EUCHNER
More than safety.

EKS *FSA* an Siemens S7-1200 – An-/Abmeldesystem in PL d

Inhalt

1 Verwendete Bauteile / Module

1.1 EUCHNER

Beschreibung	Best.-Nr. / Artikelbezeichnung
EKS Profinet *FSA*	106306 / EKS-A-IIXA-G01-ST02/03/04 122353 / EKS-A-AIXA-G18
EKS Schlüssel	077859 / EKS-A-K1RDWT32-EU 084735 / EKS-A-K1BKWT32-EU 091045 / EKS-A-K1BLWT32-EU 094839 / EKS-A-K1GNWT32-EU 094840 / EKS-A-K1YEWT32-EU 123097 / EKS-A-K1WHWT32-EU 123098 / EKS-A-K1OGWT32-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.

1.2 Andere

Beschreibung	Artikel
S7-1200, CPU 1215FC DC/DC/DC	6ES7 215-1AF40-0XB0
Digitaleingabe SM 1226, F-DI8/16 x DC24V	6ES7 226-6BA32-0XB0
Digitalausgabe SM 1226, F-DQ4 x DC24V	6ES7 226-6DA32-0XB0

1.3 Software Euchner

Beschreibung	Artikel
Electronic Key Manager EKM	093322, ANWPG ELECTRONIC KEY MANAGER 098578, ANWPG ELECTRONIC KEY MANAGER EINZEL
KeyProtect	Über EUCHNER Support anfordern
TIA Library	AP000247 EKS FSA Logon Logoff TIA Vxx.zalxx

1.4 Software Andere

Beschreibung	Artikel
Totally Integrated Automation Portal	Version V14 SP1 Update 9
STEP 7 Professional	Version V14 SP1 Update 9
STEP 7 Safety	Version V14 SP1 Update 9

2 Funktionsbeschreibung

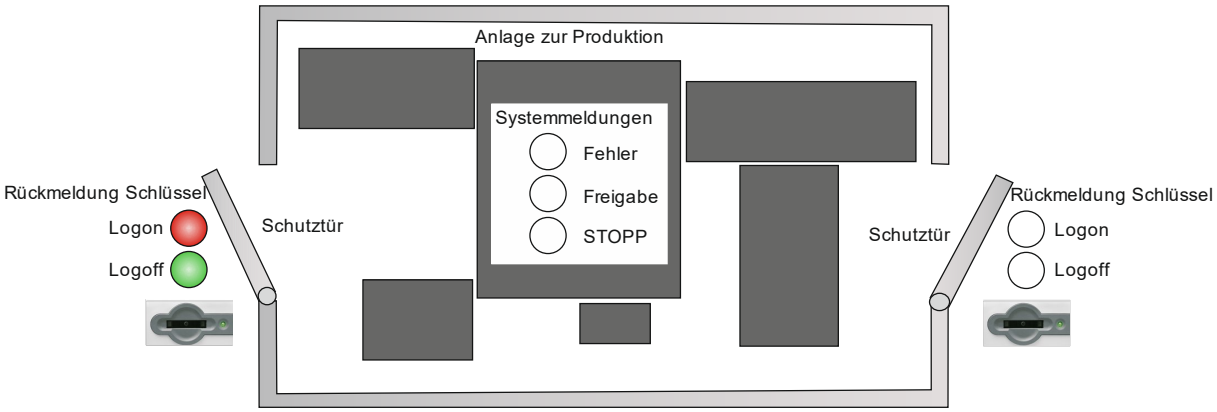
2.1 Allgemein

Das System besteht aus mehreren Zugängen, an denen EKS Lesestationen zum sicheren Anmelden und Abmelden bereitstehen. Sobald mindestens eine Person angemeldet ist, wird ein Signal gesetzt, mit dem verhindert werden kann, dass von der Maschine eine Gefährdung ausgeht.

Jeder Mitarbeiter, der einen Zutritt in eine Anlage haben soll, erhält einen EKS-Schlüssel, der im System speicherbar ist. Vor dem Betreten der Anlage muss der Schlüssel vom Mitarbeiter in eine beliebige EKS Schlüsselaufnahme eingesteckt werden. Der EKS-Schlüssel wird ausgewertet und gespeichert. Sobald mindestens ein Schlüssel gespeichert ist, wird ein sicherer Merker im System zurückgesetzt. Dieser Merker kann zusammen mit anderen Sicherheitssignalen, wie z.B. dem Überwachungssignal einer Zuhaltung verwendet werden, um die Anlage in einem sicheren Stopp zu halten. Zudem kann das Signal als Anforderung zum Stopp der Maschine oder auch zur Freischaltung von Tasten verwendet werden.

Die Abmeldung beim Verlassen der Anlage erfolgt durch erneutes Stecken des persönlichen Schlüssels in eine EKS Schlüsselaufnahme. Sobald alle Schlüssel abgemeldet sind, wird der Merker gesetzt, der anzeigt, dass nun die Anlage wieder einschalten kann.

Zur Rückmeldung an den Bediener werden für jede EKS Schlüsselaufnahme Signale erzeugt, die sicher anzeigen, wie der EKS-Schlüssel ausgewertet wurde, der gerade in der EKS Schlüsselaufnahme steckt.



Schlüsselverwaltung

Jeder Mitarbeiter erhält einen eigenen EKS-Schlüssel, der eine einmalig vergebene Nummer enthält. Die Nummer kann personalisiert werden,. Es gibt verschiedene Arten von EKS-Schlüsseln (s.u.) für das An- und Abmeldesystem.
Die Schlüsselverwaltung erfolgt mit der auf Windows basierenden Software EKM von EUCHNER. Alle an Mitarbeiter ausgegebenen EKS-Schlüssel werden in einer Datenbank gespeichert. Jeder EKS-Schlüssel muss nach dem Beschreiben mit EKM mit der Software KeyProtect von EUCHNER freigegeben werden. EKS-Schlüssel werden mittels EKM gegen einfaches Kopieren geschützt.

Nutzung durch die Mitarbeiter

Ein Mitarbeiter, der in einer gefährlichen Anlage arbeiten muss, steckt vor dem Betreten der Anlage seinen EKS-Schlüssel in eine der vorhandenen EKS Schlüsselaufnahmen. Er bekommt eine Rückmeldung, dass die Anmeldung erfolgreich war. Die Rückmeldung erfolgt nicht durch das EKS selbst, hierzu muss eine geeignete Meldefunktion (z.B. Anzeigeleuchten) vorgesehen werden.
Nach Beenden der Arbeiten steckt der Mitarbeiter seinen EKS-Schlüssel erneut in eine der EKS Lesestationen. Er bekommt die Rückmeldung, dass die Abmeldung erfolgreich war. Auch diese Rückmeldung erfolgt nicht durch das EKS.
Hinweis: Zum Abmelden muss nicht dieselbe EKS-Schlüsselaufnahme wie zur Anmeldung verwendet werden.

Arten von EKS-Schlüsseln

Es gibt unterschiedliche EKS-Schlüssel, die von der sicheren Steuerung ausgewertet werden.

1. Gültiger Schlüssel: Mitarbeiterschlüssel
   Schlüssel, der an Mitarbeiter ausgegeben wird und der für diese Applikation freigegeben ist. Mit diesem Schlüssel können sowohl An-, als auch Abmeldungen gemacht werden.
   Anmerkung: Der Schlüssel kann auch zur Steuerung einer Zuhaltung genutzt werden, um auf diese Weise nur berechtigten Personen den Zutritt zur Anlage zu gewähren.
2. Gültiger Schlüssel: Sonderschlüssel
   Schlüssel, der an gesondert geschulte Mitarbeiter ausgegeben wird und der zudem für diese Applikation freigegeben ist. Mit diesem Schlüssel können sowohl An-, als auch Abmeldungen gemacht werden.
   Darüber hinaus ist der Inhaber dieses Schlüssels berechtigt, das System im Falle eines Fehlers zurückzusetzen und somit die Anlage im Fehlerfall wieder in Betrieb zu nehmen.
   Anmerkung: Der Schlüssel kann auch zur Steuerung einer Zuhaltung genutzt werden, um auf diese Weise nur berechtigten Personen den Zutritt zur Anlage zu gewähren.
3. Ungültiger Schlüssel: Kopierter Schlüssel
   Elektronische Schlüssel können kopiert werden. Die Steuerung erkennt einen solchen Kopierversuch. In der Applikation kann ein Mitarbeiter sich mit einem dieser Schlüssel aus Sicherheitsgründen zwar anmelden, sich aber nicht mehr abmelden. Um das System zurückzusetzen, muss ein Sonderschlüssel verwendet werden.
   Anmerkung: Realisierbar ist auch, dass eine Zuhaltung mit diesem Schlüssel nicht geöffnet werden kann und der Zugang verweigert wird.
4. Ungültiger Schlüssel: Defekter Schlüssel
   Schlüssel, auf dem ein Fehler vorliegt, z.B. ein Defekt im Speicher des Schlüssels. Derselbe Fehler kann auch durch

eine fehlerhafte Datenübertragung entstehen. Hierbei wird dann ein gültiger Schlüssel gesteckt, jedoch werden dessen Daten falsch an die sichere Steuerung übertragen.
Die sichere Steuerung erkennt diese Fehler.
Der Mitarbeiter wird mit einem dieser Schlüssel zwar angemeldet, kann sich aber nicht mehr abmelden. Um das System zurückzusetzen, muss ein Sonderschlüssel verwendet werden.
Anmerkung: Realisierbar ist auch, dass eine Zuhaltung mit diesem Schlüssel nicht geöffnet werden kann und der Zugang verweigert wird.

5. Ungültiger Schlüssel: Fremder Schlüssel
   Schlüssel der völlig falsche Daten enthält und nicht zu der Anlage gehört. In der Applikation kann sich ein Mitarbeiter mit einem solchen Schlüssel weder an- noch abmelden.
   Falls auch dieser Mitarbeiter geschützt werden soll, muss die Anmeldung erlaubt werden, jedoch die Abmeldung unterbunden.
   Anmerkung: Realisierbar ist auch, dass eine Zuhaltung mit diesem Schlüssel nicht geöffnet werden kann und der Zugang verweigert wird.

Rückmeldungen der Steuerung an den Nutzer

Die Steuerung gibt sichere Meldungen bezogen auf jedes einzelne EKS aus, wenn ein EKS-Schlüssel gesteckt wird.

1. Anmeldung akzeptiert
   Die Anmeldedaten wurden gespeichert.
2. Abmeldung akzeptiert
   Passende Anmeldedaten wurden gefunden und gelöscht. Damit ist der Mitarbeiter abgemeldet.
3. Abmeldung nicht akzeptiert
   Der Schlüssel wurde zuvor angemeldet. Es wird kein Eintrag gelöscht, eine Abmeldung ist nicht möglich. Um das System zurücksetzen zu können, wird ein Sonderschlüssel benötigt.
4. Anmeldung nicht akzeptiert
   Die Anmeldedaten können nicht gespeichert werden. Diese Meldung wird nur generiert, wenn Schlüssel abgewiesen werden sollen.

Meldungen der Steuerung zur Anlage

1. Mindestens ein Mitarbeiter ist angemeldet
   Die Anlage darf nicht starten. Diese Meldung ist ein sicheres Signal.
2. Anlage darf starten
   Es sind keine Mitarbeiter im System angemeldet. Diese Meldung ist ein sicheres Signal.
3. Fehler bezogen auf Schlüssel
   Es gibt Fehlermeldungen, die sich auf den Schlüssel beziehen, der gerade gesteckt ist. Die Fehlermeldung verschwindet, wenn der Schlüssel gezogen wird. Die Auswirkung des Fehlers bleibt erhalten, sofern der Schüssel angemeldet wurde. In diesem Fall kann der Fehler nur durch Verwendung eines Sonderschlüssels gelöscht werden. Aus- und Wiedereinschalten der Steuerung löscht den Fehler nicht.
   3.1. Fremder Schlüssel
   3.2. Anmeldung enthält fehlerhafte Daten (Schlüssel oder Datenübertragung)
   Die Anmeldung ist erfolgt, jedoch aufgrund fehlerhafter Daten. Eine Abmeldung wird nicht möglich sein.
   3.2.1. Fehler in Schlüssel: kopierter Schlüssel
   3.2.2. Fehler in Schlüssel: defekter Schlüssel
4. Systemfehlermeldung
   Wenn eine Systemfehlermeldung vorliegt, kann die Anlage nicht gestartet werden. Systemfehlermeldungen können durch Ausschalten und erneutes Einschalten der Steuerung oder durch Einsatz eines Sonderschlüssels gelöscht werden.
   4.1. Zu viele Anmeldungen
   Es sollen mehr als die maximale Anzahl möglicher Schlüssel gespeichert werden.
   4.2. Fehler in Hardware erkannt (pro EKS)
   Eine Unterbrechung, ein Kurzschluss oder ein Fehler in der Hardware liegt vor.

Rücksetzen des Systems

Ein Löschen von Anmeldedaten durch Aus- und erneutes Wiedereinschalten der Steuerung ist nicht möglich. Löschen von Anmeldedaten ist nur durch den Einsatz eines Sonderschlüssels möglich.

2.2 Einsatz EKM und KeyProtect

Für diese Applikation ist der Einsatz der beiden Softwarepakete EKM und KeyProtect von Euchner unerlässlich. Mit EKM werden alle relevanten Schlüsseldaten geschrieben. Die Struktur der Schlüsseldaten, wie sie in diesem Beispiel genutzt wurde, finden Sie im Abschnitt „Struktur des Schlüssels“. Im Datenwort EKS_UniqueCode muss in jedem Schlüssel, der beschrieben wird, ein anderer Wert stehen.
Die Software KeyProtect prüft, ob der Wert in EKS_UniqueCode nur einmalig vergeben wurde. Es empfiehlt sich aus Gründen der Datensicherung, die beiden Programme auf unterschiedlichen Computern einzusetzen.

WICHTIG: In jedem Fall muss eine regelmäßige Datensicherung der Computer dafür sorgen, dass die gespeicherten Daten der Schlüssel nicht verloren gehen können. Andernfalls wäre es möglich, das zwei verschiedene Schlüssel mit unterschiedlichen Kennungen ausgegeben werden. Das könnte zu der gefährlichen Situation führen, dass die erste Person sich mit einem solchen Schlüssel anmeldet, jedoch eine zweite Person die Möglichkeit hat, diesen Schlüssel abzumelden.

