount	EB 257	BYTE	Zählerstand des EKS
12		EKSJobactive	E 256.7	BOOL	EKS Auftrag aktiv
13		EKSJobDone	E 256.6	BOOL	EKS Auftrag beendet
14		EN_AIRT	SFC 42	SFC 42	Enable Higher Priority Interrupts and Asynchronous Errors
15		F_CTRL_1	FB 273	FB 273	
16		F_CTRL_2	FB 274	FB 274	F_: Test Block an Programm Run Control
17		F_DIAG_N	FB 275	FB 275	F_: Diagnosticbuffer Message with non CPU-Stop
18		F_GLOBDB	DB 545	DB 545	F_: F_Global_Data Block
19		F_JO_CGP	FB 272	FB 272	F_: Driver Block In-Output with Channel Granular Passivation
20		I/O_FLT1	OB 82	OB 82	I/O Point Fault 1
21		Main Programm	OB 1	OB 1	
22		PROG_ERR	OB 121	OB 121	Programming Error
23		RDSYSST	SFC 51	SFC 51	Read a System Status List or Partial List
24		ReadBufferEKS	DB 1	DB 1	
25		ReadEKS	FB 1	FB 1	
26		STP	SFC 46	SFC 46	Change the CPU to STOP
27		VAT_1	VAT 1		
28		WriteBufferEKS	DB 2	DB 2	

Bild 8

AWL Programm im FB1- ReadEKS

Netzwerk: 1		Prüfen, ob EKS bereit
Prüfen, ob EKS bereit, wenn nicht Bausteinende.		
UN	"EKSactive"	// EKS bereit
BEB		
Netzwerk: 2		Prüfen, ob Schlüssel gesteckt
UN	"EKSIn"	// Prüfen ob ein Schlüssel gesteckt ist
SPB	RES	
Netzwerk: 3		Schlüssel gesteckt
Prüfen, ob es sich um einen neuen Schlüssel handelt.		
L	#KeyCount	// Wenn in KeyZaehler derselbe Wert steht, wurde bereits gelesen
L	"EKSInCount"	// Derzeitiger Zählerstand im EKS
==I		// Vergleiche, nur wenn Zählerstand im EKS höher
SPBN	ZS	// Nur mit neuem Schlüssel auch Daten holen
R	#Error	// Rückmeldung, kein Fehler
R	#NewKey	// Rückmeldung, kein Schlüssel
BE		

Bild 9a

```
Netzwerk: 4      Start Zyklusablauf zum Lesen der Schlüsseldaten
Start der eigentlichen Leseroutine von einem neu gesteckten Schlüssel.
Zyklusablauf.

ZS:  L   #CycleMarker
     SPL  NEXT

     SPA  z1           // 1.Zyklus: Lesekommando setzen
     SPA  z2           // 2.Zyklus: Lesekommando rücksetzen
     SPA  z3           // 3.Zyklus: Daten lesen

NEXT: BEA

Netzwerk: 5      1.Zyklus
Übertragen der Parameter an das EKS, damit der Bereich ab Byte 103
einschließlich der KeyID gesendet wird.
Der Sendebereich ist bereits durch Initialisierung im DB2 vorbelegt und
muss nicht neu beschrieben werden.
Das Kommando muss jeweils gesetzt werden.

z1:  U   "EKSJobactive"           //Warte bis kein Auftrag mehr in Arbeit und
     UN  "EKSJobDone"            //und Auftrag fertig
     BEB

     L   64                       // Setze Kommando Lesen von Schlüsselbyte 103 mit 21 Byte
     T   "WriteBufferEKS".WriteCommand
     L   103
     T   "WriteBufferEKS".WriteStartAddress
     L   21
     T   "WriteBufferEKS".WriteNumberBytes

     L   1
     T   #CycleMarker            // Zyklusmarker für nächsten Zyklus vorbereiten

     CALL "DPWR_DAT"              // Aufruf SFC 15 DPWR_DAT
     LADDR :=W#16#100            // Adresse des EKS Speicherbereichs
     RECORD :=P#DB2.DBX0.0 BYTE 32 // Anfangsadresse des DB, Länge muss 32 sein
     RET_VAL:=MW1                // Rückmeldung

// Prüfe, ob ein Fehler aufgetreten ist
     L   MW      1                // Rückmeldung von Profibus schreiben
     L   0
     ==I
     SPBN  MERR                    // wenn ein Wert <> 0 zurückgegeben wurde: Fehler

     BEA
```

