

Applikation



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

Inhalt

| 1. | Zu d | iesem Dokument | . 3 |
|----|-------|--|-----|
| | 1.1. | Version | 3 |
| | 1.2. | Gültigkeit | 3 |
| | 1.3. | Zielgruppe | 3 |
| | 1.4. | Ergänzende Dokumente | 3 |
| | 1.5. | Hinweis | 3 |
| 2. | Verw | vendete Bauteile / Module | . 4 |
| | 2.1. | EUCHNER | 4 |
| | 2.2. | Andere | 4 |
| | 2.3. | Software | 4 |
| 3. | Funk | tionsbeschreibung | . 4 |
| 4. | Impo | ortieren der Add-On Instruction | . 5 |
| 5. | Einbi | inden 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. | Wich | tiger 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 | 5.1 |
| 0007/21 | 10.04.2021 | - "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 | www |
| Ggf. beiliegende Daten- blätter | 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

 (\mathbf{i})

| Beschreibung | Bestellnummer / Artikel |
|------------------------------|------------------------------------|
| EKS mit TCP/IP-Schnittstelle | 100401 / EKS-A-IEX-G01-ST02/03 |
| | 099265 / EKS-A-IEXA-G01-ST02/03/04 |

TIPP!

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

2.2. Andere

| Beschreibung | Bestellnummer / Artikel |
|---|-------------------------|
| 1756-L81ES GuardLogix® 5580 Safety Controller | 1756L81ES |
| 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 <u>www.euchner.de</u> im Bereich *Service/Downloads/Software/Beispieldateien und Bibliotheken/EKS* heruntergeladen werden.

4. Importieren der Add-On Instruction

1. 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

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



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

| Import Configuration - AOI_EKS_TCPIP_V32_YYYYMMDD.L5X | | | | | | | |
|---|---------------------|---------------------------------|-------------------|-------------|--|--|--|
| Find: Find: Find Name | ~ 🍂 🕅 | Eind/Replace | | | | | |
| Import Content: | | | | | | | |
| Add-On Instructions | onfigure Add-C | In Instruction Properties | | | | | |
| AOI_EKS_TCPIP_V32_YYYYM | import Name: | AOI_EKS_TCPIP_V32_YYYYMMDD | | | | | |
| - Routines | Operation: | Use Existing V | | | | | |
| Mata Types | | configured in the References fo | Iders | | | | |
| Errors/Warnings* | Final <u>N</u> ame: | AOI_EKS_TCPIP_V32_YYYYMN 🗸 | Collision Details | | | | |
| 1 | Description: | ^ | | | | | |
| | | | | | | | |
| | | ~ | | | | | |
| | Class: | Standard | | | | | |
| F | Revision: | v1.0 | | | | | |
| F | Revision Note: | | | | | | |
| \ \ \ \ | /endor: | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | ОК | Cancel Help | | | |
| Ready | | | | | | | |

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

| OF EKS TODID 1/22 VVVMM | 00 | |
|----------------------------|----|-----------------------|
| OLEKS TOPIP V32 YYY | 2 | 1 |
| SktConEnable | 22 | SktConToServ)- |
| SktEKSData | 2 | |
| ktTimeout | ? | -(SktUsed)- |
| | ?? | |
| iktCreateMsg | ? | SktError)- |
| SktConnectMsg | ? | 1 |
| SktReadMsg | ? | SktOpen)- |
| SktWriteMsg | ? | 1 |
| SktDelMsg | ? | CSktRead)— |
| EKSReadMode | ? | |
| | ?? | -CSktWriteMan) |
| obFinishedActiveTime | ?? | |
| EKSStartAddressRead | ? | -CEKSKeyIN)- |
| | 22 | |
| KSNumberOfBytesRead | ? | -CEKSStatusMessage >- |
| | 11 | CHARTER OF T |
| KSKeyDatakead | | -CJob-Inished J |
| TVO OFBITWORLESS AAULE | 22 | |
| Webburger Of Dute of Maile | 1 | |
| Karvanberoloyteswite | 22 | |
| K RK ou Data Mitta | 2 | |
| I/ORestustionshare | 22 | |