KeyProtect muss über die Konfigurationsdatei „KeyProtect.exe.config“ angepasst werden. Folgende Datei wird bei KeyProtect mitgeliefert bzw. automatisch beim ersten Starten von KeyProtect erzeugt:

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
        <startup>
             <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.6" />
        </startup>
        <appSettings>
             <add key="KeyCodeKeyAddress" value="106"></add>
             <add key="CRCKeyAddress" value="108"></add>
             <add key="ComPort" value="COM1"></add>
        </appSettings>
</configuration>
```

Ändern Sie die Werte für „KeyCodeKeyAddress“ und für „CRCKeyAddress“ bei Bedarf passend zu Ihrer gewählten Schlüsselstruktur ab. Die COM-Schnittstelle in der Variablen „ComPort“ können Sie im Programm selbst anpassen, dort werden auch die vorhandenen Schnittstellen angezeigt.

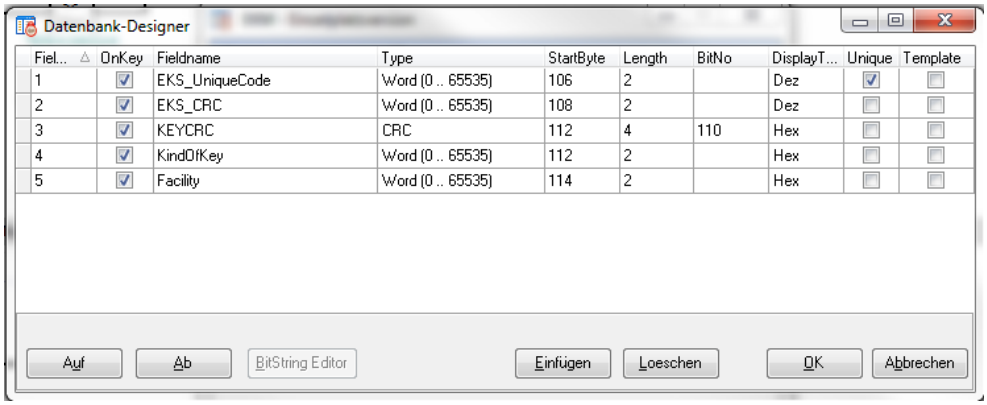
Um nach dem Beschreiben eines Schlüssels aus EKM heraus, den Schlüsselinhalt zu sichern, muss die Software KeyProtect eingesetzt werden. Diese schreibt bei gestecktem EKS-Schlüssel einen Prüfwert an die Stelle CRCKeyAddress nach Klicken auf den Button „Write CRC to EKS“. Dadurch wird sichergestellt, dass der Code an der Stelle KeyCodeKeyAddress einmalig ist. Im Fehlerfall wird eine Meldung angezeigt.

2.3 Struktur des Schlüssels

Die EKS-Schlüssel werden mit der Schlüsselverwaltungssoftware EKM verwaltet. Hierzu wird in dieser Applikation folgende Schlüsselstruktur verwendet.

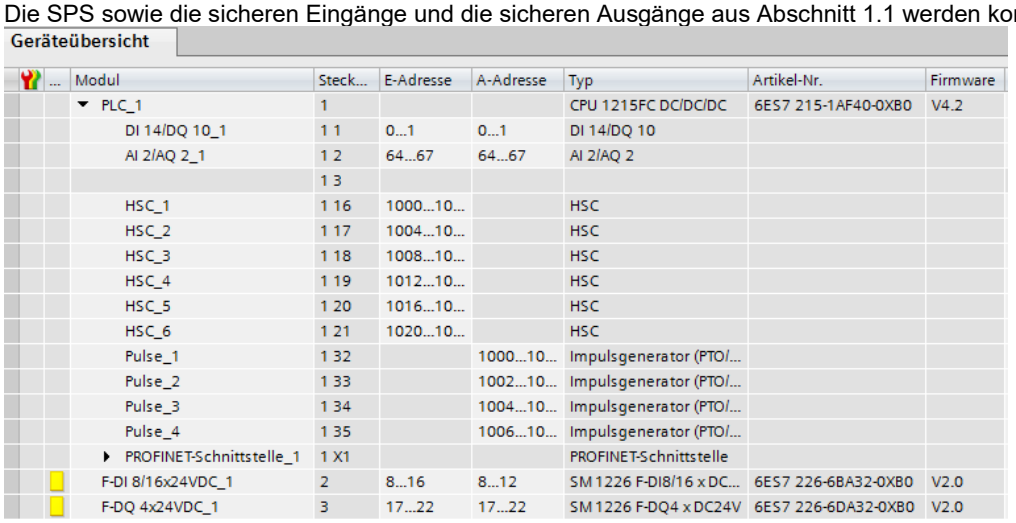
106	Word	EKS_UniqueCode: Einmalige Nummer von EKM vergeben. Diese Adresse heißt in KeyProtect KeyCodeAddress und die Nummer wird mit KeyProtect auf Einmaligkeit geprüft. **WICHTIG:** Hierzu muss in EKM in der Spalte Unique der Haken gesetzt sein!
108	Word	EKS_CRC: Wird nicht mit EKM sondern mit KeyProtect beschrieben. In KeyProtect heißt diese Stelle CRCKeyAddress.
110	KEYCRC	Prüfsumme aus EKM. In diesem Beispiel ab der Adresse 112 mit der Länge 4. Tipp: Die FCS kann beliebig berechnet werden, darf jedoch nicht das Wort EKS_CRC umfassen.
112	Word	KindOfKey: Wert aus EKM zur Kennzeichnung eines Schlüssels als Reset-Schlüssel mit Rücksetzberechtigung. Sollte mittels TSTRING in EKM beschrieben werden.
114	Word	Facility: Wert aus EKM, der für jede Firma immer identisch sein muss. Es können weitere Werte verwendet werden, wenn z.B. ein Schlüssel nur innerhalb einer Abteilung verwendet werden darf. Dazu müsste die Struktur erweitert werden.
116	8 Byte	Eindeutige Schlüsselkennung (wird von EUCHNER so ausgeliefert)

Tipp: Sie können jede beliebige andere Struktur verwenden. Für die Applikation notwendig sind die beiden Worte, die in diesem Beispiel auf Adresse 106 und 108 stehen.

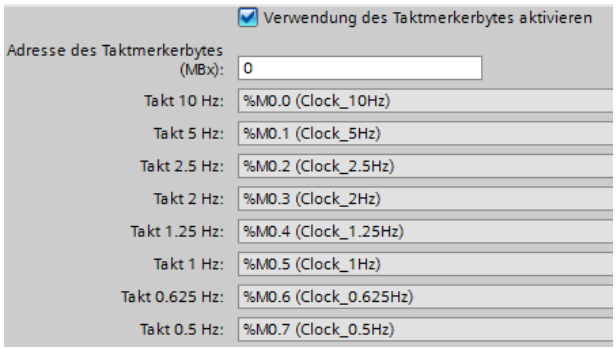


2.4 Gerätekonfiguration in der SPS

Die SPS sowie die sicheren Eingänge und die sicheren Ausgänge aus Abschnitt 1.1 werden konfiguriert.



In der SPS werden die Taktmerkerbytes aktiviert und auf das Merkerbyte 0 gelegt. Diese Merker werden ausschließlich zur Signalisierung der Rückmeldungen zum Benutzer in der SPS verwendet. Sie können jede beliebige andere Signalisierung verwenden.



Pro verwendetem EKS muss ein einkanaliger sicherer Eingang für den FSA-Ausgang angeschlossen werden, der parametriert werden muss. In diesem Beispiel sind die sicheren Eingänge auf Adresse 8.0 und 9.0 verwendet. Folgende Parameter müssen eingestellt werden:

DI-Parameter	Wert
Kurzschlusstest	Aktiviert

Kanal - Parameter	Wert
Auswertung der Geber	1oo1 (1v1) Auswertung

Kanal 0 - Parameter	Wert
Geberversorgung	Intern

Kanal 8 - Parameter	Wert
Geberversorgung	Intern

Pro verwendetem EKS muss ein sicherer Ausgang zur Signalisierung verwendet werden, der parametriert werden muss. In diesem Beispiel sind die sicheren Ausgänge auf Adresse 17.0 bis 17.3 verwendet.

Die EKS werden alle mit dem Kopfmodul „EKS-A-IIXA-G01-ST02/03/04“ eingerichtet. In diesem Beispiel werden die Module „Lesen:033 Bytes“ und „Schreiben: 012 Bytes“ in die Slots eingesteckt.

Geräteübersicht

	Modul	Baugr...	Steck...	E-Adresse	A-Adres...	Typ
	EKS-PN_1	0	0			EKS-A-IIXA-G01-ST...
	Interface	0	0 X1			EKS-PN
	Lesen: 033 Bytes_1	0	1	68...100		Lesen: 033 Bytes
	Schreiben: 012 Bytes_1	0	2		68...79	Schreiben: 012 Bytes

Hinweis: Verwenden Sie die Module und Submodule entsprechend der Menge der von Ihnen benötigten Schlüsseldaten.

2.5 Definition von Konstanten, Datentypen und Speicherbereichen

2.5.1 Anwenderdatentypen

Datentyp **EKS_Key_Read_Type**

Für jedes verwendete EKS müssen Informationen zum aktuellen Status und zum Schlüsselinhalt angelegt werden. In diesem Datentyp ist auch die Struktur des Schlüssels abgebildet. Die Struktur muss den Definitionen aus Abschnitt 2.2 entsprechen. Dies betrifft alle Felder, die in der Spalte Schlüssel mit X gekennzeichnet sind.

Name	Datentyp	Auf Schlüssel	Beschreibung
W_Key_Status	Word	Nein, wird vom EKS gesendet	Der aktuelle Status vom EKS wird im vorderen Byte dieses Worts abgelegt. Das erste Byte auf dem Schlüssel folgt unmittelbar. Da anschließend keine Bytes folgen, sondern DInt und Worte, bleibt in diesem Beispiel das erste Byte vom Schlüssel ungenutzt, um in der SPS die nachfolgenden Daten auf geradzahlige Adressen zu legen.
I_EKS_UniqueCode	Int	Ja	Code geschrieben von der Schlüsselverwaltungssoftware EKM Dieser Code muss einmalig sein.
I_EKS_KeyProtect	Int	Ja	Zweiter Code, geschrieben mit der Software KeyProtect, die sicherheitstechnische Merkmale zum Code I_EKS_UniqueCode zufügt.
I_EKM_FCS	Word	Ja	Prüfsumme geschrieben von der Schlüsselverwaltungssoftware EKM. Diese Prüfsumme dient zur Sicherung gegen einfaches Kopieren eines Schlüsselinhalts. Die Regeln zur Bildung der Prüfsumme werden in EKM festgelegt. Diese Regeln müssen im FC FC1_Check_FCS identisch abgebildet werden. Nähere Informationen hierzu gibt das Handbuch Zusatzinformationen zum EKM.
I_EKS_KindOfKey	Word	Ja	Hier wird ein beliebiger Code hinterlegt, der einen Schlüssel mit der besonderen Berechtigung zum Rücksetzen der Anlage berechtigt.

			In dieser Applikation wird als Code der Wert 32693 (0x7FB5) verwendet. Wählen Sie einen beliebigen Wert aus. Der Wert ist in der Anwenderkonstante C_WORD_SPECIAL_KEY hinterlegt.
I_EKS_Facility	Word	Ja	In diesem Feld ist die Zuordnung zu einer Firma festgelegt. Der zugehörige Wert ist in der Anwenderkonstante C_VALID_FACILITY hinterlegt. In dieser Applikation wird der Wert 1 verwendet. Nehmen Sie einen beliebigen anderen Wert.
B_Key_SerialNumber	Array[0..7] of Byte	Ja	Diese Seriennummer wird fest auf jedem Schlüssel bereits von EUCHNER einprogrammiert und kann nicht verändert werden. Sie dient zum Schutz des Schlüssels gegen kopieren und vor allem als eindeutige Kennung z.B. für Datenbankzugriffe.
M_Read_Error	Bool	Nein	Das Bit wird gesetzt, wenn das Lesen des Schlüssels fehlschlägt. Es ist derzeit keine Reaktion programmiert. Beim Ziehen des Schlüssels wird das Bit zurückgesetzt.
L_EKS_Complete_Code	DInt	Nein	Diese Variable wird gebildet aus den Daten I_EKS_UniqueCode und I_EKS_KeyPoretct, um im Programm nur noch mit dieser einen Variablen arbeiten zu können.
M_FCS_Error	Bool	Nein	Das Bit wird gesetzt, wenn die Prüfsumme auf dem Schlüssel nicht mit den Regeln, die im EKM hinterlegt sind, übereinstimmt. Das deutet z.B. auf einen kopierten Schlüssel hin, sofern alle anderen Daten gültige Werte aufweisen. Beim Ziehen des Schlüssels wird das Bit zurückgesetzt.
M_Wrong_Key	Bool	Nein	Das Bit wird gesetzt, wenn ein Schlüssel erkannt wird, der keine Zugangsberechtigung zur Anlage hat. In dieser Applikation werden diese Schlüssel nicht gespeichert.

Datentyp **LOG_Data_Type**

Um die Daten aus den Schlüsseln in der sicheren SPS speichern zu können, wird ein Datentyp mit der maximal zu speichernden Menge an Schlüsseln angelegt. In diesem Beispiel können gleichzeitig 10 verschiedene Schüssel, entsprechend 10 verschiedenen gleichzeitig angemeldeten Mitarbeitern, gespeichert werden. Wenn sich mehr als 10 Personen anmelden, wird ein Fehlerbit gesetzt und ein Anfahren der Anlage wird verhindert. Der Datentyp wird verwendet, da in der sicheren SPS kein Array genutzt werden kann.

Bezeichnung	**Datentyp**	**Beschreibung**
Log_data_0	DInt	Speicherplatz für L_EKS_CompleteCode
Log_data_1	DInt	Speicherplatz für L_EKS_CompleteCode
Log_data_2	DInt	Speicherplatz für L_EKS_CompleteCode
Log_data_3	DInt	Speicherplatz für L_EKS_CompleteCode
Log_data_4	DInt	Speicherplatz für L_EKS_CompleteCode
Log_data_5	DInt	Speicherplatz für L_EKS_CompleteCode
Log_data_6	DInt	Speicherplatz für L_EKS_CompleteCode
Log_data_7	DInt	Speicherplatz für L_EKS_CompleteCode
Log_data_8	DInt	Speicherplatz für L_EKS_CompleteCode
Log_data_9	DInt	Speicherplatz für L_EKS_CompleteCode

Hinweis: Dieser Speicherbereich muss erweitert werden, wenn mehr als 10 Schlüssel gleichzeitig geloggt sein sollen.

Datentyp **Safe_Data_EKS_Type**

Es werden pro EKS sichere Daten gespeichert, die über diesen Datentyp definiert werden.

Bezeichnung	**Datentyp**	**Beschreibung**
M_KeyProtect_Error_EKS_1	Bool	Flag, das anzeigt, ob die Daten auf dem EKS Schlüssel im EKS 1 korrekt sind. Das Bit wird nicht gesetzt, wenn die Daten in I_EKS_KeyProtect nicht korrekt sind. Das kann verschiedene Ursachen haben: • Der Schlüssel ist defekt

		• Die Software KeyProtect wurde nicht verwendet, um den Schlüssel zu sichern • Die Datenübertragung ist fehlerhaft • In der SPS ist ein Fehler aufgetreten • Im EKS ist ein Fehler aufgetreten Beim Ziehen des Schlüssels wird das Bit zurückgesetzt.
M_KeyProtect_Error_EKS_2	Bool	Flag, das anzeigt, ob die Daten auf dem EKS Schlüssel im EKS 2 korrekt sind. Das Bit wird nicht gesetzt, wenn die Daten in I_EKS_KeyProtect nicht korrekt sind. Das kann verschiedene Ursachen haben: • Der Schlüssel ist defekt • Die Software KeyProtect wurde nicht verwendet, um den Schlüssel zu sichern • Die Datenübertragung ist fehlerhaft • In der SPS ist ein Fehler aufgetreten • Im EKS ist ein Fehler aufgetreten Beim Ziehen des Schlüssels wird das Bit zurückgesetzt.
M_HW_Error_EKS_1	Bool	Flag, das anzeigt, ob eine Diskrepanz zwischen den Daten und dem FSA Ausgang von EKS 1 aufgetreten ist.
M_HW_Error_EKS_2	Bool	Flag, das anzeigt, ob eine Diskrepanz zwischen den Daten und dem FSA Ausgang von EKS 2 aufgetreten ist.

Hinweis: Wenn mehr als 2 EKS verwendet werden sollen, muss dieser Datentyp angepasst werden.

2.5.2 Globale Anwenderkonstanten

Die Anwenderkonstanten sind in der Variablentabelle EKS_Logon_Logoff_Variables definiert.
In den Anwenderkonstanten werden Datenspeichergrößen und grundsätzliche Werte festgelegt.

Bezeichnung	Datentyp	Wert	Beschreibung
C_NO_OF_EKS	Int	2	Anzahl der verwendeten EKS Schlüsselaufnahmen. Die Bausteine sind für maximal 2 mögliche EKS programmiert. Falls mehr als 2 Geräte zum Einsatz kommen sollen, müssen die Bausteine angepasst werden.
C_LEN_EKS_KEY_CONTENT	Int	20	Länge des Speicherbereichs, in dem die Daten abgelegt werden, die vom EKS eingelesen werden. Die Länge umfasst neben den Daten auf dem EKS-Schlüssel auch das Statusbyte, 1 folgendes Leerbyte sowie die Seriennummer (8 Byte). Die Konstante entspricht der Länge der Datenstruktur im Datentyp EKS_Key_Read_Type von W_Key_Status bis B_Key_Serialnumber.
C_LEN_EKS_KEY_BUFFER	Int	26	Länge des gesamten Datentyps EKS_Key_Read_Type inklusive der zusätzlichen Informationen.
C_VALID_FACILITY	Word	16#0100	Dieser Wert muss im Feld Facility (Adresse 114) auf dem Schlüssel stehen. Nur Schlüssel mit diesem Wert werden als zur Anlage zugehörig betrachtet. Der Wert wird im Little Endian Format von EKM auf dem Schlüssel gespeichert (LSB vor MSB). Im EKM wird somit der Wert 1 angezeigt. Wählen Sie einen beliebigen Wert aus.
C_WORD_SPECIAL_KEY	Word	16#B57F	Willkürlicher Wert, der vom EKM im Feld KindOfKey auf den Schlüssel geschrieben wird, um den Schlüssel als Sonderschlüssel mit Berechtigung zum Reset der Anlage zu kennzeichnen. Der Wert wird im Little Endian Format von EKM auf dem Schlüssel gespeichert (LSB vor MSB). Wählen Sie einen beliebigen Wert aus.

2.5.3 Variablen

Die Variablen sind in der Variablentabelle EKS_Logon_Logoff_Variables definiert.

Name	Datentyp	Adresse	Bemerkung
I_EKS_1_Input	Bool	%E8.0 (safe)	An diesem Eingang ist der FSA Ausgang des ersten EKS angeschlossen.
I_EKS_2_Input	Bool	%E9.0 (safe)	An diesem Eingang ist der FSA Ausgang des zweiten EKS angeschlossen.
O_EKS0_Store	Bool	%A17.0 (safe)	An diesem Ausgang wird signalisiert, dass der gelesene Schlüssel am ersten EKS eingespeichert wurde und der Nutzer nun geschützt ist.
O_EKS0_Delete	Bool	%A17.1 (safe)	An diesem Ausgang wird signalisiert, dass der gelesene Schlüssel am ersten EKS gelöscht wurde und der Nutzer nun nicht mehr geschützt ist.
O_EKS1_Store	Bool	%A17.2 (safe)	An diesem Ausgang wird signalisiert, dass der gelesene Schlüssel am zweiten EKS eingespeichert wurde und der Nutzer nun geschützt ist.
O_EKS1_Delete	Bool	%A17.3 (safe)	An diesem Ausgang wird signalisiert, dass der gelesene Schlüssel am zweiten EKS gelöscht wurde und der Nutzer nun nicht mehr geschützt ist.
M_Auxiliary	Bool	%M2.0	Hilfsflag ohne spezielle Bedeutung. Das Flag wird nur lesend verwendet, der Zustand spielt keine Rolle.
M_Startup	Bool	%M2.1	Das Flag wird beim Kaltstart der SPS im OB100 rückgesetzt. Nachdem im sicheren Baustein FB1 die Daten erfolgreich aus dem Remanenzspeicher der SPS gelesen wurden, wird das Flag gesetzt. Um Fehler zu minimieren, darf dieses Flag nicht in der HMI sichtbar sein.
M_Startup_Finished	Bool	%M2.2	Das Flag wird beim Kaltstart der SPS im OB100 rückgesetzt. Das Flag wird im FB7 gesetzt, nachdem erkannt wurde, dass die Daten im sicheren Baustein gelesen wurden. Erst ab diesem Zeitpunkt dürfen neue Daten im Remanenzspeicher geschrieben werden. Um Fehler zu minimieren, darf dieses Flag nicht in der HMI sichtbar sein.
M_Show_Clock	Bool	%M2.3	Das Flag wird verwendet, da die vom System vergebenen Flags auf den Taktmerkern in der sicheren SPS nicht zur Verfügung stehen.

Die folgenden Ein- und Ausgänge werden verwendet um im Beispielprogramm eine Anzeige für den Status der sicheren Steuerung zu bekommen.

Name	Datentyp	Adresse	Bemerkung
O_Error	Bool	%A0.7	Anzeige Maschine: Fehler in Logon/Logoff System
O_Keys_Logged	Bool	%A1.0	Anzeige Maschine: Es sind Schlüssel gespeichert
O_Machine_Run	Bool	%A1.1	Anzeige Maschine: Maschine darf laufen

2.5.4 Globale Datenbausteine

Datenbaustein DB4_EKS_Data
In diesem Beispiel wird der DB4 als Standard-Datenbaustein angelegt, um die Daten der verwendeten EKS zu halten.

Name	Datentyp	Off-set	Re-ma-nenz	Bemerkung
Read_data_EKS	Ar-ray[1..C_NO_OF_EKS] of „EKS_Key_read_Type“	0	Ja	In diesem Array werden die gelesenen Daten von jedem EKS, sowie die Zustände der einzelnen EKS gehalten. Die Daten sind remanent, damit etwaige Fehler nicht durch einfaches Ein- und Ausschalten der SPS gelöscht werden können.
L_Logon_Buf-fer_Copy	„Log_Data_Type“	44	Ja	Dient als Kopie der geloggten Daten des EKS. Die Daten werden remanent gehalten. Dadurch ist es möglich, aus diesem Speicher die sicheren Daten nach einem Abschalten der Spannung zurückzugewinnen, so das nicht ein einfaches Ausschalten der Steuerung zum Zurücksetzen der Daten führen kann.

Datenbaustein DB2_EKS_Log_DB (safe)
Der DB2 wird als sicherer Datenbaustein angelegt. In diesem Baustein werden die Daten der EKS sicher eingespeichert, die gerade eingeloggt sind. Zudem werden globale Daten für die Sicherheitsfunktionen gehalten.

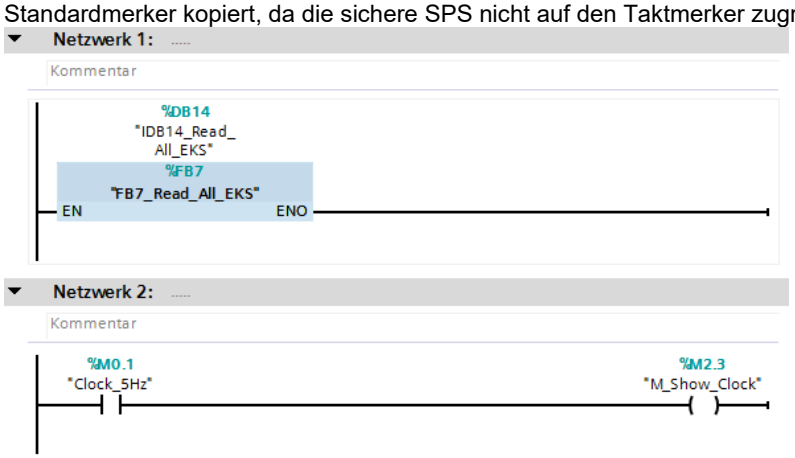
Name	Datentyp	Bemerkung
Log_Data	„Log_Data_Type“	Hier werden die Daten der eingeloggten Schlüssel abgelegt. Die Anzahl der gleichzeitig möglichen eingeloggten Schlüssel wird durch den Datentyp Log_data_Type festgelegt.
M_Error	Bool	Das Flag wird gesetzt, wenn ein beliebiger Fehler vorliegt, der durch eines der folgenden Flags gekennzeichnet wird. Rücksetzen ist nur möglich, wenn alle Fehler entweder verschwunden oder quittiert sind.
M_Error_Delete	Bool	Das Flag wird gesetzt, wenn im FB „FB5_Search_DB_Index“ ein Index ermittelt wurde, der im FB „FB4_EKS_Hanlde_Value“ nicht verfügbar ist. Dieses Flag ist nur durch einen Kaltstart der SPS rücksetzbar.
M_Error_Store	Bool	Das Flag wird gesetzt, wenn im FB „FB5_Search_DB_Index“ ein Index ermittelt wurde, der im FB „FB4_EKS_Hanlde_Value“ nicht verfügbar ist. Dieses Flag ist nur durch einen Kaltstart der SPS rücksetzbar.
M_Error_Memory	Bool	Das Flag wird gesetzt, wenn ein Schlüssel eingespeichert werden soll, jedoch kein Speicherplatz mehr frei ist. Ein Rücksetzen ist nur möglich über die Rücksetzroutine.
M_Machine_Run	Bool	Das Flag wird gesetzt, wenn kein Schlüssel im Speicher eingeloggt ist und auch kein Fehler in den obigen Fehlerbits angezeigt wird. Es wird zurückgesetzt, wenn entweder ein Schlüssel eingespeichert wird oder wenn ein Fehler angezeigt wird.
M_Keys_Logged	Bool	Das Flag wird gesetzt, wenn mindestens ein Schlüssel im Log-Speicher steht. Es wird zurückgesetzt, wenn kein Schlüssel eingespeichert ist. Ein Rücksetzen ist auch möglich über die Rücksetzroutine.
Safe_data_EKS	„Safe_data_EKS_Type“	Hier werden pro EKS die Flags gespeichert

2.6 Programm in der Standard-SPS

Alle beschriebenen Programmteile der SPS sind in der TIA-Library „AP000247 EKS FSA Logon Logoff TIA Vxx.zalxx“ zu finden, die Sie passend zu Ihrer Version des TIA Portals ab V14 von EUCHNER erhalten können.

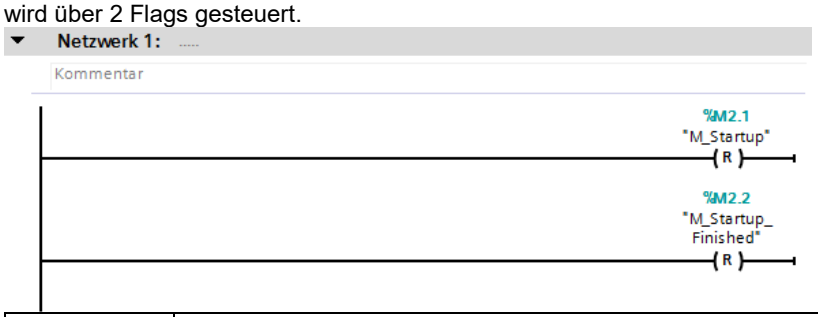
2.6.1 OB1_Main

Vom OB1 oder einem anderen geeigneten Baustein aus wird der FB7_Read_All_EKS aufgerufen. Darüber hinaus ist im Beispiel-OB1 auch die Signalisierung des Zustands der einzelnen EKS sowie der Maschine programmiert. Diese Signalisierung kann beliebig angepasst werden.
Für die sicherheitsrelevante Signalisierung an den verschiedenen EKS wird der Merker mit dem 5-Hz Taktsignal auf einen Standardmerker kopiert, da die sichere SPS nicht auf den Taktmerker zugreifen kann.



2.6.2 OB100_Startup

Da die sicheren Datenbereiche nicht remanent sind, wird eine Kopie des Speichers im nicht sicheren Bereich angelegt. Nach dem Hochlauf der Steuerung muss dieser kopierte Bereich in den sicheren Bereich übertragen werden. Das Kopieren wird über 2 Flags gesteuert.



Netzwerk 1	Die Flags, die den Startup steuern, um die Daten aus dem remanenten Bereich in den sicheren Bereich zu kopieren, werden rückgesetzt.

2.6.3 FB7_Read_all_EKS

Der Funktionsbaustein ist in SCL programmiert, da ein FOR-Loop verwendet wird. Darüber hinaus werden keine Besonderheiten verwendet, sodass der Baustein auch in KOP programmiert werden könnte.

Dieser Funktionsbaustein holt zyklisch alle Daten von den EKS Geräten ab. Aus dem Baustein heraus werden die notwendigen zugehörigen Bausteine FC1_Check_FCS sowie FC2_Check_CRC aufgerufen.
Der zugehörige Instanzenbaustein hat folgende Struktur

		Name	Datentyp	Defaultwert
4		<Hinzufügen>		
5		InOut		
6		<Hinzufügen>		
7		Static		
8		HW_EKS_Id	Array[1.."C_NO_OF_EKS"] of HW_SUBMODULE	
9		HW_EKS_Id[1]	HW_SUBMODULE	"EKS-PN_1~Lesen:_033_Bytes_1"
10		HW_EKS_Id[2]	HW_SUBMODULE	"EKS-PN_2~Lesen:_033_Bytes_1"
11		Temp		
12		I_Status	DWord	
13		I_Count	Int	
14		D_Help_Calc_FCS	DInt	
15		M_Read_Error	Bool	
16		I_help	DWord	
17		Constant		
18		BIT_EKS_KEY_STATUS	Word	16#0200

Die statische Variable HW_EKS_ID wird direkt mit der Definition initialisiert. Der Wert muss passend zur gewählten Hardware der jeweiligen EKS erfolgen. Die Hardware-Kennung ist notwendig, um mittels der Routine GETIO_PART_DB die Daten aus den EKS auszulesen.
Hinweis: Dieses Array muss erweitert werden, wenn mehr als 2 EKS-Lesegeräte zum Einsatz kommen sollen.

Innerhalb des FOR-Loops werden folgende Befehle ausgeführt.

