



Einbindung EKS mit TCP/IP Schnittstelle in Rockwell Studio 5000®

Inhalt

1.	Zu diesem Dokument	3
1.1.	Version	3
1.2.	Gültigkeit	3
1.3.	Zielgruppe	3
1.4.	Ergänzende Dokumente	3
1.5.	Hinweis	3
2.	Verwendete Bauteile / Module	4
2.1.	EUCHNER	4
2.2.	Andere	4
2.3.	Software	4
3.	Funktionsbeschreibung	4
4.	Importieren der Add-On Instruction	5
5.	Einbinden der Add-On Instruction	7
5.1.	Variablentabelle der Add-On Instruction	7
5.2.	Erstellen der Controller Tags	8
5.3.	Konfiguration der Messages	12
5.4.	Socketverbindung aktivieren	16
5.5.	Kommando zum Schlüssel beschreiben	17
5.6.	Parametrieren der IP-Adresse	18
5.7.	Start des Programms	19
5.8.	Schlüssel beschreiben	20
6.	Wichtiger Hinweis – Bitte unbedingt sorgfältig beachten!	22

1. Zu diesem Dokument

1.1. Version

Version	Datum	Änderung/Erweiterung	Kapitel
01-02/20	10.01.2020	Erstellung	Alle
02-08/20	03.08.2020	Erweiterung Schlüssel schreiben	5.5 / 5.8
03-04/21	16.04.2021	- Änderung des Datentyps der Variable "JobFinishedActiveTime" von UINT zu INT	5.1
		- "WICHTIG!" hinzugefügt: Auswahl Netzwerkkarte	5.3

1.2. Gültigkeit

Dieses Dokument dient zur Einbindung und Projektierung des EKS mit TCP/IP-Schnittstelle im Rockwell Studio 5000 Logix Designer®

1.3. Zielgruppe

Konstrukteure und Anlagenplaner für Sicherheitseinrichtungen an Maschinen, sowie Inbetriebnahme- und Servicefachkräfte, die über spezielle Kenntnisse im Umgang mit Sicherheitsbauteilen sowie über Kenntnisse bei der Installation, Inbetriebnahme, Programmierung und Diagnose von speicherprogrammierbaren Steuerungen (SPS) und Bussystemen verfügen.

1.4. Ergänzende Dokumente

Die Gesamtdokumentation für diese Applikation besteht aus folgenden Dokumenten:

Dokumenttitel (Dokumentnummer)	Inhalt	
Handbuch (2100420)	Electronic-Key-System Handbuch Schlüsselaufnahme EKS und EKS FSA mit Ethernet TCP/IP Schnittstelle	
Ggf. beiliegende Datenblätter	Artikelspezifische Information zu Abweichungen oder Ergänzungen	

1.5. Hinweis

Diese Applikation basiert auf dem Handbuch des EKS mit TCP/IP-Schnittstelle. Die technischen Details sowie weitere Informationen entnehmen Sie bitte dem Handbuch. Im weiteren Verlauf des Dokuments wird das EKS mit TCP/IP-Schnittstelle kurz "EKS" genannt.

2. Verwendete Bauteile / Module

2.1. EUCHNER

Beschreibung	Bestellnummer / Artikel
EKS mit TCP/IP-Schnittstelle	100401 / EKS-A-EX-G01-ST02/03
	099265 / EKS-A-EXA-G01-ST02/03/04



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.

2.2. Andere

Beschreibung	Bestellnummer / Artikel
1756-L81ES GuardLogix® 5580 Safety Controller	1756-L81ES
1756-L8SP GuardLogix® 5580 Safety Partner	1756-L8SP

2.3. Software

Beschreibung	Version
Studio 5000 Logix Designer	Version 32.01.00 - Professional Edition

3. Funktionsbeschreibung

Bei den EKS TCP/IP Geräten handelt es sich um ein Schreib-/Lesesystem mit Elektronik für die induktive bidirektionale Schnittstelle zum Transponder und Schnittstellenelektronik.

Die Systemanbindung erfolgt über die integrierte TCP/IP-Schnittstelle, welche als RJ45-Buchse ausgeführt ist. Zur TCP/IP-Anbindung wird ggf. ein separater Switch benötigt. Das EKS besitzt keinen integrierten Switch.

Der aktuelle Zustand der Schlüsselaufnahme wird über eine 3-farbige LED angezeigt.

Der Schlüssel wird für den Betrieb an der Schlüsselaufnahme platziert. Die Stromversorgung für den Transponder und die Daten werden kontaktlos zwischen Schlüsselaufnahme und Schlüssel übertragen.

Die Datenübertragung zwischen Steuerung und EKS wird mittels einer Add-On Instruction (AOI) realisiert. Die AOI handelt den Aufbau der Kommunikation zwischen Steuerung und EKS sowie das Senden und Empfangen der TCP/IP Kommunikationstelegramme.

Die AOI kann auf www.euchner.de im Bereich *Service/Downloads/Software/Beispieldateien und Bibliotheken/EKS* heruntergeladen werden.

4. Importieren der Add-On Instruction

- Öffnen Sie durch einen Rechtsklick auf Add-On Instructions im *Controller Organizer* das Kontextmenü und wählen Sie *Import Add-On Instruction*.

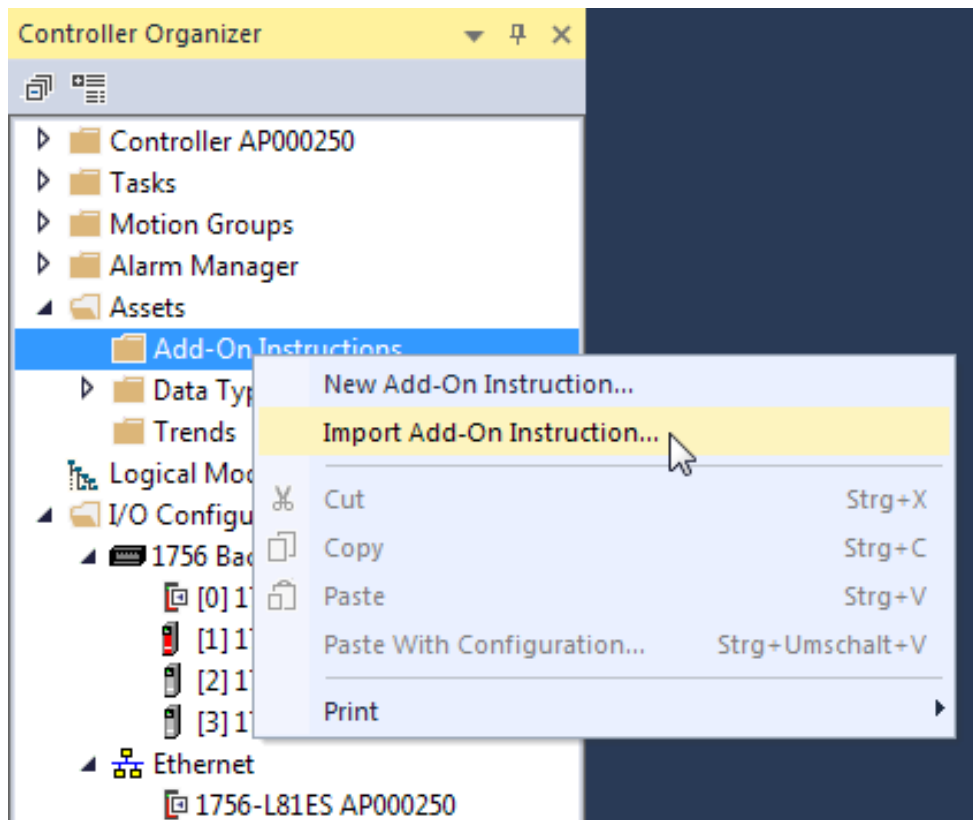


Bild 1: Kontextmenü Add-On Instruction

- Wählen Sie den Ordner mit der heruntergeladenen AOI und öffnen diese mit *Open*.

