



Contents

1.	About this document.....	3
1.1.	Version	3
1.2.	Scope.....	3
1.3.	Target group	3
1.4.	Supplementary documents	3
1.5.	Notice.....	3
2.	Components/modules used	4
2.1.	EUCHNER	4
2.2.	Others	4
2.3.	Software	4
3.	Functional description.....	4
4.	Importing the function block	5
5.	Integrating the function block	7
5.1.	Function block variable table.....	7
5.2.	Adding the function block	7
6.	Reading and writing Electronic-Key data	9
6.1.	Transferring program to the PLC.....	9
6.2.	Reading contents of the Electronic-Key memory using a watch and force table	9
6.3.	Writing contents of the Electronic-Key memory using a watch and force table.....	10
7.	Important note – please observe carefully!	12

1. About this document

1.1. Version

Version	Date	Change/addition	Chapter
01-09/20	9/9/2020	Prepared	All

1.2. Scope



The purpose of this document is the integration and configuration of the EKS with TCP/IP interface in BECKHOFF TwinCAT3®.

1.3. Target group

Design engineers and installation planners for safety systems on machines, as well as setup and servicing staff possessing special expertise in handling safety components as well as expertise in the installation, setup, programming and diagnostics of programmable logic controllers (PLCs) and bus systems.

1.4. Supplementary documents

The overall documentation for this application consists of the following documents:

Document title (document number)	Contents	
Manual (2100420)	Electronic-Key-System Manual for Electronic-Key adapter EKS and EKS FSA with Ethernet TCP/IP interface	
Possibly enclosed data sheets	Item-specific information about deviations or additions	

1.5. Notice

This application is based on the manual for the EKS with TCP/IP interface. Please refer to the manual for the technical details and other information. In the rest of this document the EKS with TCP/IP interface is referred to as the "EKS" for short.

2. Components/modules used

2.1. EUCHNER

Description	Order number / item
EKS with TCP/IP interface	100401 / EKS-A-EX-G01-ST02/03
	099265 / EKS-A-EXA-G01-ST02/03/04



TIP!

More information and downloads about the aforementioned EUCHNER products can be found at www.euchner.com. Simply enter the order number in the search box.

2.2. Others

Description	Order number / item
BECKHOFF CX9020-0110-M930	CX9020-0110-M930

2.3. Software

Description	Version
BECKHOFF TwinCAT3®	3.1.4024.10
BECKHOFF TF6310 TC3 TCP/IP including license	3.1.9.0

3. Functional description

EKS TCP/IP devices are read/write systems with electronics for the inductive bidirectional interface to the transponder and interface electronics.

The system is connected via the integrated TCP/IP interface, which is designed as an RJ45 socket. A separate switch may be required for the TCP/IP connection. The EKS does not have an integrated switch.

The current state of the Electronic-Key adapter is displayed using a 3-color LED.

The Electronic-Key is placed on the Electronic-Key adapter for operation. The power supply for the transponder and the data are transferred between the Electronic-Key adapter and the Electronic-Key without using any contacts.

The data transmission between the control system and EKS is realized using a function block. The function block handles the establishment of the communication between the control system and EKS as well as sending and receiving the TCP/IP communication telegrams.

The function block be downloaded from www.euchner.com in the area *Downloads/Software/Sample files and Libraries/EKS*.