| Variable | Verwendung | Datentyp | Beschreibung |
|----------------------------|------------|--------------------------------|--|
| AOI_EKS_TCPIP_V32_YYYYMMDD | - | AOI_EKS_TCPIP_ V32_YYYYMMDD | Instanz für die AOI |
| SktConEnable | Input | BOOL | Aktiviere TCP/IP Socketverbindung |
| SktEKSData | InOut | typeSktEKSData | Die Variable beinhaltet alle benötigten Daten für die Messages |
| SktTimeout | Input | DINT | Zeitlimit für die Messages (in ms) |
| SktCreateMsg | InOut | MESSAGE | Message zum Erstellen des TCP/IP Socket |
| SktConnectMsg | InOut | MESSAGE | Message zum Aufbau der TCP/IP Socketverbindung |
| SktReadMsg | InOut | MESSAGE | Message zum Lesen des TCP/IP Telegramminhalts |
| SktWriteMsg | InOut | MESSAGE | Message zum Schreiben des TCP/IP Telegramms |
| SktDelMsg | InOut | MESSAGE | Message zum Löschen der TCP/IP Socketverbin- dung |
| EKSReadMode | Input | SINT | Modus der Abfrage 1= manuell; 2= automatisch |
| JobFinishedActiveTime | Input | INT | Zeitwert (in ms) wie lange das Bit <i>JobFinished</i> nach dem Schreibvorgang aktiv bleiben soll |
| EKSStartAdressRead | Input | SINT | Startadresse der anzufragenden Schlüsseldaten |
| EKSNumberOfBytesRead | Input | SINT | Anzahl der anzufragenden Schlüsseldaten |
| EKSKeyDataRead | InOut | SINT[124] | Antwort der Nutzdaten des EKS Schlüssels |
| EKSStartAddressWrite | Input | SINT | Startadresse der zu schreibenden Schlüsseldaten |
| EKSNumberOfBytesWrite | Input | SINT | Anzahl der zu schreibenden Schlüsseldaten |
| EKSKeyDataWrite | InOut | SINT[116] | Zu schreibende Schlüsseldaten |
| EKSStatusNumber | Output | INT | EKS Status |
| SktConToServ | Output | BOOL | Es wird versucht eine Serververbindung herzustellen |
| SktUsed | Output | BOOL | TRUE = der Ablauf der Socketverbindung ist gestar- tet; FALSE = der Ablauf der Socketverbindung ist gestoppt |
| SktError | Output | BOOL | Fehler Socketverbindung |
| SktOpen | Output | BOOL | Socketverbindung ist geöffnet |
| SktRead | Output | BOOL | Telegramm wird empfangen |
| SktWrite | Output | BOOL | Telegramm wird gesendet |
| EKSKeyIN | Output | BOOL | EKS Schlüssel in der Schlüsselaufnahme platziert |
| EKSStatusMessage | Output | BOOL | Eine EKS Statusmeldung wurde empfangen |
| 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.

| New Tag | | × |
|-----------------------------------|----------------------------|--------|
| <u>N</u> ame: | AOI_EKS_Milling | Create |
| Description: | ^ | Cancel |
| | | Help |
| | ~ | |
| <u>U</u> sage: | <controller></controller> | |
| Typ <u>e</u> : | Base ~ Connection | |
| Alias <u>F</u> or: | × | |
| Data <u>T</u> ype: | AOI_EKS_TCPIP_V32_YYYYMMDE | |
| Para <u>m</u> eter Connection: | ~ | |
| Scope: | ■ AP000250 ~ | |
| Cl <u>a</u> ss: | Standard ~ | |
| E <u>x</u> ternal Access: | Read/Write \vee | |
| Style: | \checkmark | |
| Constant | | |
| Seguencing | 1 | |
| Open Config | guration | |
| Open <u>P</u> arar | neter 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

DE

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".