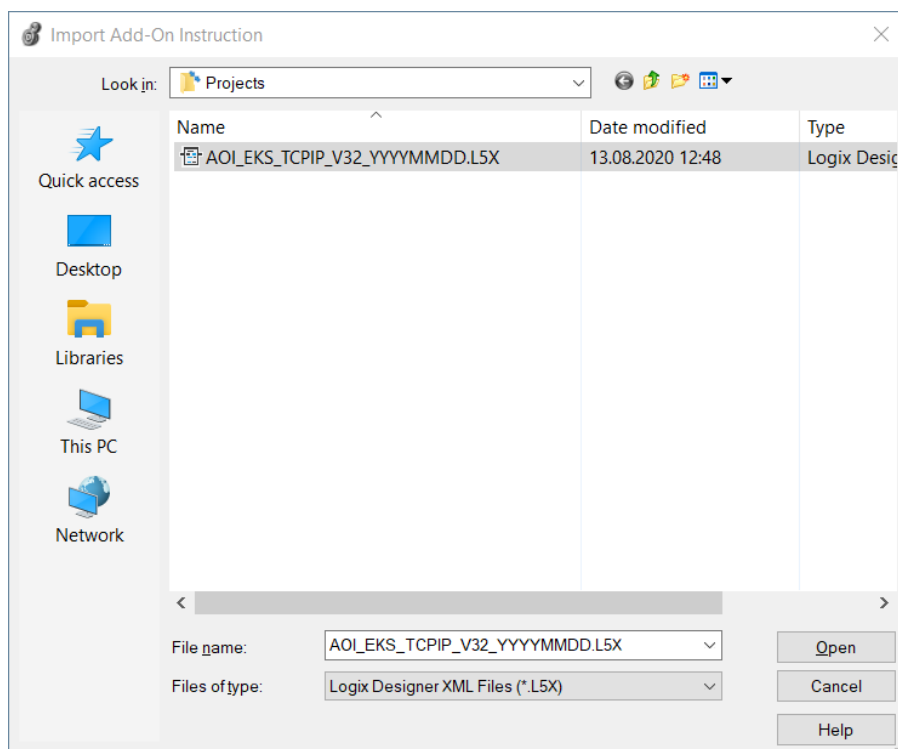


Bild 2: Import AOI

3. Schließen Sie den Import im *Import Configuration* Fenster ohne Änderungen mit *OK* ab.

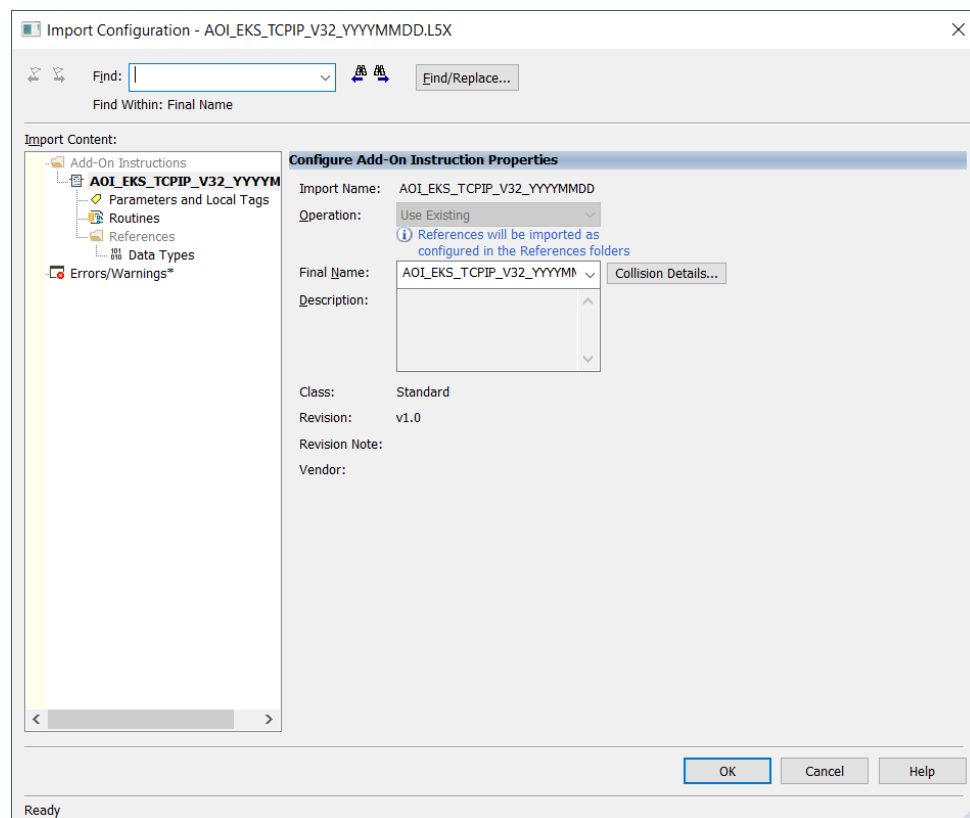


Bild 3: Import Configuration Fenster

Folgende Datentypen und die Add-On Instruction wurden beim Import dem *Controller Organizer* hinzugefügt.

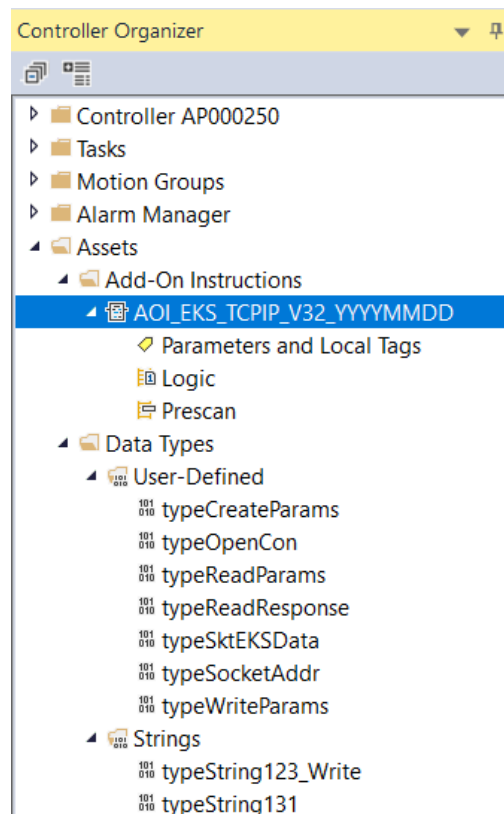


Bild 4: Importierte Datentypen und AOI

5. Einbinden der Add-On Instruction

5.1. Variablentabelle der Add-On Instruction

	Variable	Verwendung	Datentyp	Beschreibung
AOI_EKS_TCPIP_V32_YYYYMMDD	AOI_EKS_TCPIP_V32_YYYYMMDD	-	AOI_EKS_TCPIP_V32_YYYYMMDD	Instanz für die AOI
SkConEnable	SkConEnable	Input	BOOL	Aktiviere TCP/IP Socketverbindung
SkEKSDData	SkEKSDData	InOut	typeSkEKSDData	Die Variable beinhaltet alle benötigten Daten für die Messages
SkTimeout	SkTimeout	Input	DINT	Zeitlimit für die Messages (in ms)
SkCreateMsg	SkCreateMsg	InOut	MESSAGE	Message zum Erstellen des TCP/IP Socket
SkConnectMsg	SkConnectMsg	InOut	MESSAGE	Message zum Aufbau der TCP/IP Socketverbindung
SkReadMsg	SkReadMsg	InOut	MESSAGE	Message zum Lesen des TCP/IP Telegramminhalts
SkWriteMsg	SkWriteMsg	InOut	MESSAGE	Message zum Schreiben des TCP/IP Telegramms
SkDelMsg	SkDelMsg	InOut	MESSAGE	Message zum Löschen der TCP/IP Socketverbindung
EKSReadMode	EKSReadMode	Input	SINT	Modus der Abfrage 1= manuell; 2= automatisch
JobFinishedActiveTime	JobFinishedActiveTime	Input	INT	Zeitwert (in ms) wie lange das Bit <i>JobFinished</i> nach dem Schreibvorgang aktiv bleiben soll
EKSStartAdressRead	EKSStartAdressRead	Input	SINT	Startadresse der anzufragenden Schlüsseldaten
EKSNumberOfBytesRead	EKSNumberOfBytesRead	Input	SINT	Anzahl der anzufragenden Schlüsseldaten
EKSKeyDataRead	EKSKeyDataRead	InOut	SINT[124]	Antwort der Nutzdaten des EKS Schlüssels
EKSStartAddressWrite	EKSStartAddressWrite	Input	SINT	Startadresse der zu schreibenden Schlüsseldaten
EKSNumberOfBytesWrite	EKSNumberOfBytesWrite	Input	SINT	Anzahl der zu schreibenden Schlüsseldaten
EKSKeyDataWrite	EKSKeyDataWrite	InOut	SINT[116]	Zu schreibende Schlüsseldaten
EKSStatusNumber	EKSStatusNumber	Output	INT	EKS Status
SkConToServ	SkConToServ	Output	BOOL	Es wird versucht eine Serververbindung herzustellen
SkUsed	SkUsed	Output	BOOL	TRUE = der Ablauf der Socketverbindung ist gestartet; FALSE = der Ablauf der Socketverbindung ist gestoppt
SkError	SkError	Output	BOOL	Fehler Socketverbindung
SkOpen	SkOpen	Output	BOOL	Socketverbindung ist geöffnet
SkRead	SkRead	Output	BOOL	Telegramm wird empfangen
SkWrite	SkWrite	Output	BOOL	Telegramm wird gesendet
EKSKeyIN	EKSKeyIN	Output	BOOL	EKS Schlüssel in der Schlüsselaufnahme platziert
EKSStatusMessage	EKSStatusMessage	Output	BOOL	Eine EKS Statusmeldung wurde empfangen
JobFinished	JobFinished	Output	BOOL	Schreibvorgang abgeschlossen

Tabelle 1: Variablentabelle AOI

5.2. Erstellen der Controller Tags

1. Öffnen Sie ein Programm (z.B. *MainRoutine*) und ziehen Sie mit Drag & Drop die AOI in einen neuen Rung.