```
8    "GETIO_PART_DB"(ID := #HW_EKS_Id[#I_Count],
9                    OFFSET := 0,
10                   LEN := "C_LEN_EKS_KEY_CONTENT",
11                   STATUS => #I_Status,
12                   ERROR => #M_Read_Error,
13                   INPUTS := "DB4_EKS_Data".Read_data_EKS[#I_Count]);
```

Es werden die Daten vom EKS gelesen.

```
19   IF ((("DB4_EKS_Data".Read_data_EKS[#I_Count].W_Key_Status AND #BIT_EKS_KEY_STATUS) <> #BIT_EKS_KEY_STATUS) OR #M_Read_Error) THEN
20       // Delete all data if key not ready
21       "DB4_EKS_Data".Read_data_EKS[#I_Count].I_EKM_FCS := 0;
22       "DB4_EKS_Data".Read_data_EKS[#I_Count].I_EKS_Facility := 0;
23       "DB4_EKS_Data".Read_data_EKS[#I_Count].I_EKS_KindOfKey := 0;
24       "DB4_EKS_Data".Read_data_EKS[#I_Count].I_EKS_UniqueCode := 0;
25       "DB4_EKS_Data".Read_data_EKS[#I_Count].B_Key_SerialNumber[0] := 0;
26       "DB4_EKS_Data".Read_data_EKS[#I_Count].B_Key_SerialNumber[1] := 0;
27       "DB4_EKS_Data".Read_data_EKS[#I_Count].B_Key_SerialNumber[2] := 0;
28       "DB4_EKS_Data".Read_data_EKS[#I_Count].B_Key_SerialNumber[3] := 0;
29       "DB4_EKS_Data".Read_data_EKS[#I_Count].B_Key_SerialNumber[4] := 0;
30       "DB4_EKS_Data".Read_data_EKS[#I_Count].B_Key_SerialNumber[5] := 0;
31       "DB4_EKS_Data".Read_data_EKS[#I_Count].B_Key_SerialNumber[6] := 0;
32       "DB4_EKS_Data".Read_data_EKS[#I_Count].B_Key_SerialNumber[7] := 0;
33   END_IF;
```

Falls das EKS im Statusbyte meldet, dass die Daten noch nicht gültig sind (Zeile 19), oder wenn ein Fehler beim Lesen aufgetreten ist, werden alle Daten wieder auf Null gesetzt.

```
36   "DB4_EKS_Data".Read_data_EKS[#I_Count].M_FCS_Error := NOT ("FC1_Calc_FCS"(#I_Count));
```

Der Funktionsbaustein FC1_Check_FCS wird aufgerufen. Da die EKS im Index von 1 bis 2 (oder mehr) laufen, muss der Übergabewert um 1 reduziert werden. Falls ein Schlüssel kopiert wurde, kann das mit diesem Baustein festgestellt werden. Das Flag M_FCS_Error wird gesetzt, wenn das Ergebnis anzeigt, dass die Daten nicht korrekt sind.

```
40           "DB4_EKS_Data".Read_data_EKS[#I_Count].M_Wrong_key := "DB4_EKS_Data".Read_data_EKS[#I_Count].I_EKS_Facility <> 0
41           AND "DB4_EKS_Data".Read_data_EKS[#I_Count].I_EKS_Facility <> "C_VALID_FACILITY";
```

Im nächsten Schritt wird geprüft, ob die Schlüsseldaten für die Anlage richtig sind. Der Wert 0 ist richtig (kein Schlüssel gesteckt) sowie der, der für die Anlage in der Konstante C_VALID_FACTORY festgelegt wurde.

```
44   "DB4_EKS_Data".Read_data_EKS[#I_Count].L_EKS_Complete_Code := "DB4_EKS_Data".Read_data_EKS[#I_Count].I_EKS_UniqueCode;
45   "DB4_EKS_Data".Read_data_EKS[#I_Count].L_EKS_Complete_Code :=
46       SHL(IN := "DB4_EKS_Data".Read_data_EKS[#I_Count].L_EKS_Complete_Code, N := 16);
47   "DB4_EKS_Data".Read_data_EKS[#I_Count].L_EKS_Complete_Code :=
48       "DB4_EKS_Data".Read_data_EKS[#I_Count].L_EKS_Complete_Code OR
49       INT_TO_DINT("DB4_EKS_Data".Read_data_EKS[#I_Count].I_EKS_KeyProtect);
```

Mit den Schritten 44 bis 49 wird aus den beiden Einzelcodes I_EKS_UniqueCode und I_EKS_KeyProtect ein Code zur weiteren Verarbeitung in den sicheren Programmteilen gebildet.

Darüber hinaus sorgt die Routine dafür, dass die Daten, die im sicheren Datenbaustein abgelegt wurden, ständig in den Remanenzspeicher der SPS kopiert werden. Das darf jedoch erst dann geschehen, wenn die sichere SPS gemeldet hat, dass die Daten nach einem Kaltstart in den Puffer der sicheren SPS übernommen wurden.