| 💰 Logix Designer - AP000250 [1756-L81ES | 32.12] | | | - | | × |
|--|---|-----------------------------------|--|----|---------------------------|---|
| 🗏 MainProgram - MainRoutine* 🗙 | | | | | | - |
| | ab 👻 (ab) | | | | | |
| 0 AOI EKS AOI EKS SktConEn: SktEKSDa SktEKSDa SktCreatel SktConec SktReadM SktWriteM SktDelMsg EKSReadP JobFinishe EKSStartA EKSNumb | TCPIP V32 YYYY TCPIP_V32 YYYY able ta t Msg <u>EK</u> tMsg sg sg sg Mode dActiveTime ddressRead erOfBytesRead ataRead | MMDD AO S Sk J D D | LEKS_Milling O(+ EKS_Milling 2000 - (SktConToSer 2000 - (SktUsed) - New "EKS_Skt_Create_Msg" Cut Instruction Copy Instruction Paste Delete Instruction Add Ladder Element Edit Main Operand Description | y) | g+X g+C g+V Entf | |
| EKSStartA EKSNumb | .ddressWrite erOfBytesWrite | | Save Instruction Defaults | | | |
| EKSKeyDa EKSStatus | ataWrite Number | | Remove Force | | | |

- Bild 9: Erstellen der Socket Create Message Variable
- 6. Erstellen Sie anschließend die Variable ohne Änderung mit Create.

| New Tag | | × |
|------------------------------|---------------------------|------------|
| <u>N</u> ame: | EKS_Skt_Create_Msg | Create 🗸 🔻 |
| Description: | ^ | Cancel |
| | | Help |
| | ~ | |
| <u>U</u> sage: | <controller></controller> | |
| Тур <u>е</u> : | Base ~ <u>C</u> onnection | |
| Alias <u>F</u> or: | ~ | |
| Data <u>T</u> ype: | MESSAGE | |
| Parameter Connection: | × | |
| <u>S</u> cope: | P000250 ~ | |
| Cl <u>a</u> ss: | Standard ~ | |
| E <u>x</u> ternal Access: | Read/Write ~ | |
| St <u>y</u> le: | ~ | |
| Constant | | |
| Seguencin | 9 | |
| Open MES | SAGE Configuration | |
| Open <u>P</u> ara | meter Connections | |

Bild 10: Erstellen der Socket Create Message Variable

| i | |
|---|--|

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 SktWrite empfangen 2 = Automatischer Modus; Das Empfangen der Schlüsseldaten wird durch Auslesen des Key-Status KeylN (siehe Handbuch) getriggert. |
| JobFinishedActiveTime | Variabel (Wert von 0 bis 65535) → Zeitwert (in ms) wie lange das Bit JobFinished nach dem Schreibvorgang aktiv bleiben soll |
| EKSStartAddressRead | Variabel (Wert von 0 bis 116) \rightarrow Startadresse der verwendeten Nutzdaten |
| EKSNumberOfBytesRead | Variabel (Wert von 0 bis 124) \rightarrow 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 |

 (\mathbf{i})

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 byteweise 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.





DE

5.3. Konfiguration der Messages

1. Öffnen Sie den Configuration Dialog der SktCreateMsg.

| AOI_EKS_TCPIP_V32_YY | YYMMDD | |
|--|---|---------------------------------|
| AOI_EKS_TCPIP_V32_YY SktConEnable SktEKSData SktTimeout | Y AOI_EKS_Milling 0 EKS_Milling 2000 | |
| SktCreateMsg SktConnectMsg E SktReadMsg SktWriteMsg SktDelMsg EKSReadMode | EKS_Skt_Create_Msg KS_Skt_Connect_Msg EKS_Skt_Read_Msg EKS_Skt_Read_Msg EKS_Skt_Write_Msg EKS_Skt_Del_Msg 2 | -(SktError) |
| JobFinishedActiveTime EKSStartAddressRead | 500 + 0 | –(SktWriteMan)— –(EKSKeyIN)— |
| EKSNumberOfBytesRead | 124 | |
| EKSKeyDataRead EKSStartAddressWrite | EKS_Key_Data_Read 0 | (JobFinished) |
| EKSNumberOfBytesWrite | 116 | |
| EKSKeyDataWrite EKSStatusNumber | EKS_Key_Data_Write 16#0000 ← | |