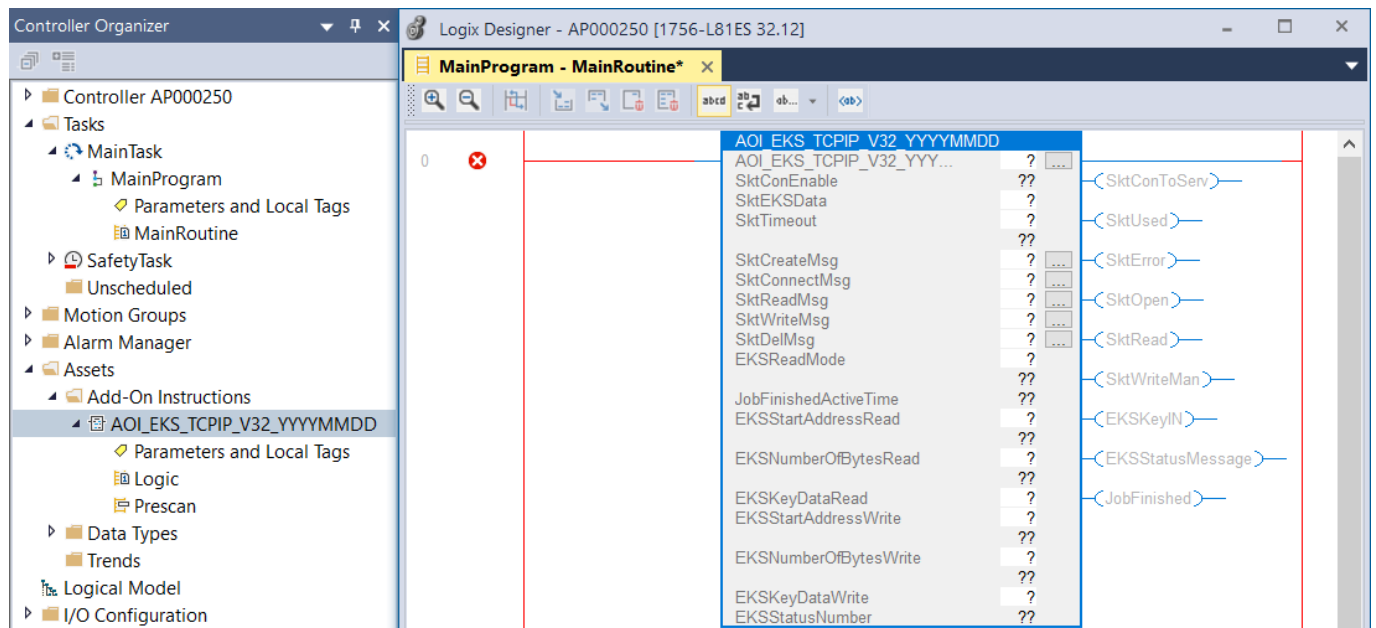


Bild 5: Hinzufügen der AOI im Hauptprogramm

2. Erzeugen der Instanz der AOI. Schreiben Sie den Namen der Instanz in das Feld für die Variable *AOI_EKS_TCPIP_V32_YYYYMMDD* und öffnen Sie mit einem Rechtsklick das Kontextmenü. In diesem Beispiel wählen Sie *New "AOI_EKS_Milling"*.

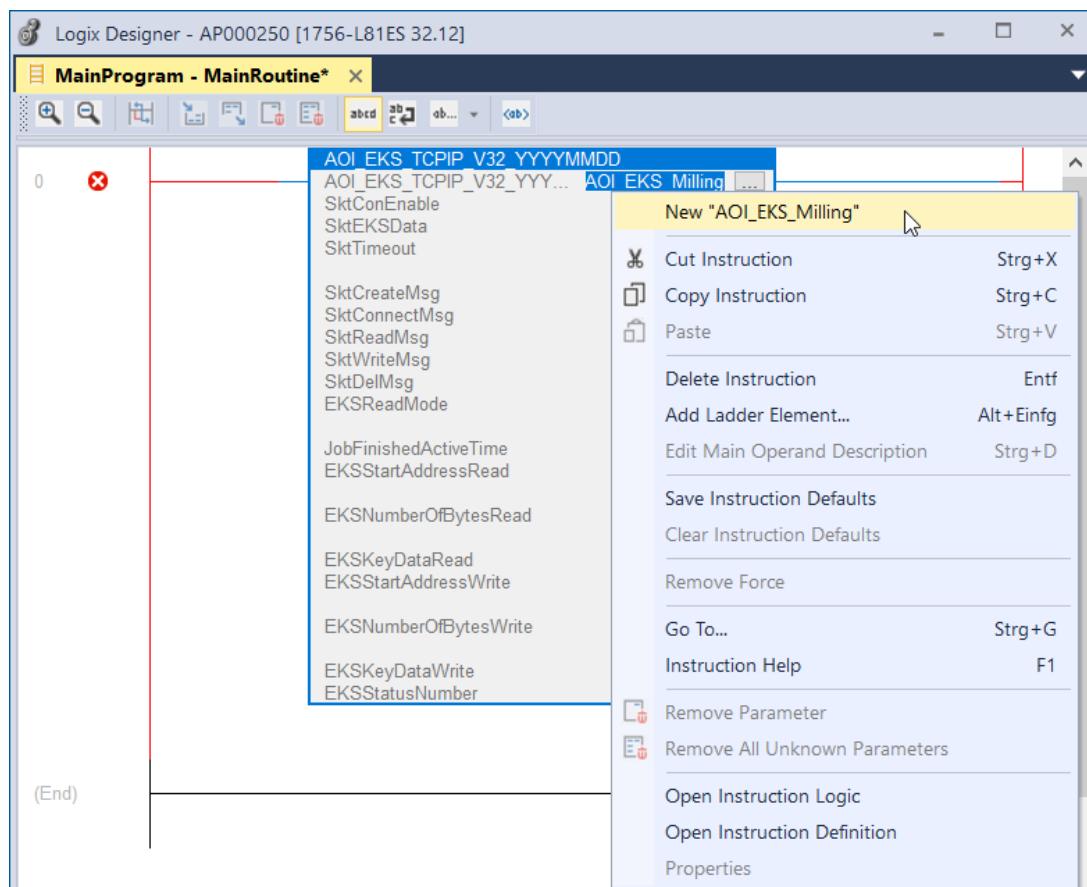


Bild 6: Instanzvariable anlegen

3. Im *New Tag* Fenster sind die notwendigen Informationen bereits vorausgefüllt. Klicken Sie auf *Create* um die Variable zu erstellen.

The 'New Tag' dialog box contains the following fields and values:

- Name:** AOI_EKS_Milling
- Description:** (Empty text area)
- Usage:** <controller>
- Type:** Base
- Alias For:** (Empty)
- Data Type:** AOI_EKS_TCPIP_V32_YYYYMMDD
- Parameter Connection:** (Empty)
- Scope:** AP000250
- Class:** Standard
- External Access:** Read/Write
- Style:** (Empty)
- ☐ Constant
- ☐ Sequencing
- ☐ Open Configuration
- ☐ Open Parameter Connections

Bild 7: Instanzvariable erstellen

4. Erstellen der Variable der EKS Socketverbindung. Schreiben Sie den Variablennamen in das zugehörige Variablenfeld und öffnen Sie mit einem Rechtsklick das Kontextmenü. In diesem Beispiel wählen Sie *New "EKS_Milling"*. Erstellen Sie anschließend die Variable ohne Änderung mit *Create*.

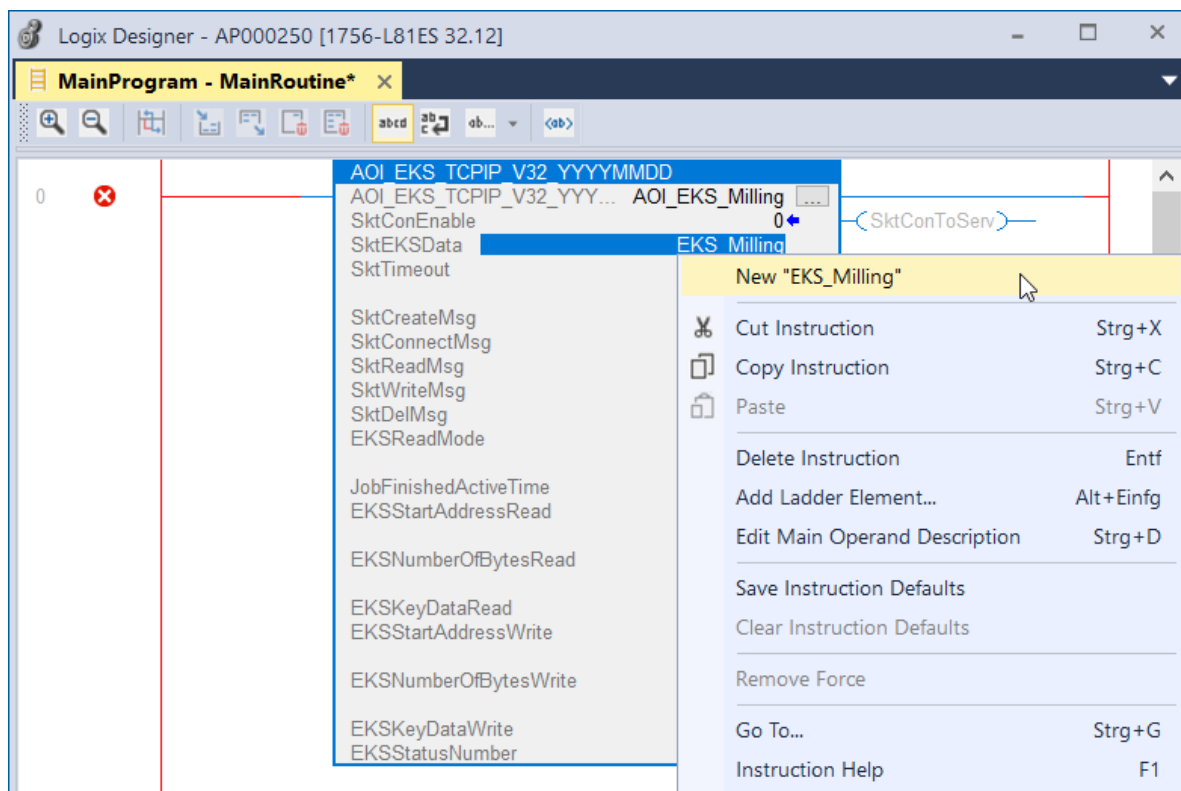


Bild 8: Variable für die EKS Socketdaten erstellen

5. Erstellen der Variable der Socket Create Message. Schreiben Sie den Variablennamen in das zugehörige Variablenfeld und öffnen Sie mit einem Rechtsklick das Kontextmenü. In diesem Beispiel wählen Sie *New "EKS_Skt_Create_Msg"*.