```
47 IF ("M_Startup") THEN
48     "DB4_EKS_Data".L_Logon_Buffer_Copy.Log_Data_0 := "DB2_EKS_Log_DB".Log_Data.Log_Data_0;
49     "DB4_EKS_Data".L_Logon_Buffer_Copy.Log_Data_1 := "DB2_EKS_Log_DB".Log_Data.Log_Data_1;
50     "DB4_EKS_Data".L_Logon_Buffer_Copy.Log_Data_2 := "DB2_EKS_Log_DB".Log_Data.Log_Data_2;
51     "DB4_EKS_Data".L_Logon_Buffer_Copy.Log_Data_3 := "DB2_EKS_Log_DB".Log_Data.Log_Data_3;
52     "DB4_EKS_Data".L_Logon_Buffer_Copy.Log_Data_4 := "DB2_EKS_Log_DB".Log_Data.Log_Data_4;
53     "DB4_EKS_Data".L_Logon_Buffer_Copy.Log_Data_5 := "DB2_EKS_Log_DB".Log_Data.Log_Data_5;
54     "DB4_EKS_Data".L_Logon_Buffer_Copy.Log_Data_6 := "DB2_EKS_Log_DB".Log_Data.Log_Data_6;
55     "DB4_EKS_Data".L_Logon_Buffer_Copy.Log_Data_7 := "DB2_EKS_Log_DB".Log_Data.Log_Data_7;
56     "DB4_EKS_Data".L_Logon_Buffer_Copy.Log_Data_8 := "DB2_EKS_Log_DB".Log_Data.Log_Data_8;
57     "DB4_EKS_Data".L_Logon_Buffer_Copy.Log_Data_9 := "DB2_EKS_Log_DB".Log_Data.Log_Data_9;
58
59     // After startup from safe control mark that startup is now completed.
60     "M_Startup_Finished" := "M_Startup";
61 END_IF;
```

2.6.4 FC1_Check_FCS

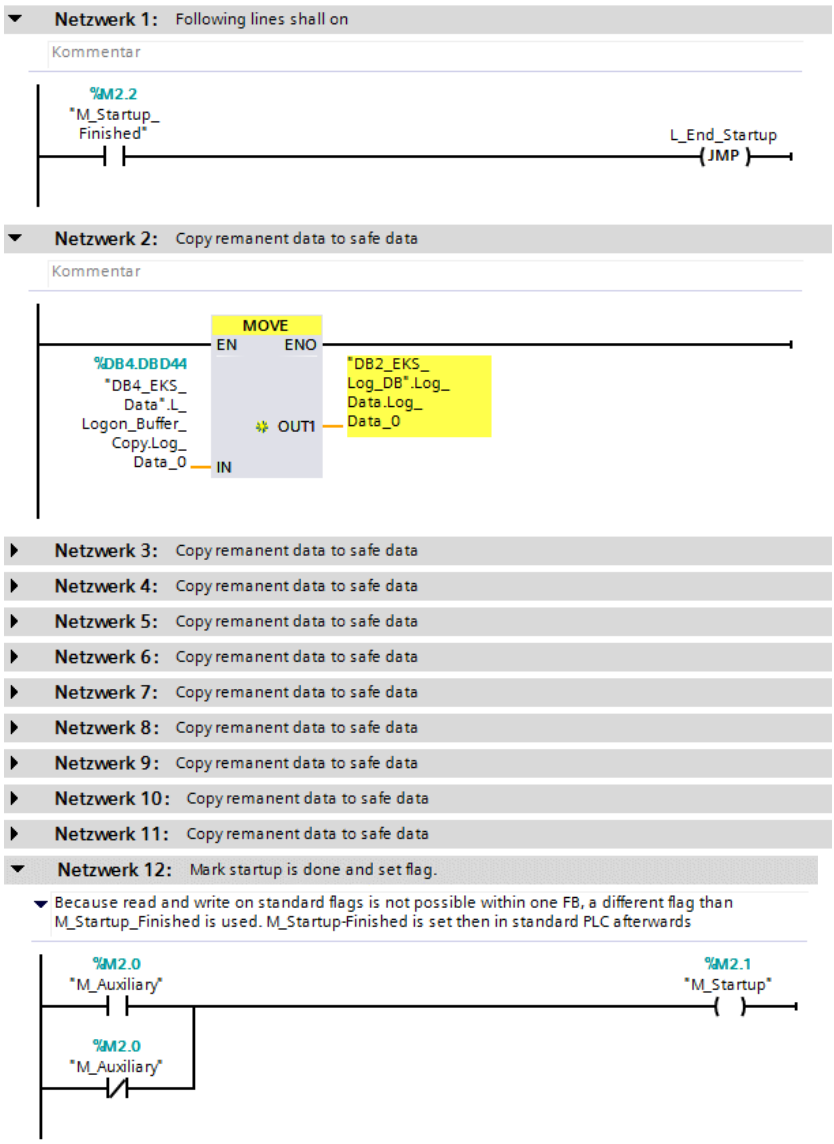
Die Routine dient dazu, kopierte EKS-Schlüssel zu erkennen, die ohne den Einsatz von EKM und KeyProtect erzeugt wurden. Der Aufruf erfolgt direkt aus der Routine FB7_Read_All_EKS heraus.
Schlüssel, die mit dem EKM beschrieben wurden, enthalten in den Schlüsseldaten an der Stelle KEYCRC eine Prüfsumme. Diese Prüfsumme wird über bestimmte Speicherbereiche des Schlüssels sowie über die eindeutige Schlüsselkennung gebildet. Da die Schlüsselkennung nicht mit kopiert werden kann, weist die Prüfsumme bei kopierten EKS-Schlüsseln einen falschen Wert auf.
Die Routine ist in SCL geschrieben.
Hinweis: Falls eine andere, als die hier vorgestellte Schlüsselstruktur verwendet werden soll, muss diese Routine angepasst werden. Für nähere Hinweise hierzu verwenden Sie bitte die „Zusatzdokumentation zum Electronic Key Manager EKM“.

2.7 Programm in der Sicherheits-SPS

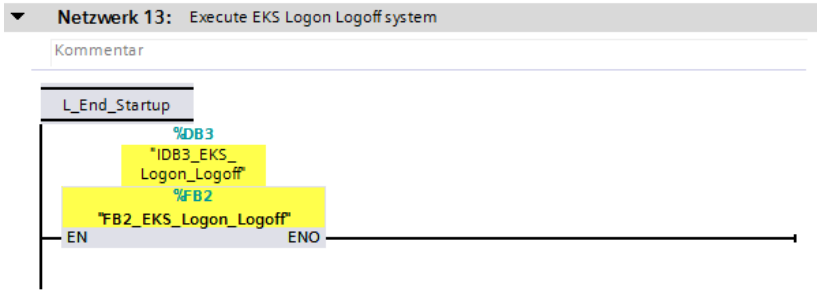
2.7.1 FB1_Main_Safety_RTG1 (safe)

In diesem oder einem anderen zyklisch aufgerufenen Baustein in der sicheren SPS wird der Funktionsbaustein zum sicheren Ein- und Ausloggen von Schlüsseln aufgerufen.

In den Netzwerken 1 bis 11 wird der Kaltstart des Systems bearbeitet. Beim Hochlaufen werden einmalig die Daten aus dem Remanenzspeicher der SPS ausgelesen und in die sicheren Daten eingespeichert. Da die SPS einen Fehler aufweisen könnte, kann der Remanenzspeicher gelöscht werden. Das würde dazu führen, dass die Anlage nach einem Hochlaufen der Steuerung immer starten könnte. Wenn dieser Fehler ausgeschlossen werden soll, könnte stattdessen beim Hochlaufen das Flag DB2_EKS_Log_DB.M_Error_Memory gesetzt werden. Dieses Bit lässt sich ausschließlich über die Resetroutine zurücksetzen und erzwingt somit nach dem Maschinenstart den Reset der Anlage über die Resetprozedur.



Netzwerk 1	Es wird geprüft, ob die Startup Routine bereits abgelaufen ist.
Netzwerk 2 bis 11	Die Daten werden aus dem nicht sicheren, aber remanenten Bereich in den sicheren Bereich kopiert. **Hinweis**: Sollen mehr als 10 Schlüssel eingeloggt werden können, müssen dementsprechend weitere MOVE-Befehle eingeführt werden.
Netzwerk 12	Das Kopieren wird als abgeschlossen markiert. Der direkte SET-Befehl ist auf Merker im Standardbereich nicht zugelassen.



Netzwerk 13	Es wird zyklisch der Logon-Logoff Baustein aufgerufen.

2.7.2 FB2_EKS_Logon_Logoff (safe)

Die folgenden Daten werden im Baustein angelegt.

7	Static		
8	M_State_EKS_1_No_Data	Bool	false
9	M_State_EKS_1_Evaluated	Bool	true
10	M_State_EKS_2_No_Data	Bool	false
11	M_State_EKS_2_Evaluated	Bool	true
12	M_Detect_EKS_1_In	Bool	false
13	M_Detect_EKS_2_In	Bool	false
14	M_Detect_EKS_1_Out	Bool	false
15	M_Detect_EKS_2_Out	Bool	false
16	M_Store_EKS_1	Bool	false
17	M_Delete_EKS_1	Bool	false
18	M_Store_EKS_2	Bool	false
19	M_Delete_EKS_2	Bool	false
20	M_Detect_Edge	Bool	false
21	Temp		
22	M_Store_DB	Bool	
23	M_Delete_DB	Bool	
24	I_Index	Int	
25	Constant		
26	C_DINT_DATA_ZERO	DInt	0
27	C_EKS_TIMEOUT	DInt	500

Statische Variable:

Name	**Da-tentyp**	**Bemerkung**
M_State_EKS_1_No_Data	Bool	Das Flag wird gesetzt, wenn erkannt wird, dass in L_EKS_CompleteCode vom EKS 1 der Wert Null vom EKS gesendet wird (kein Schlüssel). Das Flag wird rückgesetzt, wenn im Feld L_EKS_CompleteCode vom EKS 1 ein anderer Wert als Null steht.
M_State_EKS_1_Evaluated	Bool	Das Flag wird gesetzt, wenn erkannt wird, dass in L_EKS_CompleteCode vom EKS 1 ausgewertet wurde. Das Flag wird rückgesetzt, wenn im Feld L_EKS_CompleteCode vom EKS 1 der Wert Null steht. Dieses Flag wird mit dem Wert TRUE initialisiert, damit ein Schlüssel, der beim Einschalten der Anlage steckt, nicht automatisch bearbeitet wird. Der Bediener der Anlage muss den Schlüssel aktiv stecken. Damit wird ein versehentliches Ausloggen verhindert.