4. Importing the function block

1. Right-click *POUs* and select *Import from ZIP*.

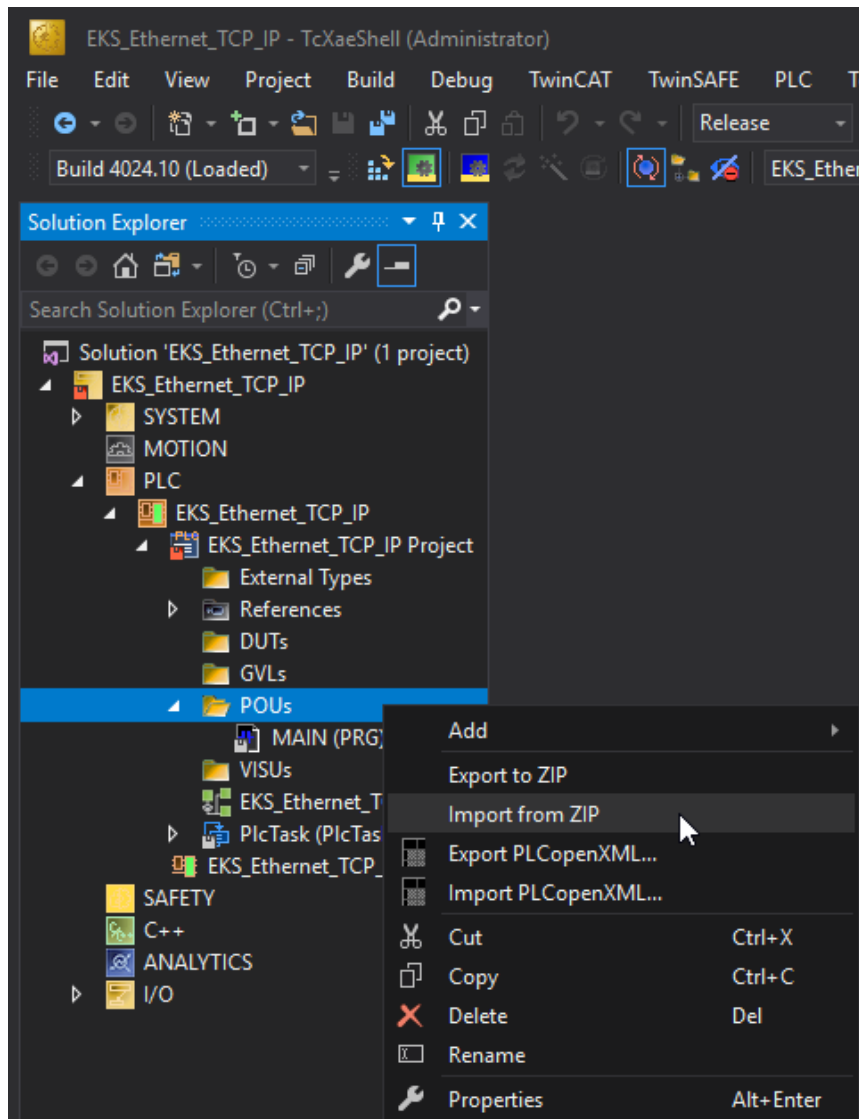


Figure 1: Selecting *Import from ZIP*

2. Select the compressed (ZIP) folder and click on *Open*.

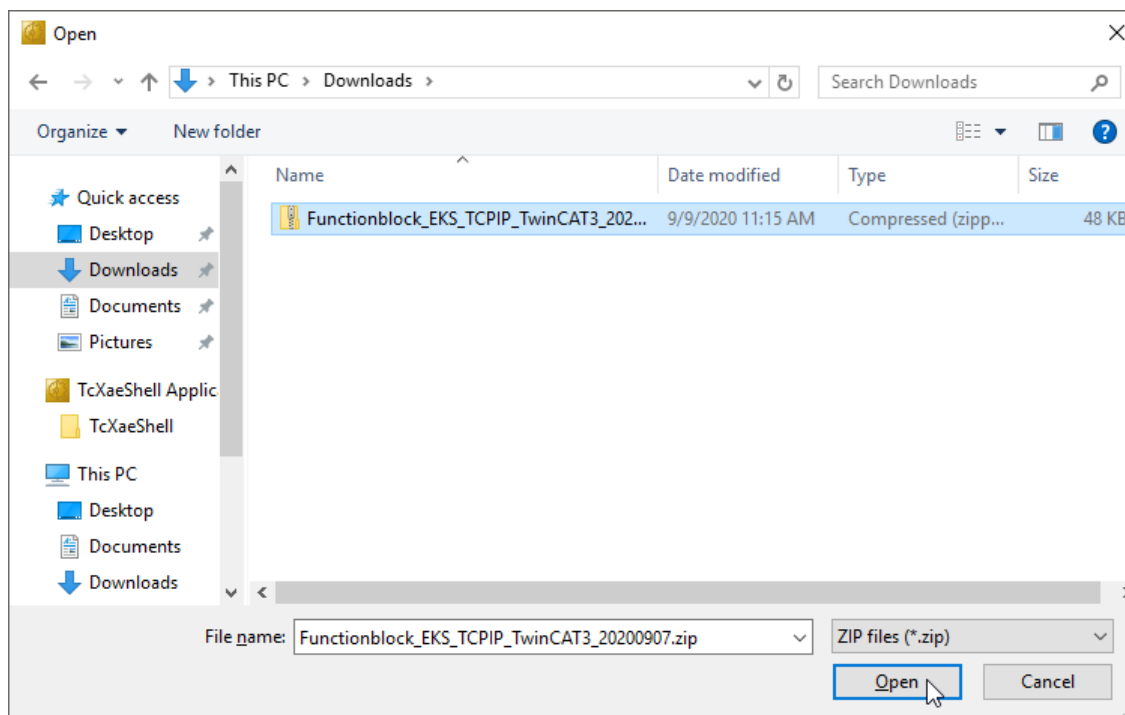


Figure 2: Selecting the compressed folder

3. Complete the function block import by clicking on *OK*.

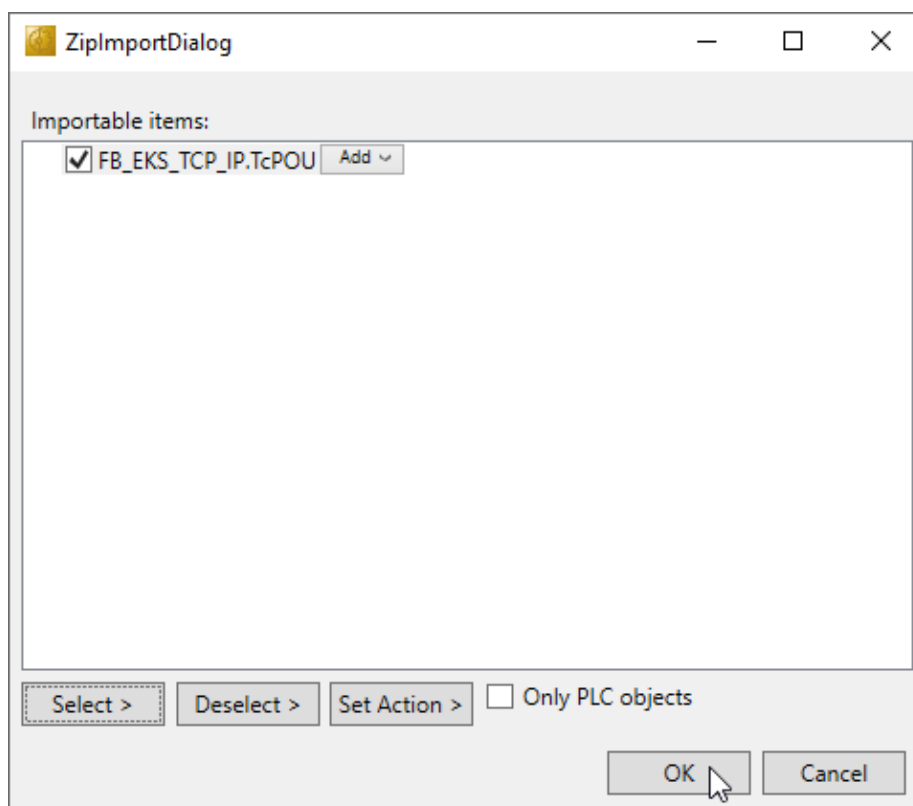


Figure 3: Completing the import

5. Integrating the function block

5.1. Function block variable table

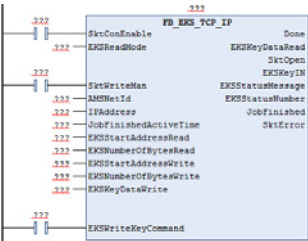
	Variable	Use	Data type	Description
	FB_EKS_TCP_IP	-	FB_EKS_TCP_IP	Instance for the FB
	SktConEnable	Input	BOOL	Activates TCP/IP socket connection
	EKSReadMode	Input	USINT	Mode for the request 1= manual; 2= automatic
	SktWriteMan	Input	BOOL	Trigger for reading the data in <i>EKSReadMode</i> = 1
	AMSNetId	Input	T_AmsNetId	Address of the local computer in the TwinCAT network
	IPAddress	Input	T_IPv4Addr	EKS IP address
	JobFinishedActiveTime	Input	TIME	Time value indicating how long the <i>JobFinished</i> bit is to remain active after the write process
	EKSStartAddressRead	Input	BYTE	Start address for the Electronic-Key data to be requested
	EKSNumberOfBytesRead	Input	BYTE	Amount of Electronic-Key data to be requested
	EKSStartAddressWrite	Input	BYTE	Start address of the Electronic-Key data to be written
	EKSNumberOfBytesWrite	Input	BYTE	Amount of Electronic-Key data to be written
	EKSKeyDataWrite	Input	Array[0..115] of BYTE	Data to be written
	EKSWriteKeyCommand	Input	BOOL	Command for writing Electronic-Key memory
	Done	Output	BOOL	-
	EKSKeyDataRead	Output	Array[0..123] of BYTE	Reply with the user data from the EKS Electronic-Key
	SktOpen	Output	BOOL	Socket connection opened
	EKSKeyIN	Output	BOOL	EKS Electronic-Key placed in the Electronic-Key adapter
	EKSStatusMessage	Output	BOOL	An EKS status message has been received
	EKSStatusNumber	Output	BYTE	EKS status
	JobFinished	Output	BOOL	Write process completed
	SktError	Output	BOOL	Socket connection error