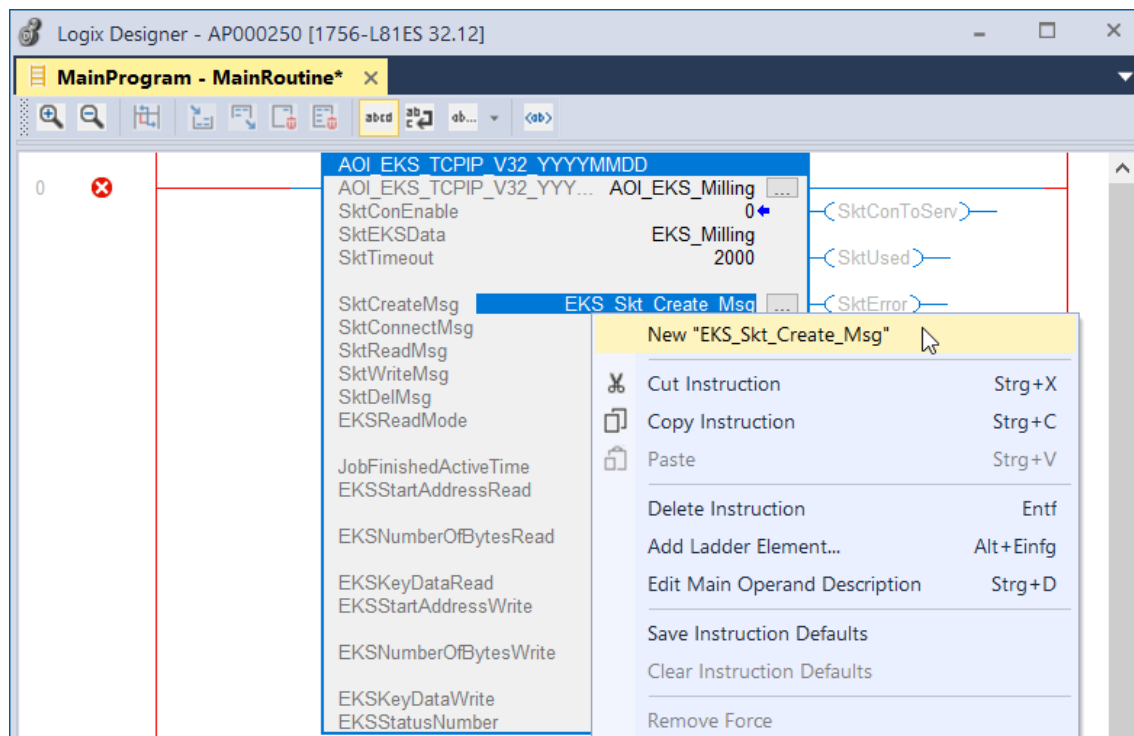


Bild 9: Erstellen der Socket Create Message Variable

6. Erstellen Sie anschließend die Variable ohne Änderung mit *Create*.

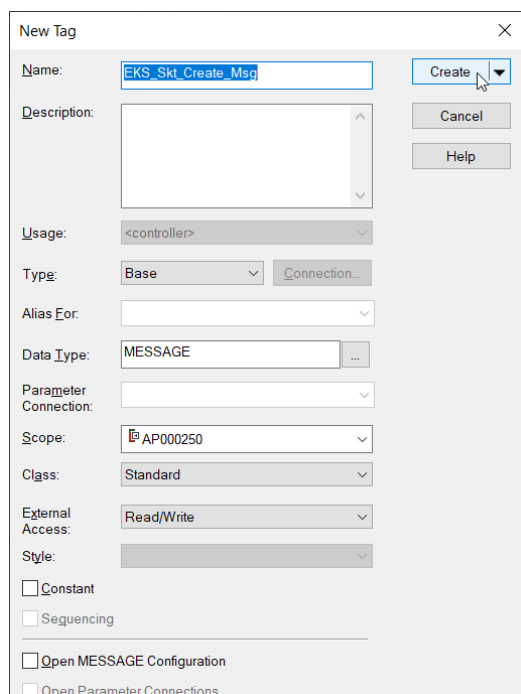


Bild 10: Erstellen der Socket Create Message Variable



WICHTIG

Wiederholen Sie das Erzeugen der Message Variablen (Schritt 5 und 6) für die Variablen *SktConnectMsg*, *SktReadMsg*, *SktWriteMsg* und *SktDelMsg*. Des Weiteren erzeugen Sie die Variablen *EKSKeyDataRead* und *EKSKeyDataWrite*.

7. Eintragen von Werten für die Eingangsvariablen.

Variable	Wert
SktTimeout	2000 (in ms)
EKSReadMode	1 = manueller Modus; Die Schlüsseldaten werden durch Triggern der Variable <i>SktWrite</i> empfangen 2 = Automatischer Modus; Das Empfangen der Schlüsseldaten wird durch Auslesen des Key-Status <i>Key/N</i> (siehe Handbuch) getriggert.
JobFinishedActiveTime	Variabel (Wert von 0 bis 65535) → Zeitwert (in ms) wie lange das Bit <i>JobFinished</i> nach dem Schreibvorgang aktiv bleiben soll
EKSStartAddressRead	Variabel (Wert von 0 bis 116) → Startadresse der verwendeten Nutzdaten
EKSNumberOfBytesRead	Variabel (Wert von 0 bis 124) → Anzahl der zu lesenden Bytes
EKSStartAddressWrite	Variabel (Wert von 0 bis 112) → Startadresse der verwendeten Nutzdaten
EKSNumberOfBytesWrite	Variabel (Wert von 4 bis 116) → Anzahl der zu schreibenden Bytes



WICHTIG

Beim Schreib-/Lese-Schlüssel mit frei programmierbaren 116 Bytes ist der Speicher in 4-Byte-Blöcken organisiert. Dies bedeutet, die Start-Adresse muss beim Schreiben im Bereich Byte Nr. 0 bis Byte Nr. 112, immer in 4-Byte-Schritten, angegeben werden (Byte Nr. 0, 4, 8 ... 112). Außerdem muss immer in einem Vielfachen von 4-Bytes großen Blöcken geschrieben werden (4, 8, 12 ... 116 Bytes)!

Beim Lesen kann allerdings wiederum bytewise auf den Speicher zugegriffen werden, ohne die oben genannte Einschränkung beim Schreiben.

Der Schreib-/Lese-Schlüssel hat zusätzlich eine einmalige 8-Byte große Seriennummer, die bei der Schlüssel-Produktion absolut unzerstörbar in den Speicher geschrieben wird. Die Seriennummer kann daher nicht geändert werden. Diese Seriennummer dient zur sicheren Unterscheidung eines jeden einzelnen Schlüssels. Für eine sichere Unterscheidung ist es erforderlich alle 8 Bytes komplett auszuwerten. Die Seriennummer schließt sich an den frei programmierbaren Speicher an. Die Seriennummer kann unter Eingabe der Start-Adresse Byte Nr. 116 und Anzahl Bytes 8 ausgelesen werden.

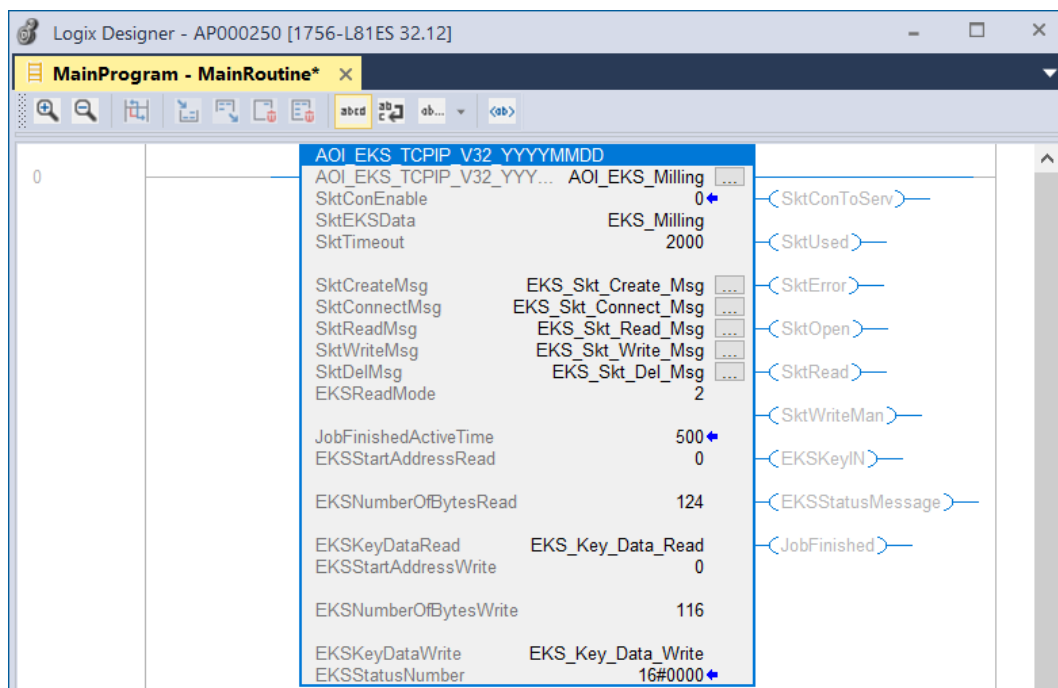


Bild 11: Komplettierte AOI

5.3. Konfiguration der Messages

1. Öffnen Sie den *Configuration Dialog* der *SktCreateMsg*.