M_State_EKS_2_No_Data	Bool	Das Flag wird gesetzt, wenn erkannt wird, dass in L_EKS_CompleteCode vom EKS 2 der Wert Null vom EKS gesendet wird (kein Schlüssel). Das Flag wird rückgesetzt, wenn im Feld L_EKS_CompleteCode vom EKS 2 ein anderer Wert als Null steht.
M_State_EKS_2_Evaluated	Bool	Das Flag wird gesetzt, wenn erkannt wird, dass in L_EKS_CompleteCode vom EKS 2 ausgewertet wurde. Das Flag wird rückgesetzt, wenn im Feld L_EKS_CompleteCode vom EKS 2 der Wert Null steht. Dieses Flag wird mit dem Wert TRUE initialisiert, damit ein Schlüssel, der beim Einschalten der Anlage steckt, nicht automatisch bearbeitet wird. Der Bediener der Anlage muss den Schlüssel aktiv stecken. Damit wird ein versehentliches Ausloggen verhindert.
M_Detect_EKS_1_In	Bool	Das Flag wird gesetzt, wenn durch einen Schlüssel im EKS 1 der FSA-Ausgang gesetzt wird und eine Zeit abgelaufen ist. Das Flag wird rückgesetzt, wenn der FSA-Ausgang des EKS 1 rückgesetzt wird.
M_Detect_EKS_2_In	Bool	Das Flag wird gesetzt, wenn durch einen Schlüssel im EKS 2 der FSA-Ausgang gesetzt wird und eine Zeit abgelaufen ist. Das Flag wird rückgesetzt, wenn der FSA-Ausgang des EKS 2 rückgesetzt wird.
M_Detect_EKS_1_Out	Bool	Das Flag wird gesetzt, wenn der FSA-Ausgang des EKS 1 rückgesetzt wird und eine Zeit abgelaufen ist. Das Flag wird rückgesetzt, wenn durch einen Schlüssel im EKS 1 der FSA-Ausgang gesetzt wird.
M_Detect_EKS_2_Out	Bool	Das Flag wird gesetzt, wenn der FSA-Ausgang des EKS 2 rückgesetzt wird und eine Zeit abgelaufen ist. Das Flag wird rückgesetzt, wenn durch einen Schlüssel im EKS 2 der FSA-Ausgang gesetzt wird.

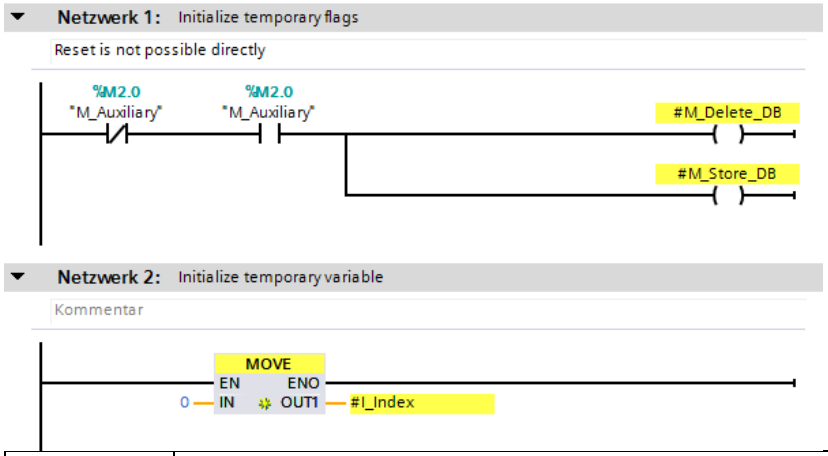
M_Store_EKS_1	Bool	Das Flag wird gesetzt, wenn erkannt wird, dass die Daten L_EKS_CompleteCode vom EKS 1 eingespeichert werden müssen. Das Flag wird bedingungslos am Anfang dieser Routine rückgesetzt.
M_Delete_EKS_1	Bool	Das Flag wird gesetzt, wenn erkannt wird, dass die Daten L_EKS_CompleteCode vom EKS 1 gelöscht werden können. Das Flag wird bedingungslos am Anfang dieser Routine rückgesetzt.
M_Store_EKS_2	Bool	Das Flag wird gesetzt, wenn erkannt wird, dass die Daten L_EKS_CompleteCode vom EKS 2 eingespeichert werden müssen. Das Flag wird bedingungslos am Anfang dieser Routine rückgesetzt.
M_Delete_EKS_2	Bool	Das Flag wird gesetzt, wenn erkannt wird, dass die Daten L_EKS_CompleteCode vom EKS 2 gelöscht werden können. Das Flag wird bedingungslos am Anfang dieser Routine rückgesetzt.
M_Detect_Edge	Bool	Das Flag dient als Hilfsflag zur Erkennung einer fallenden Flanke. Diese Variable wird nur einmal benötigt.

Hinweis: Alle Variablen, die im Namen EKS_1 tragen, müssen je einmal pro EKS angelegt werden. Sollen also mehr als 2 EKS eingesetzt werden, müssen diese Variablen dementsprechend angelegt werden.

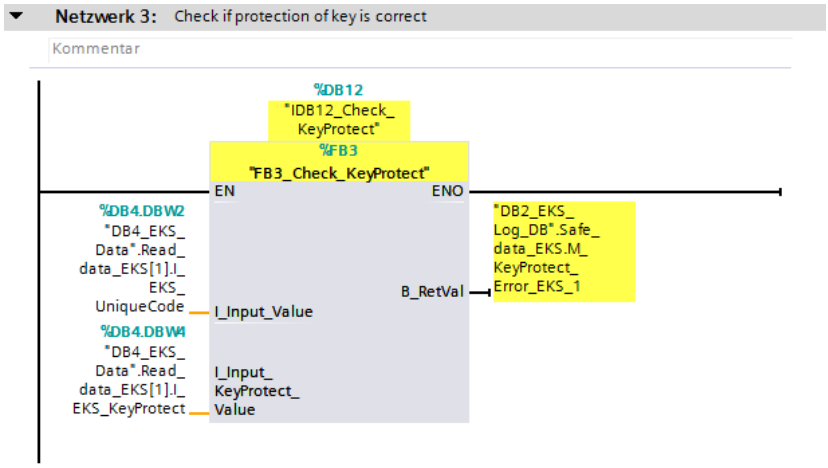
Temporäre Variablen:
Die temporären Daten sind reine Hilfswerte zum Aufruf der untergeordneten FBs.

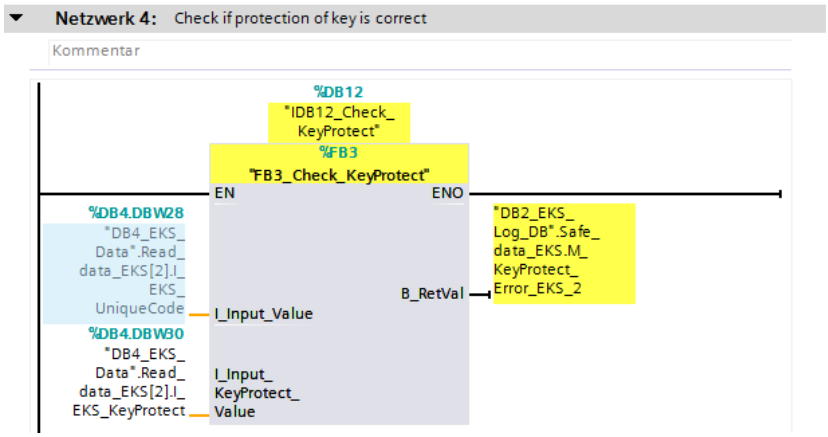
Konstanten:

Name	**Datentyp**	**Wert**	**Bemerkung**
C_LONG_DATA_ZERO	DInt	0	Wert 0
C_EKS_TIMEOUT	DInt	500	Maximal akzeptierte Zeit nach der die Übereinstimmung zwischen dem Datenkanal des jeweiligen EKS mit dem zugehörigen Ausgang LA vorhanden sein muss. Andernfalls wird ein Fehler gemeldet.

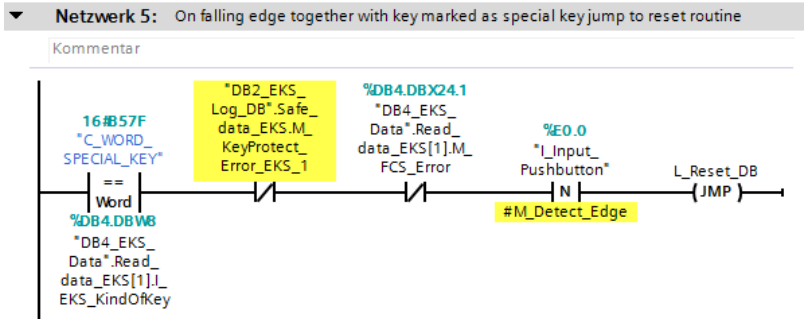


Netzwerk 1	In der sicheren SPS müssen temporäre Variablen initialisiert werden, bevor sie genutzt werden können. Beide Flags werden gelöscht, da die Daten noch nicht gespeichert und auch nicht gelöscht wurden.
Netzwerk 2	In der sicheren SPS müssen temporäre Variablen initialisiert werden, bevor sie genutzt werden können. Der Wert wird auf Null gesetzt.



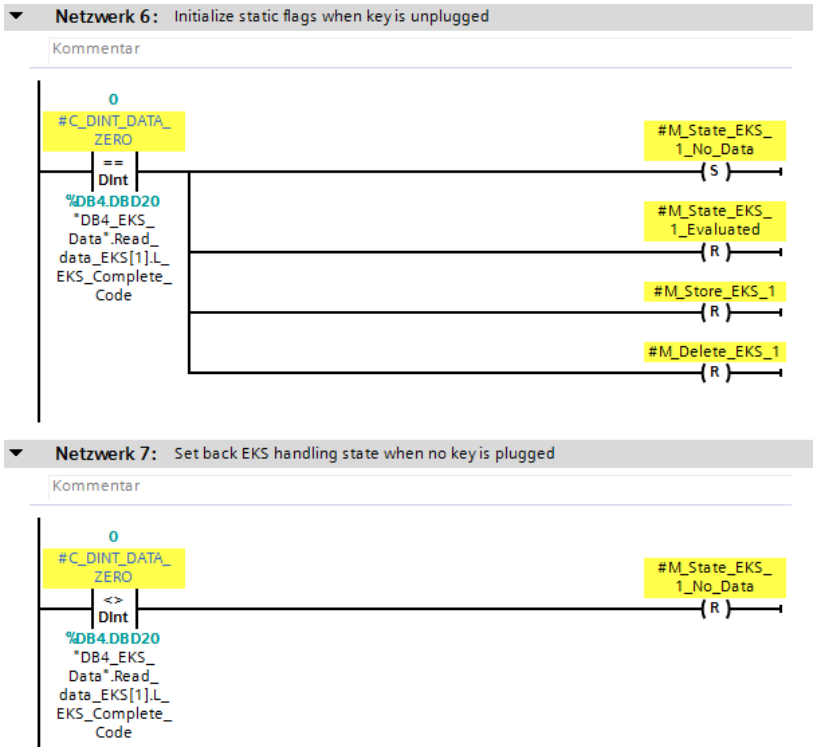


Netzwerk 3 und 4	Mit den Eingangsdaten I_EKS_UniqueCode und I_EKS_KeyProtect vom jeweiligen EKS wird die Routine zur Überprüfung der Daten aufgerufen. Der Rückgabewert muss im sicheren Flag im DB2 eingespeichert werden. Im Fehlerfall muss entsprechend reagiert werden, was in der Routine zum Speichern und Löschen von Daten erfolgt.

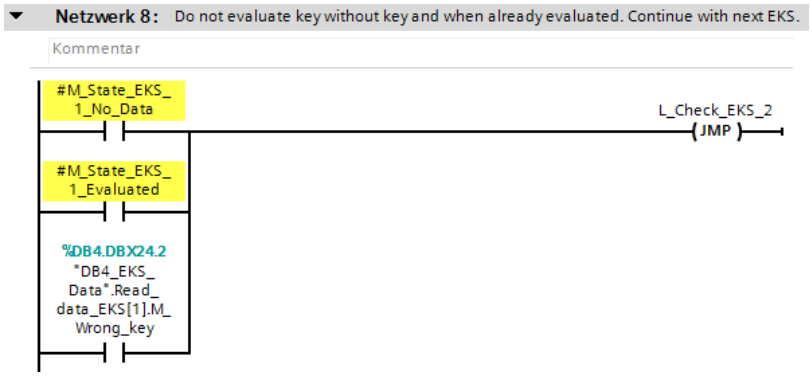


Netzwerk 5	Wenn der Reset-Schlüssel im ersten EKS steckt und der Schlüssel weder kopiert ist, noch im Feld I_EKS_1_Unique_Code einen Fehler aufweist, kann ein Rücksetzen der Anlage erlaubt werden. In diesem Beispiel wird zusätzlich die fallende Flanke an einem Eingang der SPS verwendet. Im zweiten EKS kann **kein** Rücksetzen der Anlage gestartet werden. Dies kann durch eine beliebige Prozedur ersetzt werden und auch über z.B. ein HMI gesteuert sein. Eine Autorisierung muss in jedem Fall vorhanden sein. **ACHTUNG:** Eine fallende Flanke muss immer verwendet werden, da diese Funktion der Sicherheitsfunktion „Manuelle Rückstellungsfunktion“ nach EN ISO 13849-1 entspricht. Die Fehlerlöschroutine sollte nur einmalig nach Anforderung aufgerufen werden, da alle Fehlerbits gelöscht werden. Diese müssen aber, falls der Fehler weiterhin vorhanden ist, sofort wieder gesetzt werden. Ohne alle Bedingungen zum Rücksetzen zu erfüllen, wird der Reset-Schlüssel wie jeder andere Standardschlüssel behandelt, also auch in die Anlage eingeloggt. Dadurch muss kein zweiter Schlüssel an einen Bediener mit besonderen Berechtigungen ausgegeben werden.