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 Communication) |



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.

| Message Configuration - EKS_Skt_Create_Msg | Message Configuration - EKS_Skt_Create_Msg |
|--|---|
| Configuration* Communication Tag | Configuration* Communication* Tag |
| Message Type: CIP Generic Service Socket Create Type: Source Length: Service 4b Code: 4b Instance: 0 Attribute: 0 (Hex) Hex) Bestination EKS_Milling.CreateSrc Instance: 0 Attribute: 0 (Hex) Hex) | Path: THIS THIS THIS THIS THIS THIS TOrmunication Method O(P O DH+ Channet: A* ♥ Destination Link: 0 ♥ O(P With Source ID Source Link: 0 ♥ Destination Node: 0 ♥ (Octal) Cgnnected Cachg Connections ◆ Large Connection |
| C Enable C Enable Watting C Start C Done Done Length: 0 | ◯ Enable ◯ Enable Watting ◯ Start ◯ Done Done Length: 0 |
| ○ Error Code: Extended Error Code: □ Timed Out ◆ Error Path: Error Text: | ○ Error Code: Extended Error Code: |
| OK Abbrechen Ü <u>b</u> ernehmen Hilfe | OK Abbrechen Obernehmen Hilfe |

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 Pfad (Path) finden Sie unter dem Reiter Communication) |

| Message Configuration - EKS_Skt_Connect_Msg | Message Configuration |
|--|---|
| Configuration* Communication Tag | Configuration* Communication* Tag |
| Message Type: CIP Generic Service OpenConnection Type: Source Length: Service 4c Code: (Hex) Code: 0 Attribute: 0 (Hex) Element: New Tag | Path: THIS From the second |
| Constant Code: Done Length: 0 Code: Extended Error Code: □ Timed Out ← Error Path: Error Text: | O Enable ○ Enable Waiting ○ Start ○ Done Done Length: 0 O Error Code: Extended Error Code: □ Timed Out ◆ Error Path: Error Text: |
| OK Abbrechen Übernehmen Hilfe | OK Abbrechen Übernehmen Hilfe |

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 Pfad (Path) finden Sie unter dem Reiter Communication. |

| Message Configuration - EKS_Skt_Read_Msg | Message Configuration |
|---|---|
| Configuration* Communication Tag | Configuration* Communication* Tag |
| Message Type: CIP Generic Service ReadSocket Type: Source Length: Service 4d (Hex) Class: 342 (Hex) Destination EKS_Milling.ReadRes Instance: 0 Attribute: 0 (Hex) Hex) | Path: THIS Browse THIS Broadcast: Communication Method |
| C Enable ○ Enable Waiting ○ Start ○ Done Done Length: 0 Error Code: Extended Error Code: □ Timed Out ◆ Error Path: Error Text: OK Abbrechen Übernehmen Hilfe | C Enable ○ Enable Waiting ○ Start ○ Done Done Length: 0 Error Code: Extended Error Code: □ Timed Out ◆ Error Path: Error Text: OK Abbrechen Objemehmen Hilfe |

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 Communication) |

| Message Configuration - EKS_Skt_Write_Msg | Message Configuration |
|--|---|
| Configuration* Communication Tag | Configuration* Communication* Tag |
| Message Type: CIP Generic Service WriteSocket Type: Source Element: Service 4e (Hex) Class: 342 (Hex) Destination EKS_Milling.WriteSize Instance: 0 Attribute: 0 (Hex) Element: | Path: THIS THIS THIS THIS THIS THIS Communication Method CIP DH+ Channel: A |
| C Enable ○ Enable Waiting ○ Start ○ Done Done Length: 0 Error Code: Extended Error Code: □ Timed Out ◆ Error Path: Error Text: OK Abbrechen Übernehmen Hilfe | ○ Enable ○ Enable Waiting ○ Start ○ Done Done Length: 0 ○ Error Code: Extended Error Code: □ Timed Out ◆ Error Path: Error Text: OK Abbrechen Übernehmen Hilfe |