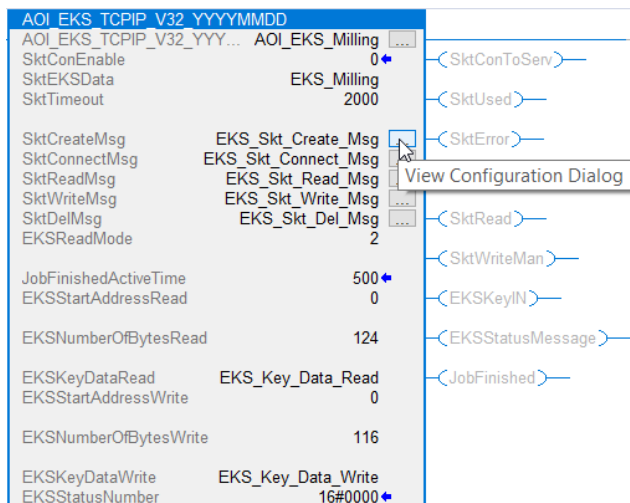


Bild 12: Öffnen des *Configuration Dialog*

2. Wählen Sie den *Service Type*: *Socket Create*. Füllen Sie folgende Parameter aus:

Parameter	Wert
Source Element	EKS_Milling.CreateSrc
Destination Element	EKS_Milling.Inst
Path	THIS (Den Parameter Pfad (Path) finden Sie unter dem Reiter <i>Communication</i>)



WICHTIG

Im Parameter *Path* muss die Netzwerkschnittstelle angegeben werden, worüber das EKS kommunizieren soll. Bei mehreren Netzwerkkarten ist darauf zu achten, dass die richtige Netzwerkkarte im Parameter *Path* ausgewählt ist. Dies betrifft jede *Message Konfiguration*.

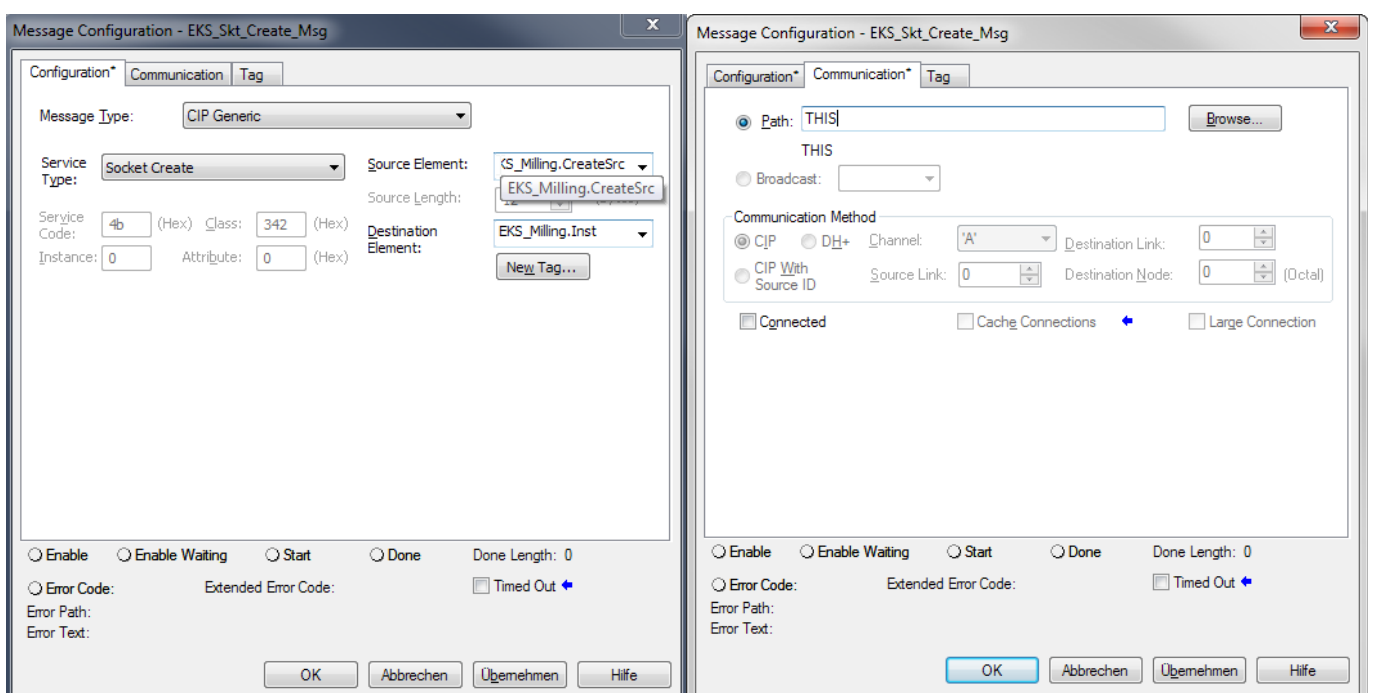


Bild 13: Parametrierung *Socket Create Message*

3. Öffnen Sie den *Configuration Dialog* der *SktConnectMsg*. Wählen Sie den *Service Type: OpenConnection*. Füllen Sie folgende Parameter aus:

Parameter	Wert
Source Element	EKS_Milling.OpenSrc
Source Length	1
Instance	0
Path	THIS (Den Parameter <i>Pfad (Path)</i> finden Sie unter dem Reiter <i>Communication</i>)

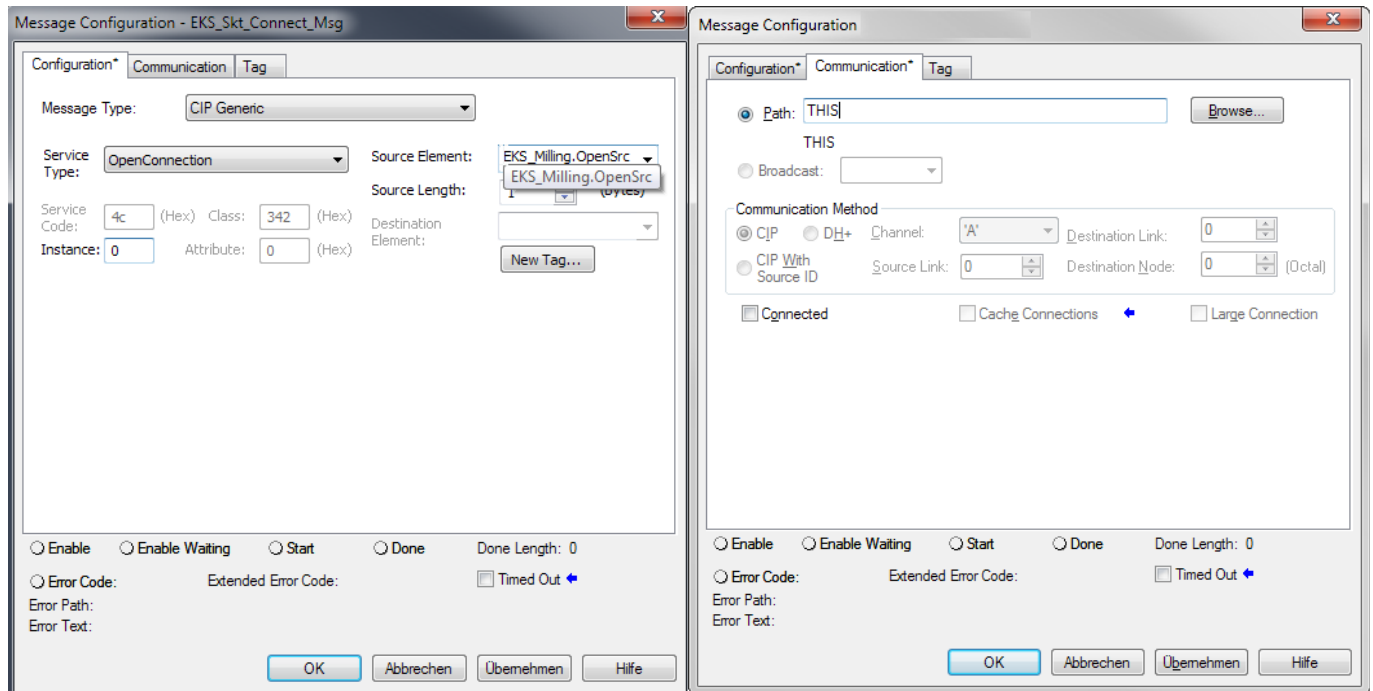


Bild 14: Parametrierung OpenConnection Message

4. Öffnen Sie den *Configuration Dialog* der *SktReadMsg*. Wählen Sie den *Service Type: ReadSocket*. Füllen Sie folgende Parameter aus:

Parameter	Wert
Source Element	EKS_Milling.ReadSrc
Destination Element	EKS_Milling.ReadResponse
Instance	0
Path	THIS (Den Parameter <i>Pfad (Path)</i> finden Sie unter dem Reiter <i>Communication</i>).