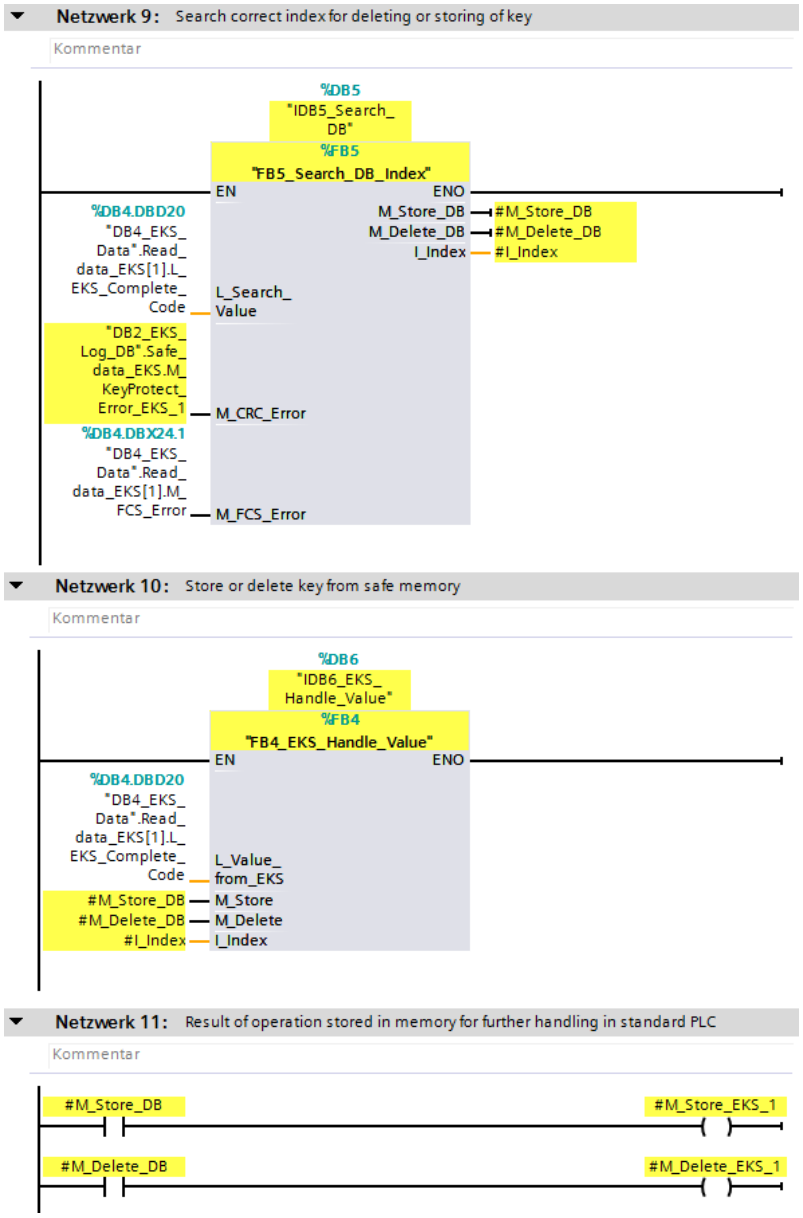
ACHTUNG: In der Betriebsanleitung der Anlage muss auf die spezielle Bedeutung des Rücksetzens der Anlage hingewiesen werden! Insbesondere ist die Verpflichtung aufzuführen, das geprüft werden muss, das sich niemand in der Anlage aufhält!



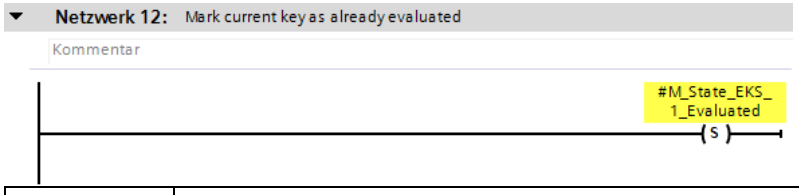
Netzwerk 6	Es wird geprüft, ob sich im EKS 1 kein Schlüssel befindet. In diesem Fall wird das Flag gesetzt, das kennzeichnet, dass keine Daten anliegen und es wird zudem ermöglicht, einen neuen Schlüssel auszuwerten. Die Flags werden rückgesetzt, die anzeigen, auf welche Weise ein Schlüssel ausgewertet wurde.
Netzwerk 7	Es wird geprüft, ob sich Daten auf dem Schlüssel im EKS 1 befinden. Das wird im entsprechenden Flag angezeigt.



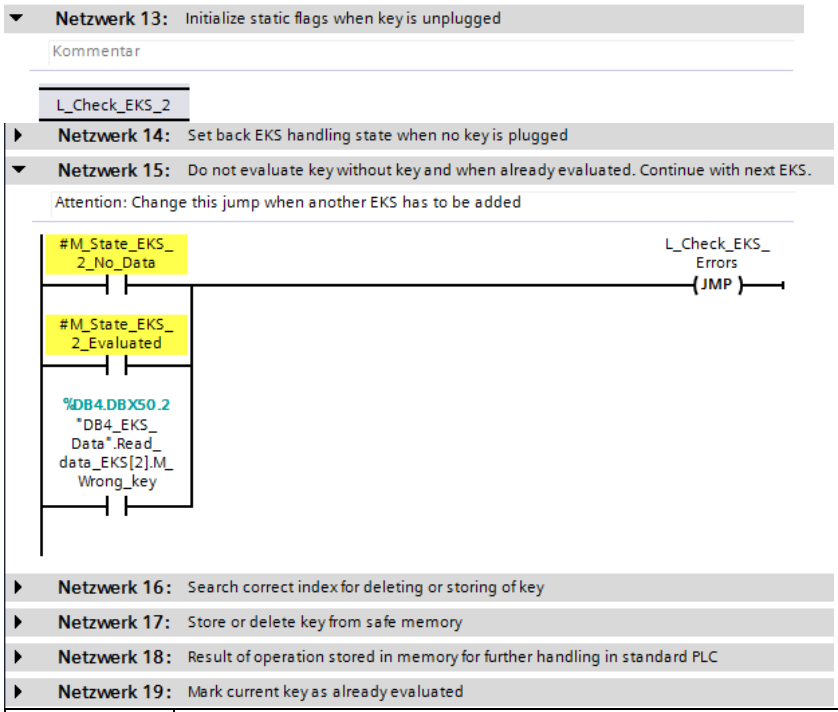
Netzwerk 8	Sofern kein Schlüssel im EKS 1 steckt, gekennzeichnet dadurch, dass der Wert von I_EKS_1_Unique_Code Null ist, oder wenn die Daten auf dem Schlüssel bereits ausgewertet wurden, wird auf die Abarbeitung des nächsten EKS gesprungen. In diesem Beispiel werden ungültige Schlüssel nicht gespeichert. Deshalb wird auch in diesem Fall die Abarbeitung übersprungen. **Hinweis**: Das Verhalten der Anlage bei Stecken eines ungültigen Schlüssels muss vom Konstrukteur beurteilt werden und kann sich von Fall zu Fall unterscheiden.



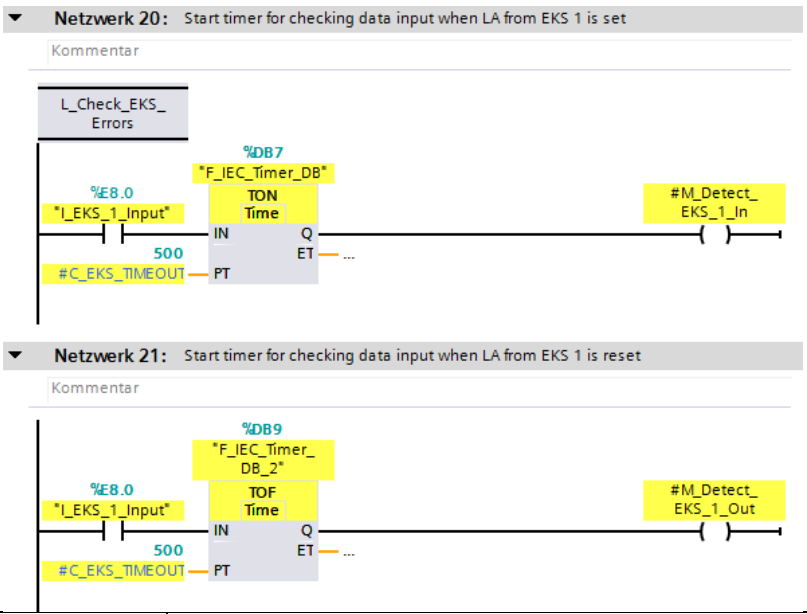
Netzwerk 9	Die Routine zum Prüfen, ob die Daten des Schlüssels gespeichert werden oder aber gelöscht, wird aufgerufen. Die Routine ermittelt auch den Platz in den Daten, auf dem gespeichert wird, oder der gelöscht werden kann.
Netzwerk 10	Mit dem Ergebnis von FB5_Search_DB_Index wird die Routine zur Abarbeitung der Schlüsseldaten aufgerufen.
Netzwerk 11	Das Ergebnis der Bearbeitung der Schlüsseldaten wird im statischen Bereich gespeichert und es wird in die Flags zur Anzeige der Bearbeitung kopiert.



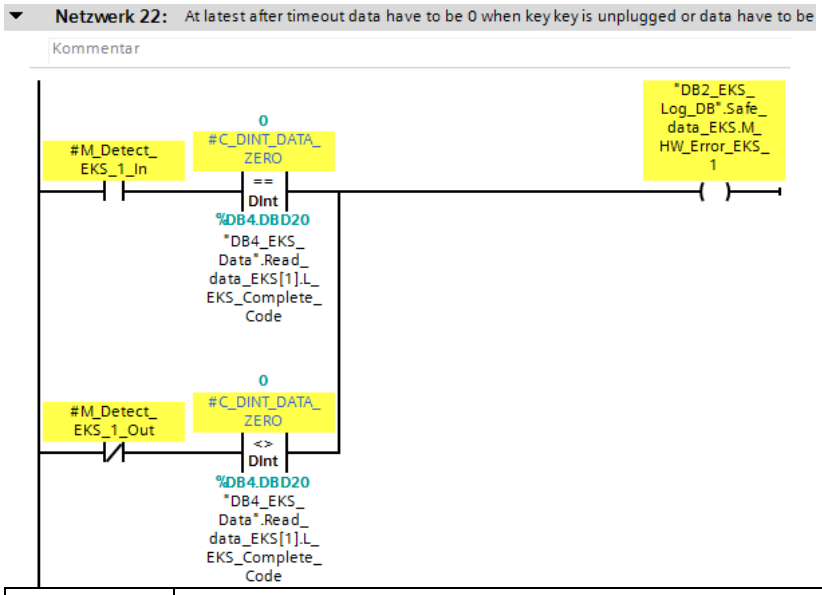
Netzwerk 12	Die Schlüsseldaten werden nun als bearbeitet markiert, damit nicht eine zweites Mal ohne erneutes Stecken des Schlüssels dieselben Daten bearbeitet werden.



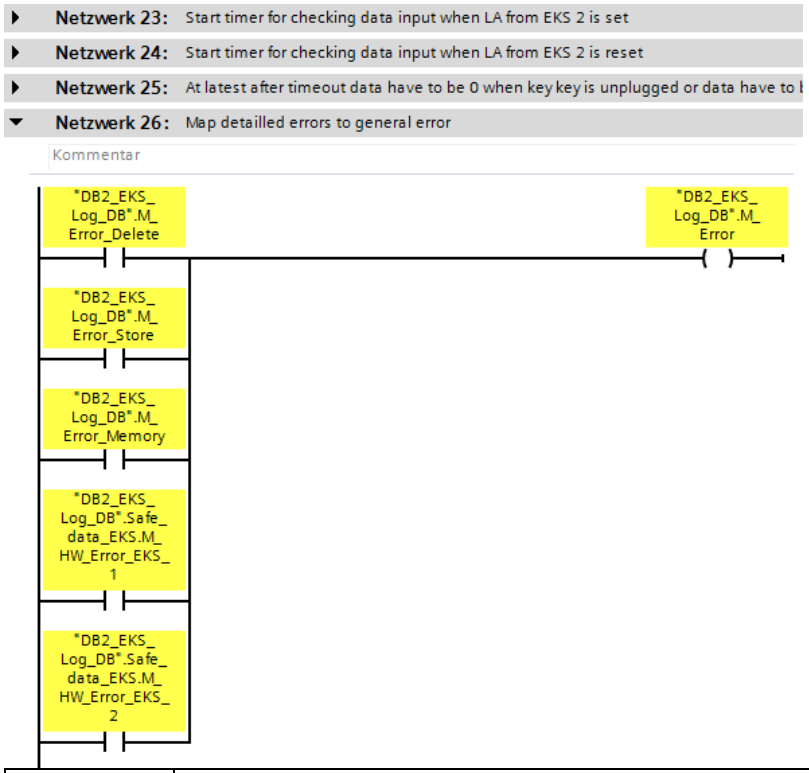
Netzwerke 13 bis 19	Ab hier wiederholen sich die Schritte von Netzwerk 4 bis Netzwerk 10, jedoch für das EKS 2. Wenn ein weiteres EKS eingefügt werden soll, müssen all diese Netzwerke identisch für das weitere EKS eingefügt werden. Im Netzwerk 13 erfolgt im Beispiel ein Sprung auf die Fehlerprüfroutine. Dieser Sprung muss bei Bedarf auf die Abarbeitung des nächsten EKS geändert werden.



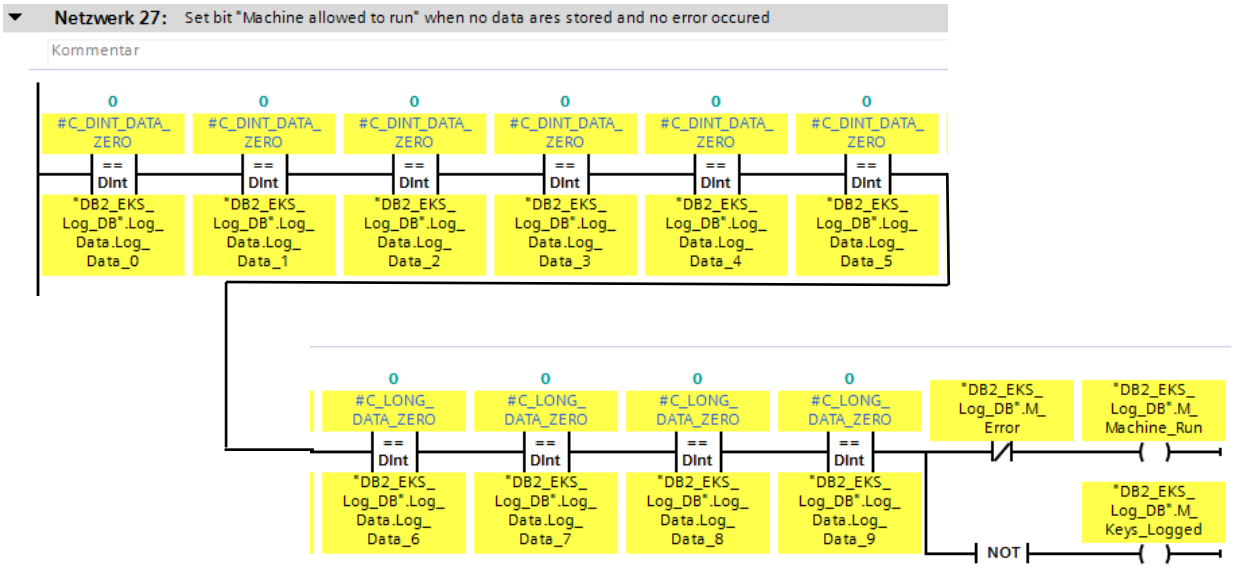
Netzwerk 20	Nach Erkennen des gesetzten FSA-Ausgangs von EKS 1 wird eine Zeit gestartet. Diese Zeit hängt davon ab, wie lange die Durchlaufzeit der Daten vom EKS bis zur sicheren SPS ist.
Netzwerk 21	Nach Erkennen des rückgesetzten FSA-Ausgangs von EKS 1 wird eine Zeit gestartet. Diese Zeit hängt davon ab, wie lange die Durchlaufzeit der Daten vom EKS bis zur sicheren SPS ist.



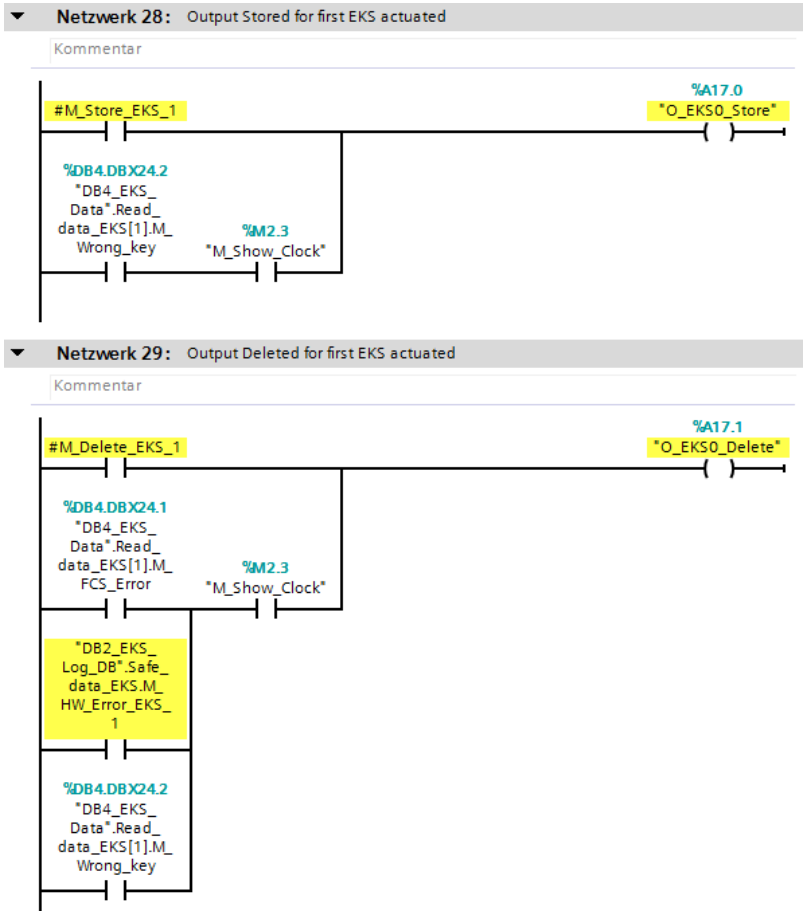
Netzwerk 22	Wenn jeweils nach Ablauf der Zeit der Zustand der Daten nicht passend zum FSA-Ausgang ist, wird das Fehlerbit des EKS 1 gesetzt. Wenn alles wieder richtig abläuft, wird das Bit automatisch zurückgesetzt. Somit ist dieser Fehler selbstquittierend.



Netzwerke 23 bis 25	Entsprechen den Netzwerken 18 bis 20, jedoch für das zweite EKS. **Hinweis**: Wenn weitere EKS verwendet werden sollen, müssen diese Fehlerprüfroutinen ergänzt werden.
Netzwerk 26	Wenn ein beliebiger Fehler an einem beliebigen EKS oder beim Bearbeiten eines Schlüssels aufgetreten ist, wird ein Fehler für das gesamte System eingetragen. Es wird kein Set-Befehl verwendet, sodass für jeden einzelnen Fehler entschieden werden kann, ob der Fehler rastend ist, oder durch beheben der Ursache sich selbst löscht.

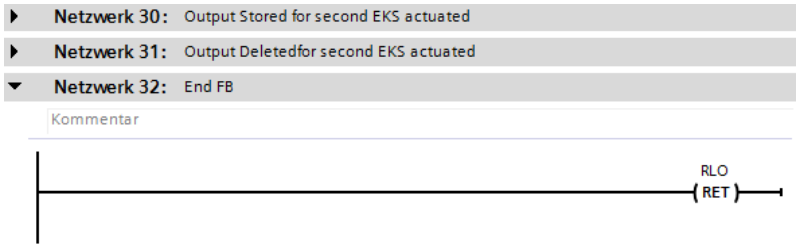


Netzwerk 27	Wenn keine Daten im Speicher stehen und kein Fehler im System aufgetreten ist, wird das Bit M_Machine_Run gesetzt. ACHTUNG: Nur wenn dieses Bit gesetzt ist, darf die Maschine laufen, sofern ein Zugang zur Maschine möglich ist. Das Bit M_Keys_Logged wird gesetzt, wenn mindestens ein Schlüssel im Speicher eingeloggt ist.

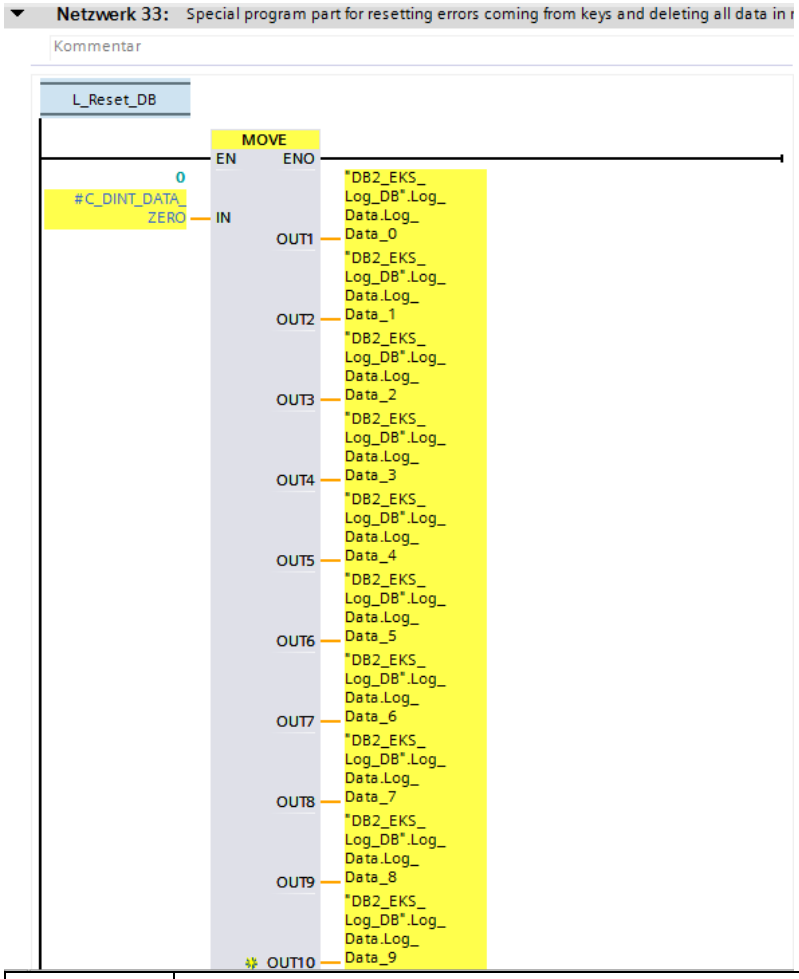


Netzwerk 28	Der sichere Ausgang zur Anzeige, ob der im ersten EKS gesteckte Schlüssel gespeichert wurde, wird aktualisiert. Im Fehlerfall wird ein Blinksignal ausgegeben.

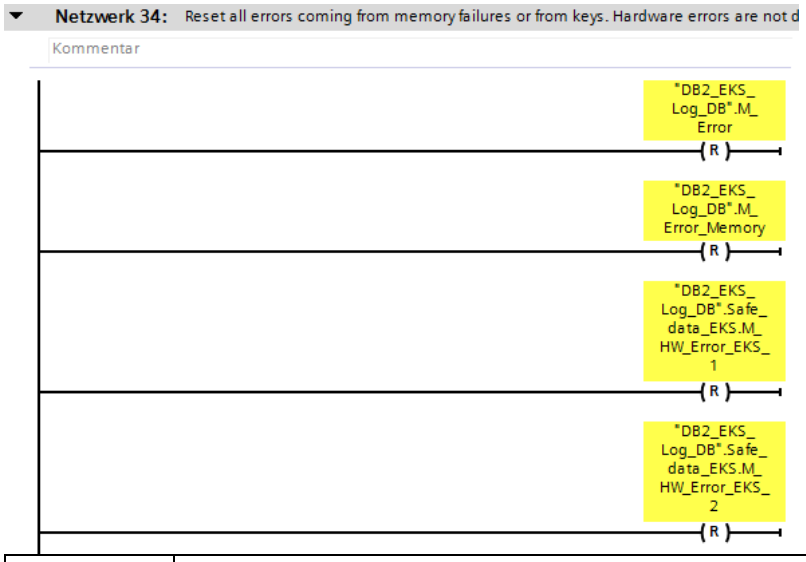
Netzwerk 29	Der sichere Ausgang zur Anzeige, ob der im ersten EKS gesteckte Schlüssel gelöscht wurde, wird aktualisiert. Im Fehlerfall wird ein Blinksignal ausgegeben.



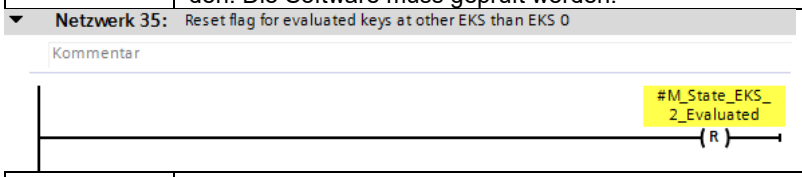
Netzwerk 30 und 31	Entsprechen den Netzwerken 26 und 27 für das zweite EKS **Hinweis**: Für weitere EKS müssen hier die weiteren Anzeigen eingefügt werden.
Netzwerk 32	Ende der Logon-Logoff Routine