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 Communication) |
| | | Ϋ́ |
| Message Configuration - EKS_Skt | _Del_Msg | Message Configuration |
| Configuration* Communication | Tag | Configuration* Communication* Tag |
| Message <u>I</u> ype: CIP Gen Service <u>DeleteSocket</u> Service <u>4f</u> (Hex) <u>Class</u> Code: <u>4f</u> (Hex) <u>Class</u> <u>Instance</u> <u>0</u> Attribute: | eric Source Element: Source Length: O (Bytes) G (Hex) Destination Element: New Tag | ● Path: THIS Browse THIS ● Broadcast: ● ● Broadcast: ● ● ● Communication Method ● ○ ● ● CIP ● DH+ Channel: ▲ ● ● ● CIP ● DH+ Channel: ▲ ● |
| Enable C Enable Waiting Error Code: Exten Error Path: Error Fath: Error Tatt: | O Start O Done Done Length: 0 ded Error Code: □ Timed Out ◆ | C Enable C Enable Waiting Start Done Done Length: 0 C Error Code: Extended Error Code: Timed Out Error Path: Error Text: |
| | OK Abbrechen Obernehmen Hilfe | OK Abbrechen Übernehmen Hilfe |

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.

| 💰 Logix D | esigner - AP000250 [| 1756-L81ES 32.12] | | | - 🗆 × | | | |
|-----------|----------------------|--|-------------------------------|--|---------------------------------------|------------------|-----------------------|------------------------------|
| 📙 MainPr | rogram - MainRoutir | ne* × | | | - | | | |
| e e | | abrd ab_ eb (ab) | | | | | | |
| 0 😵 | EnableSocket | | 4 | AOI_EKS_Milling.Skt | ConEnable | | Show All Tage | |
| | | | | I _₹ Em | crivanci nici | | Silow. All rags | ~ |
| 1 | | AOLEKS_TCPIP_V32_Y | YYYMMDD YY AQLEKS Milling | Nar | ne I EKS Milling | | | ^ |
| | | SktConEnable | 0* | -CSktConT | AOLEKS_Willing EnableIn | AUI_EKS BOOL | | |
| | | SktEKSData | EKS_Milling | Skillord | AOI_EKS_Milling.EnableOut | BOOL | | |
| | | Skillmeout | 2000 | -C SKLUSED | AOI_EKS_Milling.SktConToSe | erv BOOL | | |
| | | SktCreateMsg | EKS_Skt_Create_Msg | -CSktError | AOI_EKS_Milling.SktConEnat | ble BOOL | | |
| | | SktConnectMsg SktReadMsg | EKS_Skt_Connect_Msg | -CSktOpen | AOLEKS_Willing.SktUsed | Name: AOL EKE | Milling EldConEpoble | |
| | | SktWriteMsg | EKS_Skt_Write_Msg | | Act_Erco_winning.oktError | Data Type: BOOI | winning.skiConenable | ` |
| | | SktDelMsg EKSReadMode | EKS_Skt_Del_Msg 2 | -CSktRead Sho | w <u>c</u> ontroller tags | Description: Ena | ble Socket connection | ✓ Show <u>s</u> tandard tags |
| | | | | -CSktWrite Sho | w MainProgram tags | | | Show safety tags |
| | | JobFinishedActiveTime EKSStartAddressRead | 500 🖛 | EKSKey | | | | |
| | | Eltootarb-duressitead | U U | Show p | arameters from <u>o</u> ther program: | | | |
| | | EKSNumberOfBytesRead | 124 | -CEKSStat <none:< th=""><th>></th><th>~</th><th></th><th></th></none:<> | > | ~ | | |
| | | EKSKeyDataRead EKSStartAddressWrite | EKS_Key_Data_Read 0 | | | | | |
| | | EKSNumberOfBytesWrite | e 116 | | | | | |
| | | EKSKeyDataWrite EKSStatusNumber | EKS_Key_Data_Write 16#0000 | | | | | |