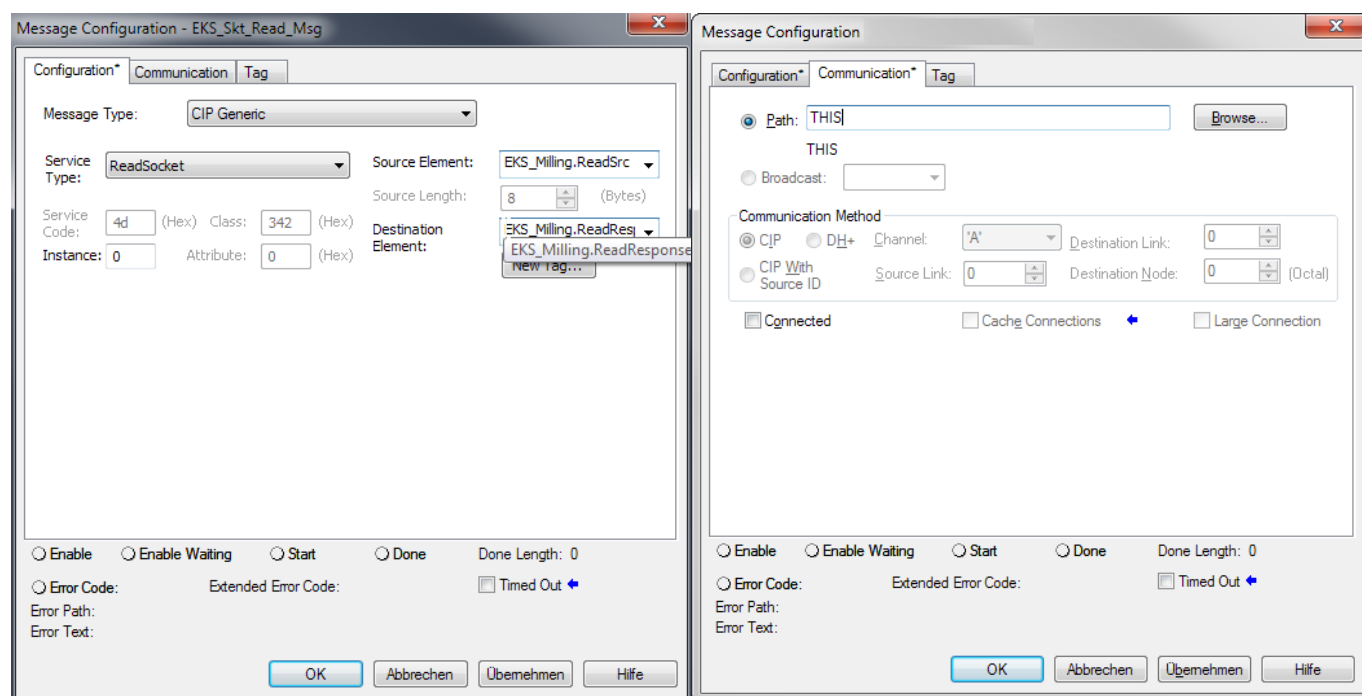


Bild 15: Parametrierung ReadSocket Message

5. Öffnen Sie den *Configuration Dialog* der *SktWriteMsg*. Wählen Sie den *Service Type: WiteSocket*. Füllen Sie folgende Parameter aus:

Parameter	Wert
Source Element	EKS_Milling.WriteSrc
Destination Element	EKS_Milling.WriteSizeSent
Source Length	1
Instance	0
Path	THIS (Den Parameter Pfad (Path) finden Sie unter dem Reiter <i>Communication</i>)

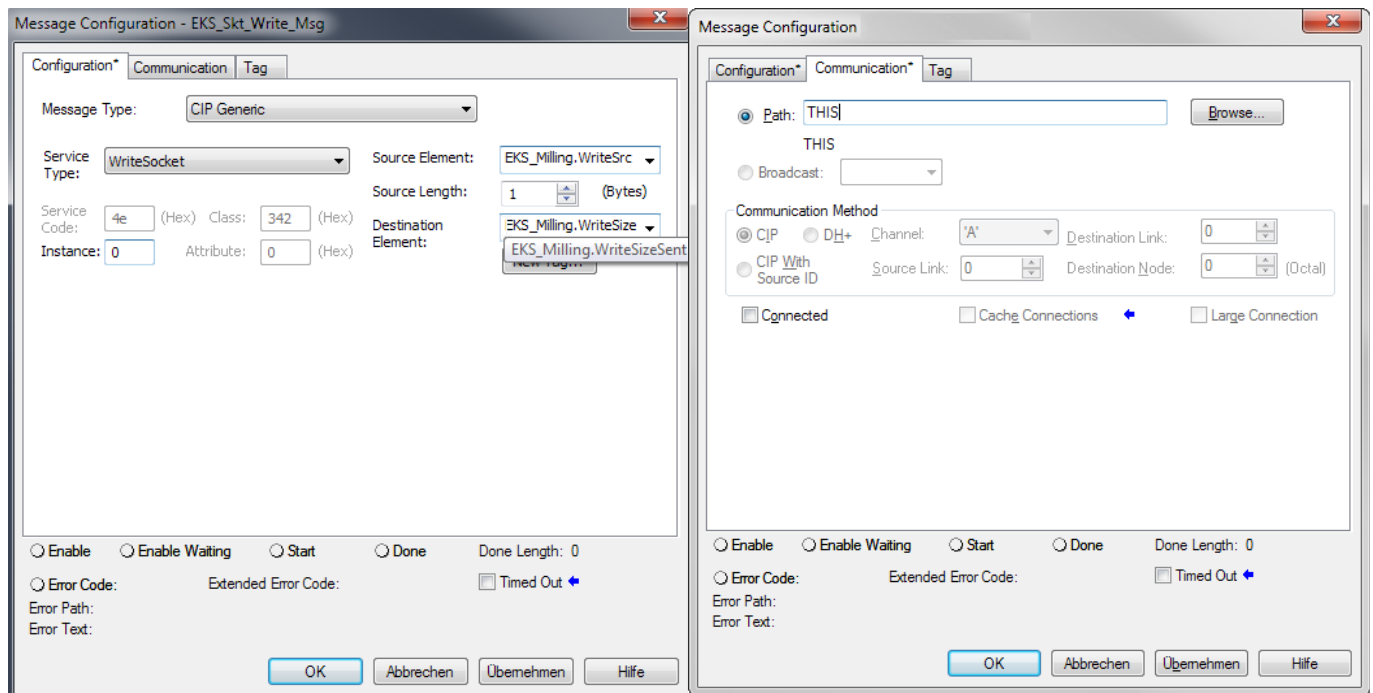


Bild 16: Parametrierung WriteSocket Message

6. Öffnen Sie den *Configuration Dialog* der *SktDelMsg*. Wählen Sie den *Service Type: DeleteSocket*. Füllen Sie folgende Parameter aus:

Parameter	Wert
Instance	0
Path	THIS (Den Parameter Pfad (Path) finden Sie unter dem Reiter <i>Communication</i>)

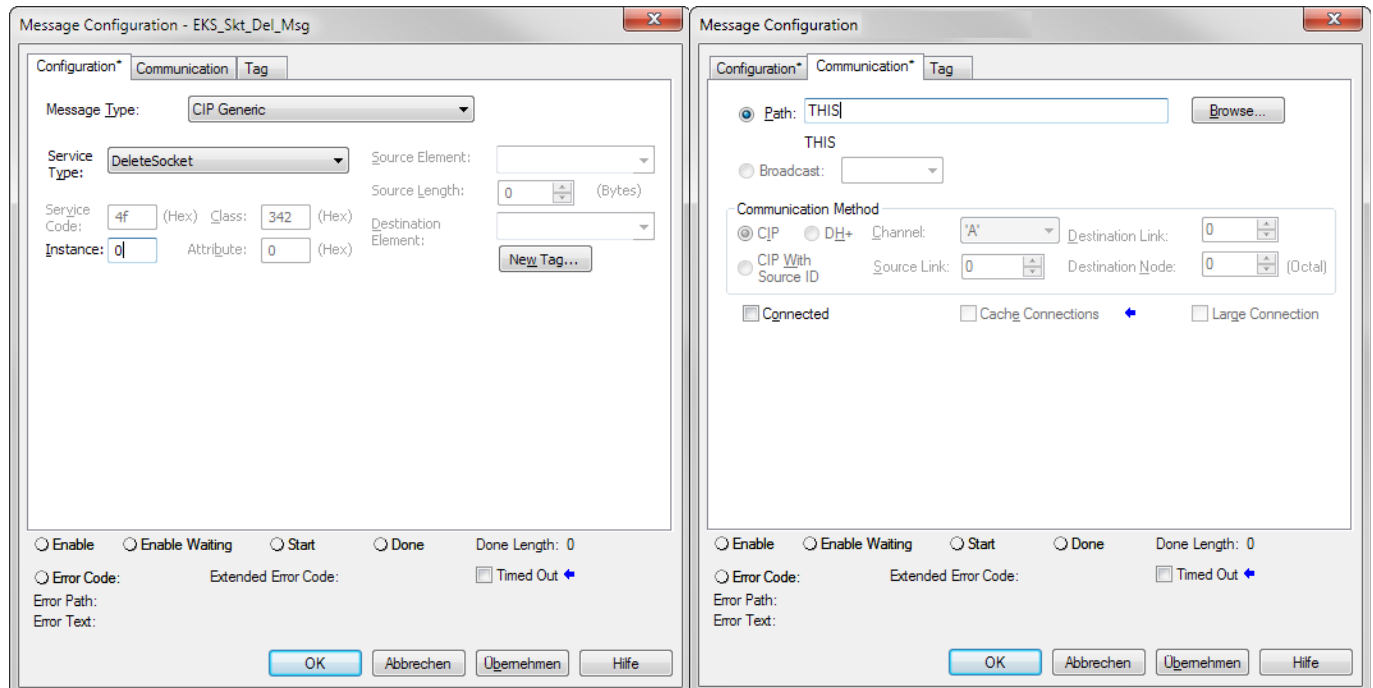


Bild 17: Parametrierung *DeleteSocket* Message

5.4. Socketverbindung aktivieren

Erstellen Sie einen neuen Rung. Fügen Sie dem Rung eine Eingangsvariable (z.B. *EnableSocket*) als Kontakt (*Examine On*) hinzu. Fügen Sie als Ausgang (*Output Energize*) die Variable *SktConEnable* aus der AOI hinzu.