Bild 9b

```
Netzwerk: 6      2.Zyklus
Rücksetzen des Lesekommandos

z2:  UN  "EKSJobactive"           // Warten das Auftrag aktiv wird und
     U   "EKSJobDone"            // Auftrag nicht beendet, dann Rücksetzen des Lesekommandos
     BEB

     L   0                       // Rücksetzen des Lesekommandos
     T   "WriteBufferEKS".WriteCommand

     L   2
     T   #CycleMarker            // Zyklusmarker für nächsten Zyklus vorbereiten

     CALL "DPWR_DAT"              // Aufruf SFC 15 DPWR_DAT
     LADDR :=W#16#100            // Adresse des EKS Speicherbereichs
     RECORD :=P#DB2.DBX0.0 BYTE 32 // Anfangsadresse des DB, der gesendet werden soll, Länge muss 32 s
     ein
     RET_VAL:=MW1                // Rückmeldung

//Prüfe, ob Fehler aufgetreten ist
     L   MW      1                // Rückmeldung von Profibus schreiben
     L   0
     ==I
     SPBN  MERR                    // wenn ein Wert <> 0 zurückgegeben wurde: Fehler

     BEA
```

Bild 9c


```

Netzwerk: 7      3.Zyklus
Lesen der Schlüsseldaten.

z3:  U   "EKSJobactive"      // Warte bis kein Auftrag mehr in Arbeit und
     UN  "EKSJobDone"        // und Auftrag fertig
     BEB

     CALL "DPRD_DAT"          // Aufruf SFC 14 DPRD_DAT
     LADDR :=W#16#100        // Adresse des EKS Speicherbereichs
     RET_VAL:=MW1             // Rückmeldung
     RECORD :=P#DB1.DBX0.0 BYTE 32 // Anfangsadresse des DB,in dem empfangen werden soll, Länge muss 32 sei
                                     n

     L    0
     T    #CycleMarker

//Prüfe, ob Fehler aufgetreten ist

     L    MW    1             // Rückmeldung von Profibus lesen
     L    0             // Nur Rückgabewert 0 ist ok
     ==I
     SPBN MERR             // wenn ein Wert <> 0 zurückgegeben wurde: Fehler

//Schlüssel vollständig gelesen, die Daten stehen jetzt im DB1
     L    "EKSInCount"       // Derzeitigen Zählerstand aus EKS auslesen
     T    #KeyCount          // Merken, dass mit diesem Zählerstand vollständig gelesen wurde
     S    #NewKey            // Zurück melden, dass ein neuer Schlüssel komplett gelesen wurde
     R    #Error             // kein Fehler
     BEA

Netzwerk: 8      Fehlerbearbeitung

MERR: L    MW    1
      T    #DPStatus        // DP Status als Rückmeldung im Fehlerfall
      S    #Error           // Rückwert = 1, Fehler aufgetreten
      R    #NewKey          // Rückmeldung, kein Schlüssel
      L    0
      T    #CycleMarker     // Ablaufzyklus zurücksetzen
      BE

Netzwerk: 9      Reset

RES:  R    #Error           // kein Fehler
      R    #NewKey          // Rückmeldung, kein Schlüssel
      L    0
      T    #CycleMarker     // Ablaufzyklus zurücksetzen
  
```

Bild 9d

Aufruf von FB1

```

//Abholen der Daten vom EKS Schlüssel
CALL "Read EKS" , "Data FB1"
Error   :=M0.0             // Rückgabewert für Fehler
NewKey  :=M0.1             // Rückgabewert, ob neuer Schlüssel
DPStatus:=#Status         // Status des DP-Slaves

U    M    0.0             // Prüfe ob Fehler aufgetreten ist
SPB  MERR             // Wenn Werte = 1 springe zur Fehleroutine
  
```

Bild 10

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 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 Risikoanalyse 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.