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

DE

5.6. Parametrieren der IP-Adresse

i

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.

| Controller Tags - AP000250 | (controller) × | | | | | | |
|---|----------------|--------------------|-----------------|--------|-----|---------|------------|
| Scope: AP000250 | Show: All Tags | | | | | | |
| Name | ; | 📰 👻 Value | | | | | • |
| Local:3:0 | | | | | | | { |
| Local:3:1 | | string Browser - E | KS Milling.IPAc | lress* | | | × |
| Local:3:C | | | | 4.0264 | | | ** |
| Local:2:1 | 192. | 168.0.220 | | | | ^ | s* |
| Local:2:C | | | | | | | \$L |
| EKS_Skt_Write_Msg | | | | | | | \$N \$P |
| EKS_Skt_Read_Msg | | | | | | | \$R |
| EKS_Skt_Del_Msg | | | | | | | ŝT |
| EKS_Skt_Create_Msg | | | | | | ~ | |
| EKS_Skt_Connect_Msg | | ОК | Cancel | Apply | H | Help | |
| EKS_Milling | 6 3 | 0 Error(s) | | 13 | INS | 13 of 8 | 2 |
| EKS_Milling.IPAdress | | — | | | | | - 2 |
| EKS_Milling.Inst | | | | | | | C |
| EKS_Milling.WriteSizeSe | ent | | | | | | C |
| EKS_Milling.CreateSrc | | | | | | | {} |
| EKS_Milling.OpenSrc | | | | | | | {] |
| EKS_Milling.ReadSrc | | | | | | | {] |
| | | | | | | | {] |
| EKS_Milling.ReadRespon | nse | | | | | | |
| EKS_Milling.ReadRespo EKS_Milling.WriteSrc | onse | | | | | | { |



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.

| 🧟 Contro | Controller Tags - AP000250(controller) × Scope: Image: Aproxema and the second | | | | | | |
|------------------------------|---|-------|----------|----------------|---------|-----------|--|
| Scope: | ■ AP000250 ~ | Show: | All Tags | | | | |
| Name | | | Value 🔹 | Force Mask 🔹 🕈 | Style | Data Type | |
| ▲ EKS | _Key_Data_Read | | {} | {} | ASCII 🗸 | SINT[124] | |
| Þ EK | S_Key_Data_Read[0] | | 'E' | | ASCII | SINT | |
| Þ EK | S_Key_Data_Read[1] | | 'U' | | ASCII | SINT | |
| ▶ EK | S_Key_Data_Read[2] | | 'C' | | ASCII | SINT | |
| ► EK | S_Key_Data_Read[3] | | .н. | | ASCII | SINT | |
| ▶ EK | S_Key_Data_Read[4] | | 'N' | | ASCII | SINT | |
| ▶ EK | S_Key_Data_Read[5] | | 'E' | | ASCII | SINT | |
| ▶ EK | S_Key_Data_Read[6] | | 'R' | | ASCII | SINT | |
| ▶ EK | S_Key_Data_Read[7] | | | | ASCII | SINT | |
| ▶ EK | S_Key_Data_Read[8] | | τ | | ASCII | SINT | |
| ▶ EK | (S_Key_Data_Read[9] | | 'E' | | ASCII | SINT | |
| ▶ EK | S_Key_Data_Read[10] | | 'V' | | ASCII | SINT | |
| ▶ EK | S_Key_Data_Read[11] | | 'E' | | ASCII | SINT | |
| ▶ EK | S_Key_Data_Read[12] | | τ | | ASCII | SINT | |
| ► EK | S_Key_Data_Read[13] | | | | ASCII | SINT | |
| ▶ EK | S_Key_Data_Read[14] | | '3' | | ASCII | SINT | |
| ▶ EK | S_Key_Data_Read[15] | | \$00' | | ASCII | SINT | |
| ▶ EK | S_Key_Data_Read[16] | | \$00 | | ASCII | SINT | |
| ⊳ Ek | S_Key_Data_Read[17] | | \$00 | | ASCII | SINT | |
| ▶ EK | S_Key_Data_Read[18] | | \$00 | | ASCII | SINT | |
| ▶ EK | S_Key_Data_Read[19] | | \$00 | | ASCII | SINT | |
| ▶ EK | S_Key_Data_Read[20] | | \$00 | | ASCII | SINT | |
| ♦ Monitor Tags { Edit Tags / | | | | | | | |