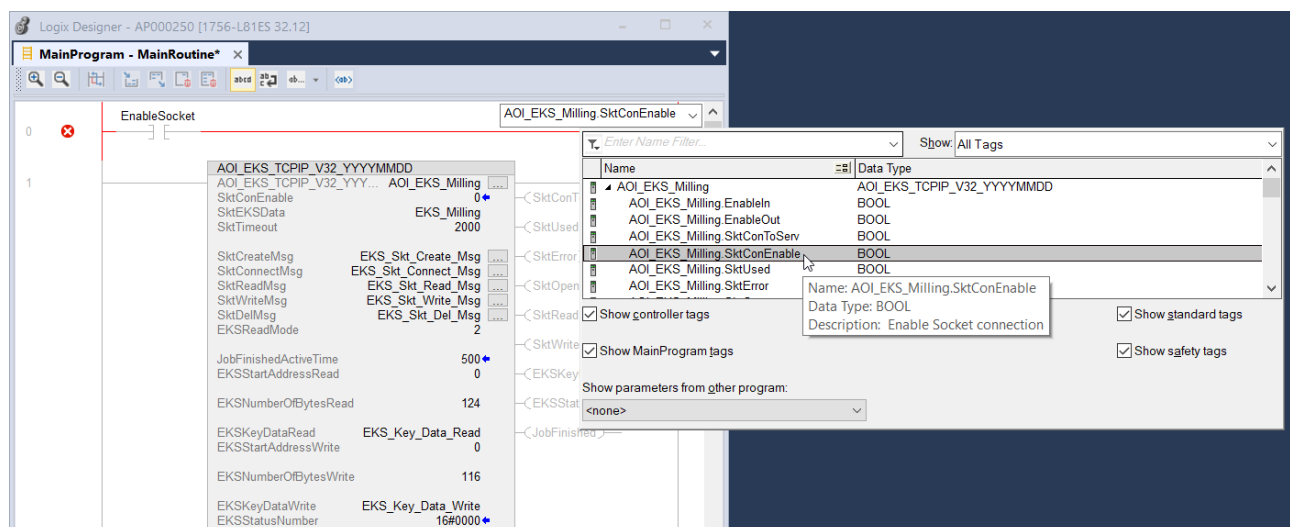


Bild 18: Rung zur Aktivierung der Socketverbindung

5.5. Kommando zum Schlüssel beschreiben

Erstellen Sie einen weiteren Rung und fügen Sie eine Eingangsvariable (z.B. WriteKey) als Kontakt hinzu. Fügen Sie als Ausgang die Variable *EKSWriteKeyCommand* aus der AOI hinzu.

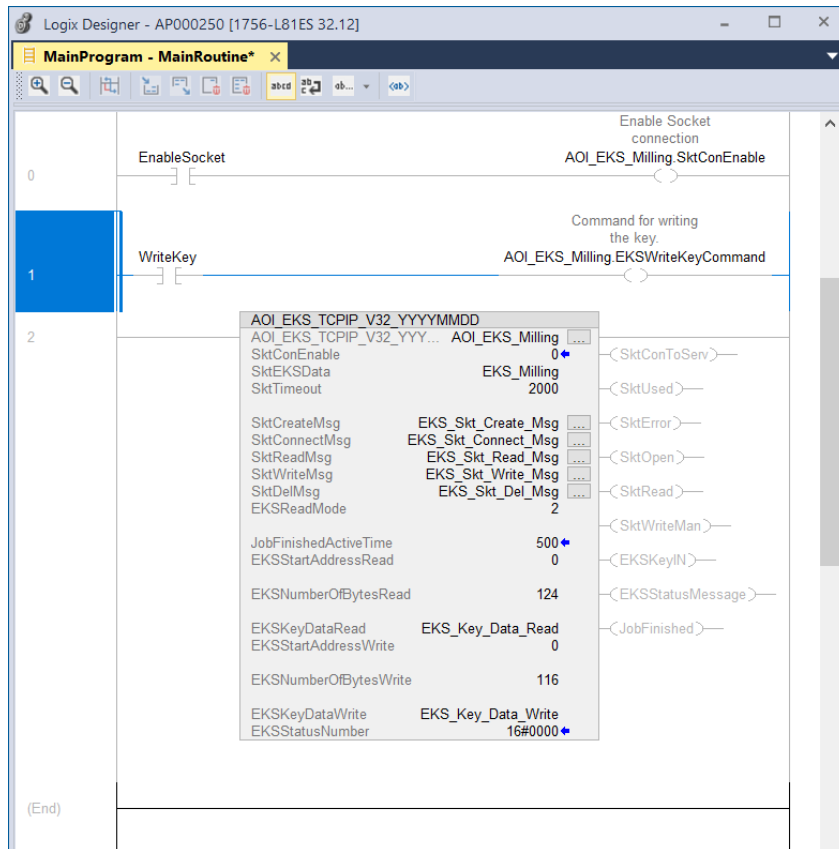


Bild 19: Rung zum Schlüssel beschreiben

5.6. Parametrieren der IP-Adresse



TIPP

Die Vergabe der IP-Adresse der Schlüsselaufnahme EKS wird mittels des Web-Interfaces durchgeführt. Die Beschreibung hierzu finden Sie im Handbuch, Kapitel 7.2

Öffnen Sie die *Controller Tags* und ändern Sie die Ansicht auf *Monitor Tags*. Wählen Sie die Instanz *EKS_Milling* und tragen die IP-Adresse als String ein.

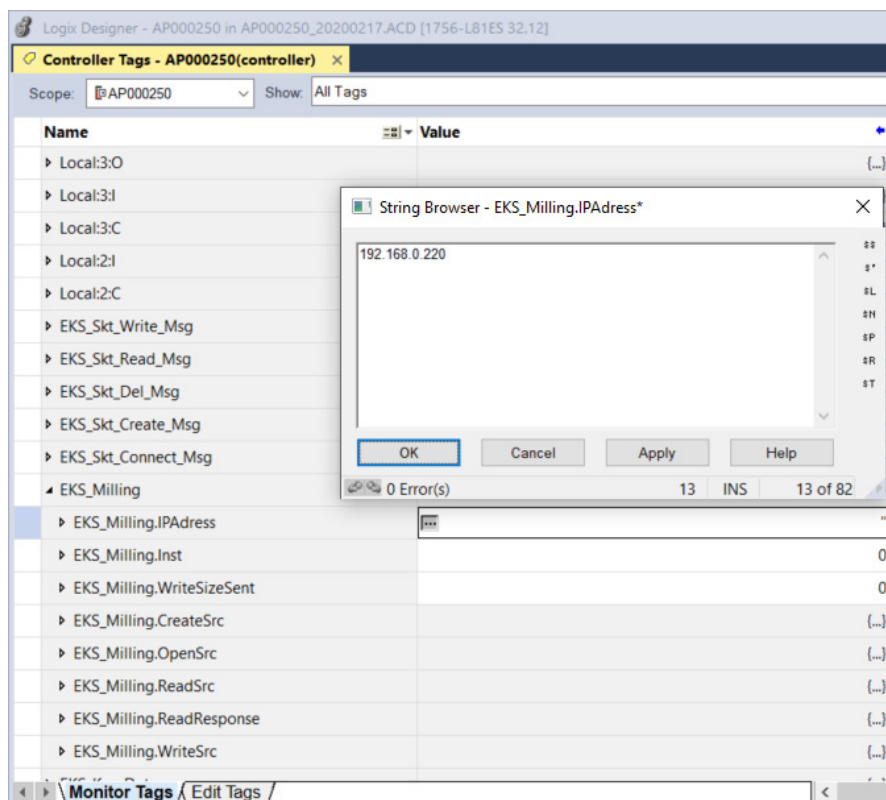


Bild 20: Eintragen der IP-Adresse

5.7. Start des Programms

Laden Sie das Programm in die Steuerung. Gehen Sie *Online* und steuern Sie das Bit *EnableSocket* (=TRUE). Die Socketverbindung ist geöffnet, wenn das Ausgangsbit *SktOpen* der AOI TRUE ist. Mit dem Platzieren eines Schlüssels in der Schlüsselaufnahme wird das Ausgangsbit *EKSKeyIN* = TRUE und die gelesenen Daten werden im *EKS_Key_Data_Read* Array abgelegt.