Netzwerk 33	Die Speicherinhalte werden vollständig gelöscht. **Hinweis**: Wenn der Speicher vergrößert werden soll, müssen hier die weiteren Speicherstellen ergänzt werden.



Netzwerk 34	Es werden alle Fehlerflags mit Ausnahme der Bits M_Delete und M_Store gelöscht. Wenn eines dieser beiden Flags gesetzt ist, hat ein FB falsche Ergebnisse geliefert. Dieser Fehler kann nicht behoben werden. Die Software muss geprüft werden!



Netzwerk 35	Rücksetzen des Flags zur Kennzeichnung eines bearbeiteten Schlüssels in einem anderen EKS. Damit wird ein eventuell noch gesteckter Schlüssel sofort wieder eingeloggt.

Hinweis: Falls nicht EKS0 als bevorzugtes EKS zum Rücksetzen von Fehlern genutzt wird, muss hier EKS0 ebenfalls rückgesetzt werden. Falls weitere EKS eingebunden werden sollen, müssen auch diese rückgesetzt werden.
Hinweis: In dieser Routine können auch etwaige Hardwarefehler korrigiert werden, indem der Baustein ACK_GL verwendet wird.

2.7.3 FB5_Search_DB_Index (safe)

Die folgenden Variablen werden im Baustein angelegt.

	Name	Datentyp	Defaultwert
1	Input		
2	_L_Search_Value	DInt	0
3	M_CRC_Error	Bool	false
4	M_FCS_Error	Bool	false
5	Output		
6	M_Store_DB	Bool	false
7	M_Delete_DB	Bool	false
8	I_Index	Int	0
9	InOut		
10	<Hinzufügen>		
11	Static		
12	<Hinzufügen>		
13	Temp		
14	M_Value_Found	Bool	
15	Constant		
16	LONG_DAT_ZERO	DInt	0

Eingangswerte

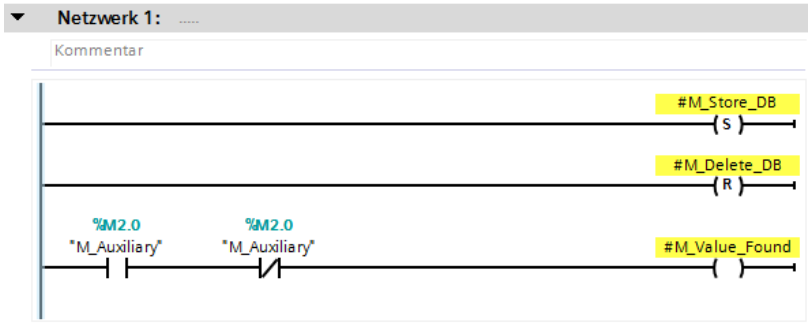
Name	Datentyp	Bemerkung
L_Search_Value	DInt	Der Wert, der vom Schlüssel gelesen wurde.
M_CRC_Error	Bool	Flag, das anzeigt, ob die Prüfung des Schlüssel keinen Fehler ergeben hat.
M_FCS_Error	Bool	Flag, das anzeigt, ob der Schlüssel kopiert wurde.

Ausgangswerte

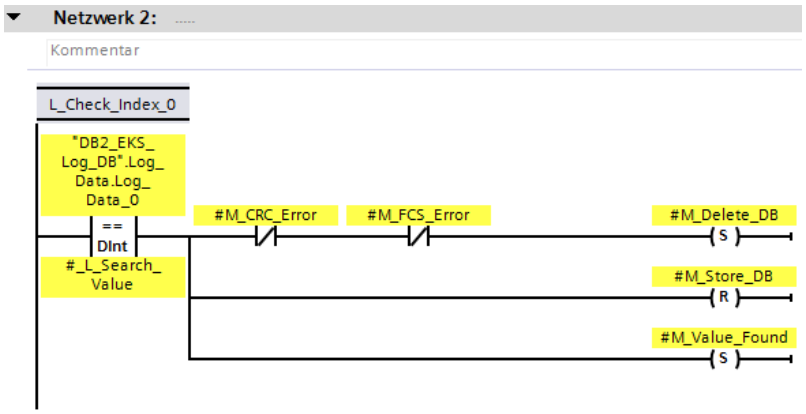
Name	Datentyp	Bemerkung
M_Store_DB	Bool	Flag, das anzeigt, ob der Wert eingespeichert werden muss.
M_Delete_DB	Bool	Flag, das anzeigt, ob der Wert aus dem Speicher gelöscht werden darf.
I_Index	Int	Stelle im sicheren Datenbereich, an der entweder gespeichert oder gelöscht werden kann.

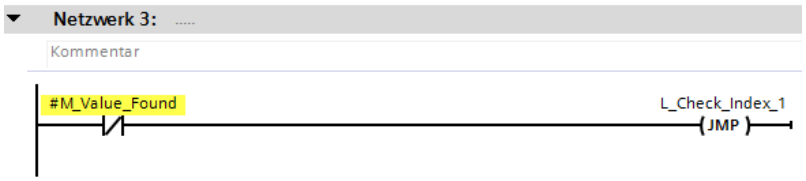
Temporäre Werte

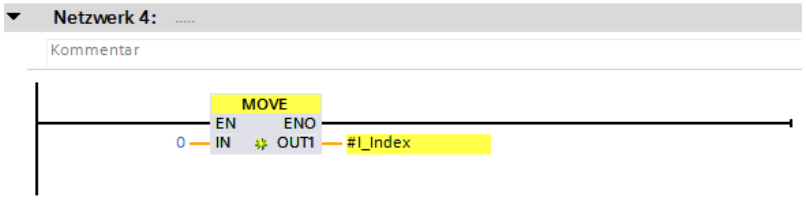
Name	Datentyp	Bemerkung
M_Value_Found	Bool	Flag, das anzeigt, ob der Wert gefunden wurde

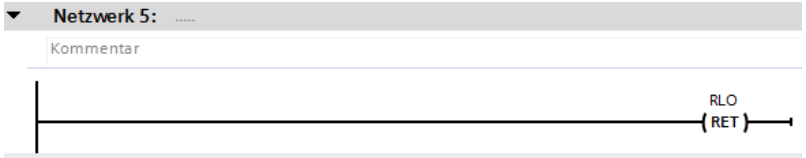


Netzwerk 1	Die Ausgangsdaten werden initialisiert. Es wird gekennzeichnet, daß ein Wert eingespeichert wird, da nur mit gespeicherten Werten die Anlage nicht gestartet werden kann. Die temporäre Variable M_Value_Found wird rückgesetzt. Ein direktes Rücksetzen ist bei temporären Variablen nicht möglich.

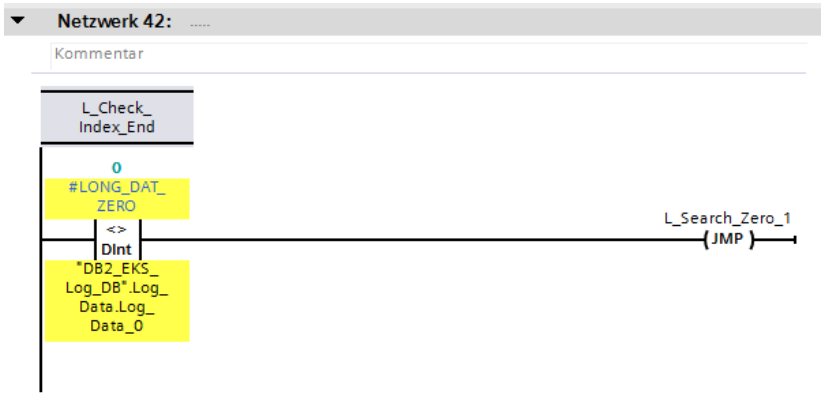


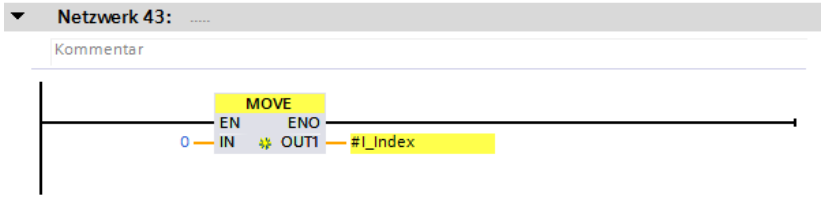


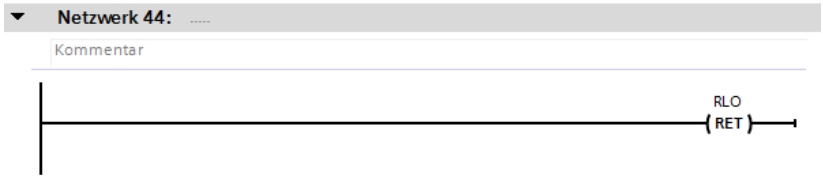


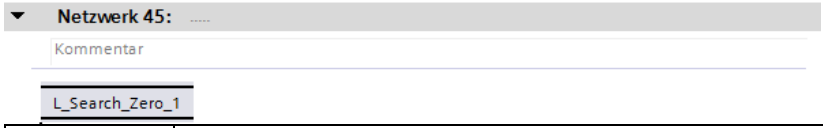


Netzwerk 2	In diesem Schritt findet der Vergleich statt, ob der übergebene Wert an der Stelle 0 in den Daten gespeichert ist. Falls der Wert übereinstimmt, wird die temporäre Variable M_Value_Found gesetzt und der Rückgabewert M_Store_DB rückgesetzt. Der Rückgabewert, der das Löschen erlaubt, wird nur dann gesetzt, wenn die Daten keinen Fehler aufweisen und der Schlüssel nicht kopiert ist.
Netzwerk 3	Falls es keine Übereinstimmung im vorigen Schritt gab, wird die nächste Speicherstelle geprüft.
Netzwerk 4	Der Index der gefundenen Stelle wird zurückgegeben.
Netzwerk 5	Nachdem der Wert gefunden wurde, kann die Routine beendet werden.
Netzwerk 6 bis 41	Ab hier werden die 4 Schritte von Netzwerk 2 bis 5 für jede Speicherstelle im sicheren Datenbaustein wiederholt. **Hinweis:** Diese 4 Schritte müssen für jede weitere gewünschte Speicherstelle programmiert werden.

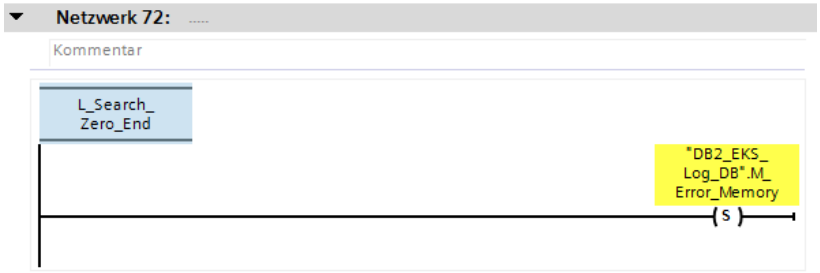








Netzwerk 42	Wenn bis hierher keine Übereinstimmung mit den Schlüsseldaten gefunden wurde, wird die erste freie Stelle mit einer Null gesucht. Auf diese Stelle kann der neue Wert eingespeichert werden. In diesem Schritt wird die erste Speicherstelle geprüft, ob der Wert 0 in dieser Stelle steht. Falls nein, wird die nächste Stelle geprüft.
Netzwerk 43	Es wurde eine Null gefunden, deshalb wird als Index die Stelle 0 zurückgegeben. Die anderen Rückgabewerte im Netzwerk 1 sind richtig initialisiert.
Netzwerk 44	Nachdem ein freier Speicherplatz gefunden wurde, ist die Routine beendet.
Netzwerk 45 bis 71	Die 3 Schritte aus den Netzwerken 42 bis 45 werden für jede Speicherstelle wiederholt. **Hinweis:** Diese 3 Schritte müssen für jede weitere gewünschte Speicherstelle programmiert werden.



Netzwerk 72	Es wurde keine Stelle gefunden, die den gesuchten Wert enthält und es wurde auch keine freie Speicherstelle gefunden, in die der Wert eingespeichert werden konnte. Ursache dafür dürfte ein Schlüssel mehr sein, als gespeichert werden konnte. Das Fehlerflag M_Error_Memory wird gesetzt. Dieser Fehler darf nicht selbst quittierend sein, da mindestens ein Schlüssel mehr als möglich eingespeichert werden soll. Ein Rücksetzen darf nur manuell über die Fehler-Rücksetzroutine erfolgen.

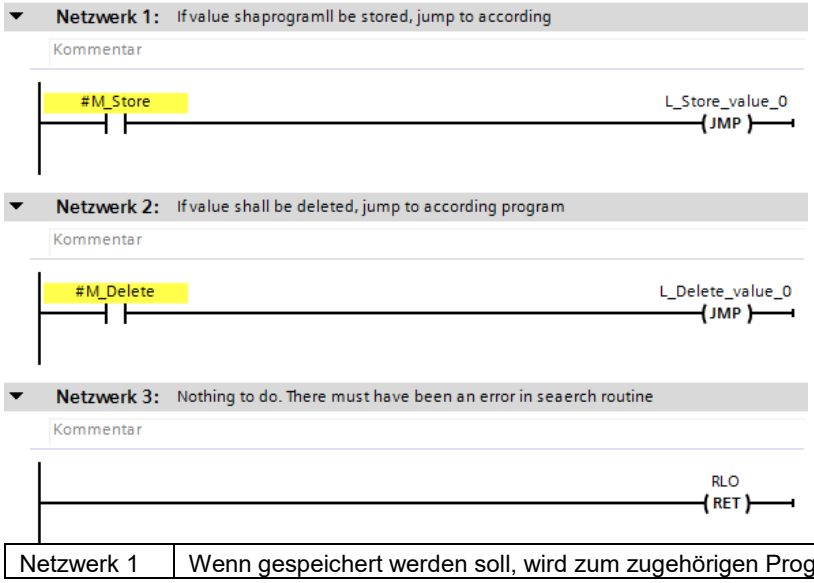
2.7.4 FB4_EKS_Handle_Value (safe)

Die folgenden Variablen werden im Baustein angelegt.