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 Bespiel ändern wir die Stufe 3 auf Stufe 5.

| Controller Tags - AP000250(controller) × | | | | | | | |
|--|-------|----------------|--------------|-------|-----------|--|--|
| Scope: PAP000250 | 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] | | .н. | | 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] | | Ϋ́ | | ASCII | SINT | | |
| EKS_Key_Data_Write[9] | | 'E' | | ASCII | SINT | | |
| EKS_Key_Data_Write[10] | I | 'V' | | ASCII | SINT | | |
| EKS_Key_Data_Write[11] | l | 'E' | | ASCII | SINT | | |
| EKS_Key_Data_Write[12] | 1 | T | | ASCII | SINT | | |
| EKS_Key_Data_Write[13] | 1 | | | ASCII | SINT | | |
| EKS_Key_Data_Write[14] | 1 | 5' | | ASCII | SINT | | |
| EKS_Key_Data_Write[15] | 1 | '\$00' | | ASCII | SINT | | |
| EKS_Key_Data_Write[16] | 1 | \$00 | | ASCII | SINT | | |
| EKS_Key_Data_Write[17] | 1 | '\$00' | | ASCII | SINT | | |
| EKS_Key_Data_Write[18] | I | '\$00' | | ASCII | SINT | | |
| EKS_Key_Data_Write[19] | l | '\$00' | | ASCII | SINT | | |
| EKS_Key_Data_Write[20] | 1 | ' \$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] |
| ►E | KS_Key_Data_Read[0] | | | 'E' | | ASCII | SINT |
| ►E | KS_Key_Data_Read[1] | | | 'U' | | ASCII | SINT |
| ▶ E | KS_Key_Data_Read[2] | | | 'C' | | ASCII | SINT |
| ►E | KS_Key_Data_Read[3] | | | 'H' | | ASCII | SINT |
| ▶ E | KS_Key_Data_Read[4] | | | 'N' | | ASCII | SINT |
| ▶ E | KS_Key_Data_Read[5] | | | 'E' | | ASCII | SINT |
| ►E | KS_Key_Data_Read[6] | | | 'R' | | ASCII | SINT |
| ▶ E | KS_Key_Data_Read[7] | | | | | ASCII | SINT |
| ►E | KS_Key_Data_Read[8] | | | T | | ASCII | SINT |
| ►E | KS_Key_Data_Read[9] | | | 'E' | | ASCII | SINT |
| ÞЕ | KS_Key_Data_Read[10] | | | 'V' | | ASCII | SINT |
| ►E | KS_Key_Data_Read[11] | | | 'E' | | ASCII | SINT |
| ►E | KS_Key_Data_Read[12] | | | Ľ | | ASCII | SINT |
| ►E | KS_Key_Data_Read[13] | | | | | ASCII | SINT |
| ÞΕ | KS_Key_Data_Read[14] | | | '5' | | ASCII | SINT |
| ►E | KS_Key_Data_Read[15] | | | '\$00' | | ASCII | SINT |
| ▶ E | KS_Key_Data_Read[16] | | | '\$00' | | ASCII | SINT |
| ▶ E | KS_Key_Data_Read[17] | | | '\$00' | | ASCII | SINT |
| ►E | KS_Key_Data_Read[18] | | | '\$00' | | ASCII | SINT |
| ►E | KS_Key_Data_Read[19] | | | '\$00' | | ASCII | SINT |
| ►E | KS_Key_Data_Read[20] | | | '\$00' | | ASCII | SINT |
| ► \ M | onitor Tags (Edit Tag | s / | | | | < | |

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.

DE

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.