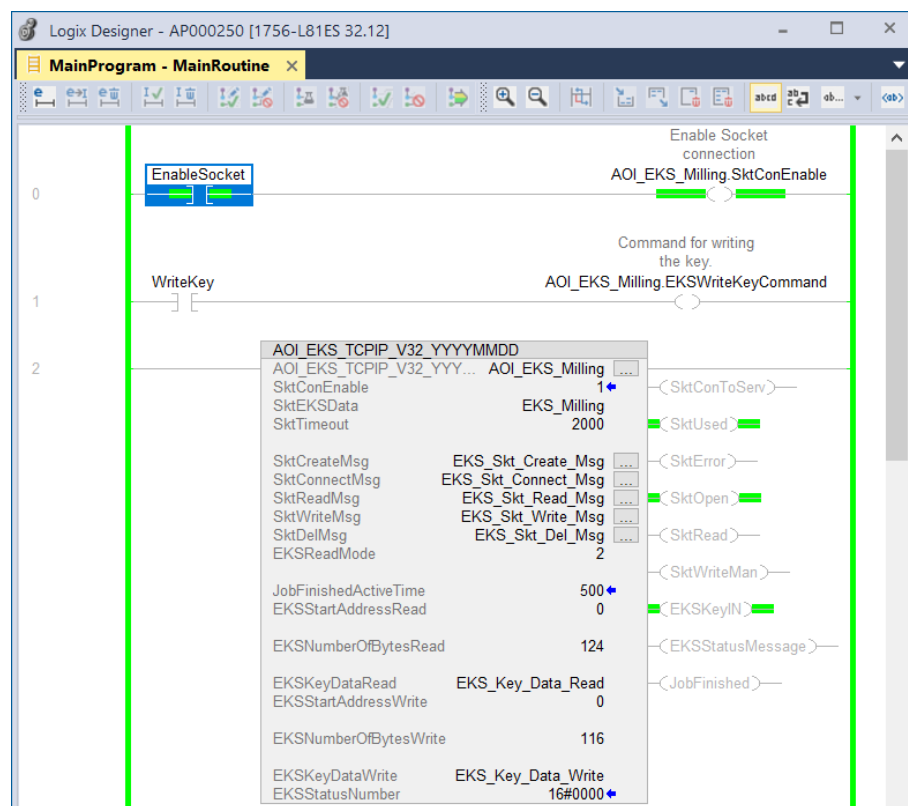


Bild 21: Socketverbindung hergestellt.

Controller Tags - AP000250(controller)					
Scope: AP000250 Show: All Tags					
Name	Value	Force Mask	Style	Data Type	
▲ EKS_Key_Data_Read	(...)	(...)	ASCII	SINT[124]	
▶ EKS_Key_Data_Read[0]	'E'		ASCII	SINT	
▶ EKS_Key_Data_Read[1]	'U'		ASCII	SINT	
▶ EKS_Key_Data_Read[2]	'C'		ASCII	SINT	
▶ EKS_Key_Data_Read[3]	'H'		ASCII	SINT	
▶ EKS_Key_Data_Read[4]	'N'		ASCII	SINT	
▶ EKS_Key_Data_Read[5]	'E'		ASCII	SINT	
▶ EKS_Key_Data_Read[6]	'R'		ASCII	SINT	
▶ EKS_Key_Data_Read[7]	''		ASCII	SINT	
▶ EKS_Key_Data_Read[8]	'L'		ASCII	SINT	
▶ EKS_Key_Data_Read[9]	'E'		ASCII	SINT	
▶ EKS_Key_Data_Read[10]	'V'		ASCII	SINT	
▶ EKS_Key_Data_Read[11]	'E'		ASCII	SINT	
▶ EKS_Key_Data_Read[12]	'L'		ASCII	SINT	
▶ EKS_Key_Data_Read[13]	''		ASCII	SINT	
▶ EKS_Key_Data_Read[14]	'3'		ASCII	SINT	
▶ EKS_Key_Data_Read[15]	'\$00'		ASCII	SINT	
▶ EKS_Key_Data_Read[16]	'\$00'		ASCII	SINT	
▶ EKS_Key_Data_Read[17]	'\$00'		ASCII	SINT	
▶ EKS_Key_Data_Read[18]	'\$00'		ASCII	SINT	
▶ EKS_Key_Data_Read[19]	'\$00'		ASCII	SINT	
▶ EKS_Key_Data_Read[20]	'\$00'		ASCII	SINT	

Bild 22: Ergebnis im EKS_Key_Data_Read Array

5.8. Schlüssel beschreiben

Um einen Schlüssel zu beschreiben, muss das Array *EKS_Key_Data_Write* mit den zu schreibenden Daten gefüllt werden. Da der Schlüssel immer in 4-Byte-Blöcken beschrieben wird, werden alle Bytes in denen Sie keine Werte eingetragen haben, mit 0 überschrieben. Sollten Sie einzelne Daten auf den Schlüssel ändern wollen, müssen immer die Daten, die nicht geändert werden sollen, auch in das Array geschrieben werden. In diesem Beispiel ändern wir die Stufe 3 auf Stufe 5.

Controller Tags - AP000250(controller)					
Scope: AP000250		Show: All Tags			
Name	Value	Force Mask	Style	Data Type	
▲ EKS_Key_Data_Write	{...}	{...}	ASCII	SINT[116]	
▶ EKS_Key_Data_Write[0]	'E'		ASCII	SINT	
▶ EKS_Key_Data_Write[1]	'U'		ASCII	SINT	
▶ EKS_Key_Data_Write[2]	'C'		ASCII	SINT	
▶ EKS_Key_Data_Write[3]	'H'		ASCII	SINT	
▶ EKS_Key_Data_Write[4]	'N'		ASCII	SINT	
▶ EKS_Key_Data_Write[5]	'E'		ASCII	SINT	
▶ EKS_Key_Data_Write[6]	'R'		ASCII	SINT	
▶ EKS_Key_Data_Write[7]	''		ASCII	SINT	
▶ EKS_Key_Data_Write[8]	'L'		ASCII	SINT	
▶ EKS_Key_Data_Write[9]	'E'		ASCII	SINT	
▶ EKS_Key_Data_Write[10]	'V'		ASCII	SINT	
▶ EKS_Key_Data_Write[11]	'E'		ASCII	SINT	
▶ EKS_Key_Data_Write[12]	'L'		ASCII	SINT	
▶ EKS_Key_Data_Write[13]	''		ASCII	SINT	
▶ EKS_Key_Data_Write[14]	☒ '5'		ASCII	SINT	
▶ EKS_Key_Data_Write[15]	'\$00'		ASCII	SINT	
▶ EKS_Key_Data_Write[16]	'\$00'		ASCII	SINT	
▶ EKS_Key_Data_Write[17]	'\$00'		ASCII	SINT	
▶ EKS_Key_Data_Write[18]	'\$00'		ASCII	SINT	
▶ EKS_Key_Data_Write[19]	'\$00'		ASCII	SINT	
▶ EKS_Key_Data_Write[20]	'\$00'		ASCII	SINT	

Bild 23: Füllen des Arrays *EKS_Key_Data_Write*

Wenn Sie die Daten aus dem Array nun auf den Schlüssel schreiben wollen, benötigt das Bit *WriteKey* eine Flanke. Die Daten werden auf den Schlüssel geschrieben und direkt im Anschluss im Array *EKS_Key_Data_Read* angezeigt.



Bild 24: Daten auf den Schlüssel schreiben

Controller Tags - AP000250(controller)					
Scope: AP000250		Show: All Tags			
Name	Value	Force Mask	Style	Data Type	
▲ EKS_Key_Data_Read	{...}		{...} ASCII	SINT[124]	
▶ EKS_Key_Data_Read[0]	'E'		ASCII	SINT	
▶ EKS_Key_Data_Read[1]	'U'		ASCII	SINT	
▶ EKS_Key_Data_Read[2]	'C'		ASCII	SINT	
▶ EKS_Key_Data_Read[3]	'H'		ASCII	SINT	
▶ EKS_Key_Data_Read[4]	'N'		ASCII	SINT	
▶ EKS_Key_Data_Read[5]	'E'		ASCII	SINT	
▶ EKS_Key_Data_Read[6]	'R'		ASCII	SINT	
▶ EKS_Key_Data_Read[7]	''		ASCII	SINT	
▶ EKS_Key_Data_Read[8]	'L'		ASCII	SINT	
▶ EKS_Key_Data_Read[9]	'E'		ASCII	SINT	
▶ EKS_Key_Data_Read[10]	'V'		ASCII	SINT	
▶ EKS_Key_Data_Read[11]	'E'		ASCII	SINT	
▶ EKS_Key_Data_Read[12]	'L'		ASCII	SINT	
▶ EKS_Key_Data_Read[13]	''		ASCII	SINT	
▶ EKS_Key_Data_Read[14]	☒ 5		ASCII	SINT	
▶ EKS_Key_Data_Read[15]	'\$00'		ASCII	SINT	
▶ EKS_Key_Data_Read[16]	'\$00'		ASCII	SINT	
▶ EKS_Key_Data_Read[17]	'\$00'		ASCII	SINT	
▶ EKS_Key_Data_Read[18]	'\$00'		ASCII	SINT	
▶ EKS_Key_Data_Read[19]	'\$00'		ASCII	SINT	
▶ EKS_Key_Data_Read[20]	'\$00'		ASCII	SINT	

Bild 25: Neue Schlüsseldaten

6. Wichtiger Hinweis – Bitte unbedingt sorgfältig beachten!

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

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

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

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

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

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

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

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

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

Verwendung von Marken- und Firmennamen

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

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

Ausgabe:
AP000250-03-04/21
Titel:
Applikation EKS
Einbindung EKS mit TCP/IP Schnittstelle in Rockwell Studio
5000®

Copyright:
© EUCHNER GmbH + Co. KG, 04/2021

Technische Änderungen vorbehalten,
alle Angaben ohne Gewähr.