	Name	Datentyp	Defaultwert
1	Input		
2	L_Value_from_EKS	DInt	0
3	M_Store	Bool	false
4	M_Delete	Bool	false
5	I_Index	Int	0
6	Output		
7	<Hinzufügen>		
8	InOut		
9	<Hinzufügen>		
10	Static		
11	<Hinzufügen>		
12	Temp		
13	<Hinzufügen>		
14	Constant		
15	LONG_DATA_ZERO	DInt	0
16	BOOL_ALWAYS_TRUE	Bool	true

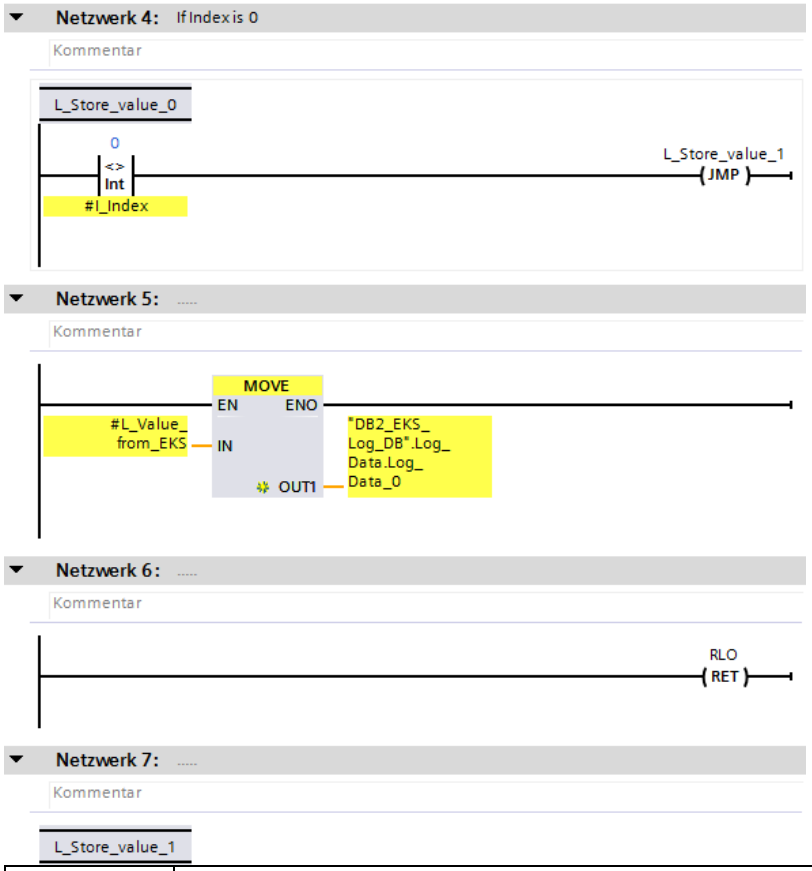
Eingangswerte

Name	Datentyp	Bemerkung
L_Value_from_EKS	DInt	Der Wert, der vom Schlüssel gelesen wurde.
M_Store	Bool	Flag, das anzeigt, das der Wert gespeichert werden soll.
M_Delete	Bool	Flag das anzeigt, das der Wert gelöscht werden soll.
I_Index	Int	Stelle im Speicher, die bearbeitet werden soll.

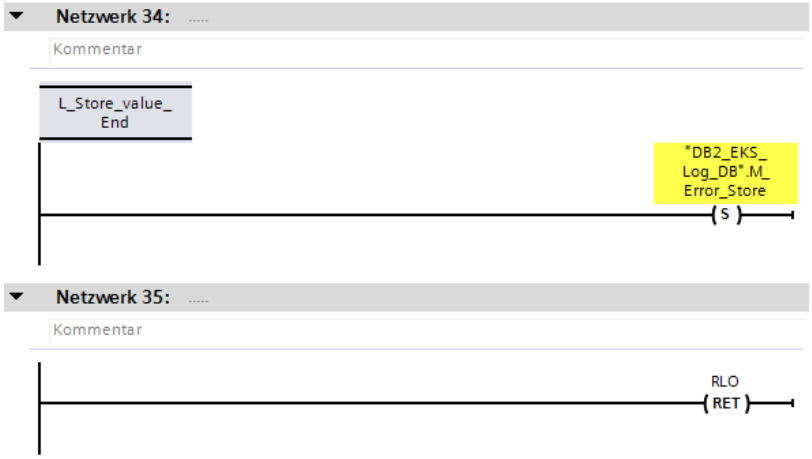


Netzwerk 1	Wenn gespeichert werden soll, wird zum zugehörigen Programmteil gesprungen.

Netzwerk 2	Wenn gelöscht werden soll, wird zum zugehörigen Programmteil gesprungen.
Netzwerk 3	Wenn weder gespeichert, noch gelöscht werden soll (Schlüssel wurde bereits bearbeitet), ist die Routine beendet.

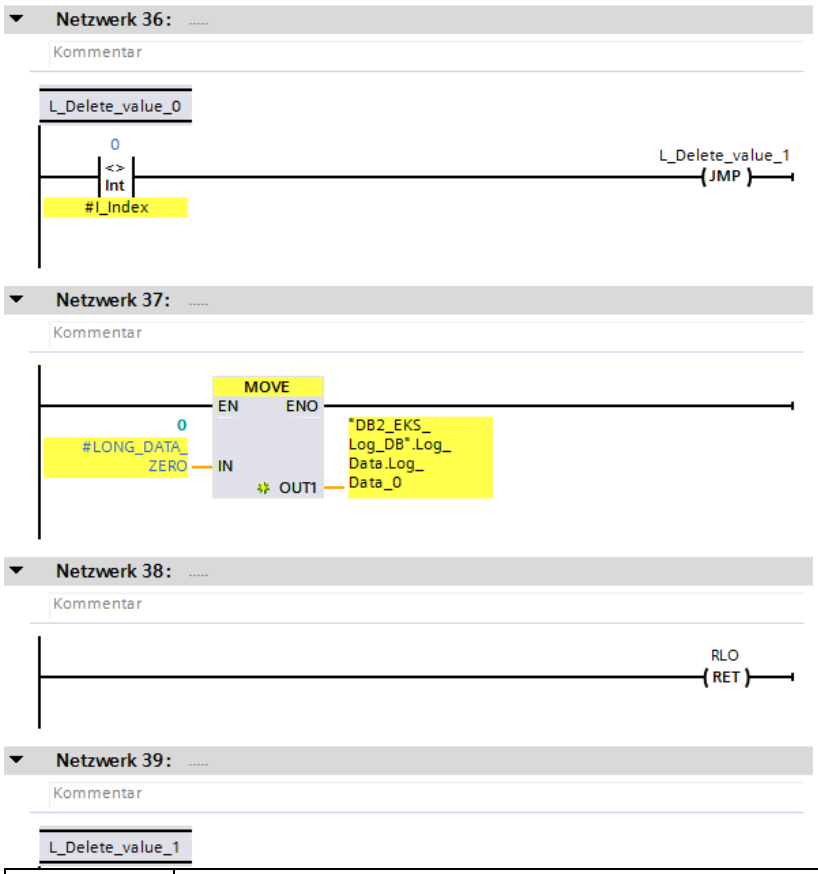


Netzwerk 4	Es wird geprüft, ob an der ersten Stelle im Speicher geschrieben werden soll. Falls nicht, wird die nächste Stelle im Speicher geprüft.
Netzwerk 5	Falls an der ersten Stelle geschrieben werden soll, wird der Wert vom Schlüssel eingespeichert.
Netzwerk 6	Nach dem Speichern ist die Routine beendet.
Netzwerk 7 bis 33	Die 3 Schritte aus dem Netzwerk 4 bis 7 werden für jede Speicherstelle im sicheren Speicher wiederholt. **Hinweis**: Wenn mehr Speicherstellen verwendet werden sollen, müssen für jede neue Speicherstelle diese 3 Schritte einprogrammiert werden.

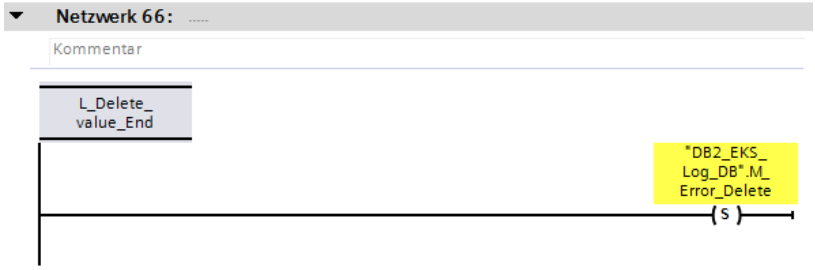


Netzwerk 34	Das Flag, das anzeigt, das gespeichert werden soll, war gesetzt. Es wurde aber der zugehörige Index nicht gefunden. Aus diesem Grund wird ein Fehlerflag gesetzt, das nicht in der Routine zum Löschen von Fehlern rückgesetzt werden kann. Zudem darf dieser Fehler nicht selbst quittierend sein.

Netzwerk 35	Die Routine endet anschließend



Netzwerk 36	Es wird geprüft, ob an der ersten Stelle im Speicher gelöscht werden soll. Falls nicht, wird die nächste Stelle im Speicher geprüft.
Netzwerk 37	Falls an der ersten Stelle gelöscht werden soll, wird eine Null eingespeichert.
Netzwerk 38	Nach dem Löschen ist die Routine beendet.
Netzwerk 39 bis 65	Die 3 Schritte aus dem Netzwerk 36 bis 38 werden für jede Speicherstelle im sicheren Speicher wiederholt. **Hinweis**: Wenn mehr Speicherstellen verwendet werden sollen, müssen für jede neue Speicherstelle diese 3 Schritte einprogrammiert werden.



Netzwerk 66	Das Flag zum Löschen war gesetzt, jedoch wurde der zugehörige Index nicht gefunden. Aus diesem Grund wird ein Fehlerflag gesetzt, das nicht in der Routine zum Löschen von Fehlern rückgesetzt werden kann. Zudem darf dieser Fehler nicht selbst quittierend sein.

3 Prinzipielles Schaltbild

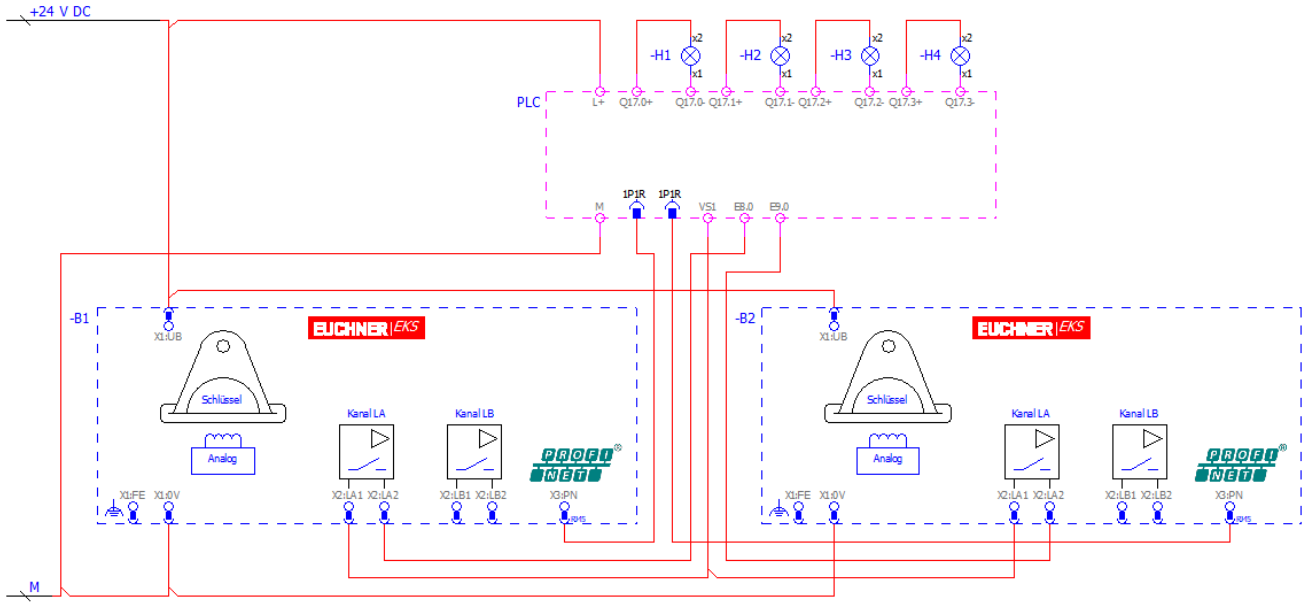


Bild 1

4 Sicherheitstechnische Beschreibung

Das beschriebene System erfüllt die Anforderungen für PL d nach EN ISO 13849-1 bei Verwendung der Baugruppen entsprechend Abschnitt „Verwendete Bauteile / Module“. Zur Berechnung der Ausfallwahrscheinlichkeit PFHD nach EN ISO 13849-1 wird das sicherheitstechnische Diagramm entsprechend Bild 2 verwendet.

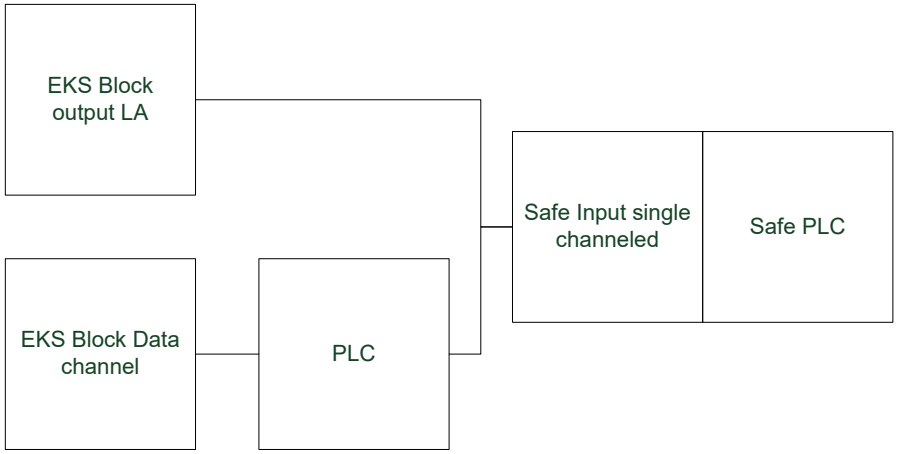


Bild 2

Verwendete PFHD Werte:

Block	Sicherheitstechnische Werte	Quelle
EKS Block output LA	MTTFD 416 Jahre, DC 92%	Bibliothek SISTEMA-V2-EUCHNER-v2.13-DE-PUB.slb
EKS Block Data channel	MTTFD 416 Jahre, DC 92%	Bibliothek SISTEMA-V2-EUCHNER-v2.13-DE-PUB.slb
PLC	MTTFD 10 Jahre, DC 60%	MTTFD Wert nach EN ISO 13849-1, DC Wert ermittelt gilt bei Verwendung der beschriebenen Software in dieser Applikation
Safe input single channeled	PFHD 1,0 E-08, PL d	VDMA Bibliothek Siemens_SafetyIntegrated_1.15.xml
Safe PLC	PFHD 2,5 E-09, PL e	VDMA Bibliothek Siemens_SafetyIntegrated_1.15.xml

Wichtig: Es müssen ausgangsseitig alle sicherheitsrelevanten Teile ergänzt werden. Die Werte müssen für die in Ihrer Applikation verwendeten Baugruppen eingesetzt werden.

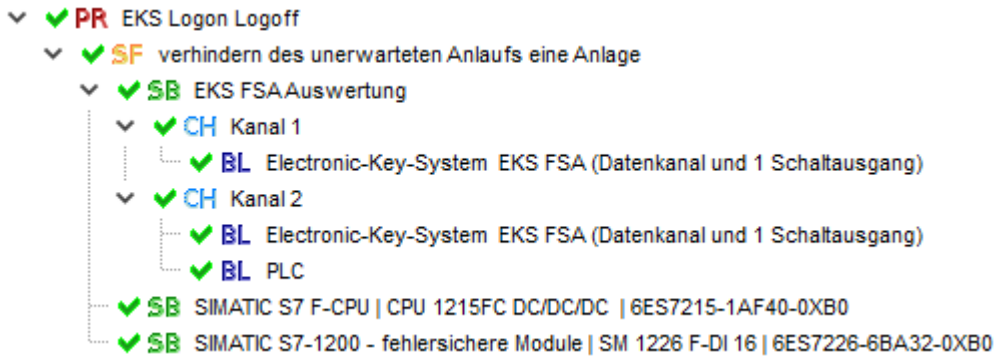


Bild 3

Bei Verwendung obiger Werte ergibt sich für das gesamte System eine PFHD von 1,9 E-07 und somit PL d.

4.1 Software

Die Software in der F-SPS ist sicherheitsrelevant. Zur Erstellung und Beurteilung der Software in der F-SPS müssen die Methoden und Maßnahmen, die im Abschnitt 4.6.3 der EN ISO 13849-1:2016 für SRASW beschrieben sind, herangezogen werden. Die Software muss entsprechend Abschnitt 9.5 der EN ISO 13849-2:2013 validiert werden.
Die Erstellung der Software in SPS und HMI muss dem Abschnitt 4.6.4 der EN ISO 13849-1:2016 entsprechen. Die in dieser Applikation vorgestellte Methodik erfüllt diese Anforderungen, jedoch muss auch die Programmierung dementsprechend umgesetzt werden. Die Software muss nach Abschnitt 4.6.4 validiert werden.

5 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 offenbleiben, 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.