Table 1: Function block variable table

5.2. Adding the function block

1. Open a program (e.g. *MAIN (PRG)*) and drag the block from the *Toolbox* to a new network using drag & drop.

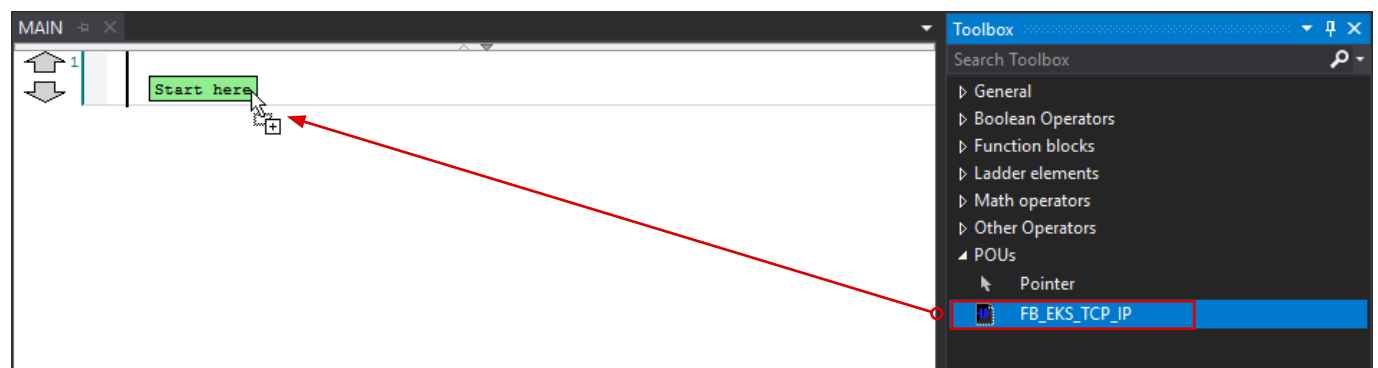


Figure 4: Adding the block to the main program

2. An instance of the block is created automatically. Type the **name** of the instance *FB_EKS_TCP_IP* in the field for the variable. In this example, enter *EKS_Milling* and confirm by pressing the enter key. In the variable table (*MAIN (PRG)*), the variable is automatically created with the associated variable type.
3. Now create the corresponding variables for every input and output, as in Step 2.
4. Add an input variable (e.g. *EKS_Milling_SktConEnable*) to the network as a contact.

Variable	Value
EKSReadMode	1 = manual mode; the Electronic-Key data are received by triggering the variable <i>SktWriteMan</i> 2 = automatic mode; the reception of the Electronic-Key data is triggered by reading the Electronic-Key status <i>KeyIN</i> (see manual)
AMSNetId	Address of the local computer in the TwinCAT network, enclosed in single quotation marks
IPAddress	IPv4 address, enclosed in single quotation marks, e.g. '192.168.0.222'
JobFinishedActiveTime	Variable (time value (e.g. T#500 ms) → Time value indicating how long the <i>JobFinished</i> bit is to remain active after the write process
EKSStartAddressRead	Variable (value from 0 to 116) → Start address for the user data used
EKSNumberOfBytesRead	Variable (value from 1 to 124) → Number of bytes to be read
EKSStartAddressWrite	Variable (value from 0 to 112) → Start address for the user data to be written
EKSNumberOfBytesWrite	Variable (value from 4 to 116) → Number of bytes to be written

Table 2: Input variables

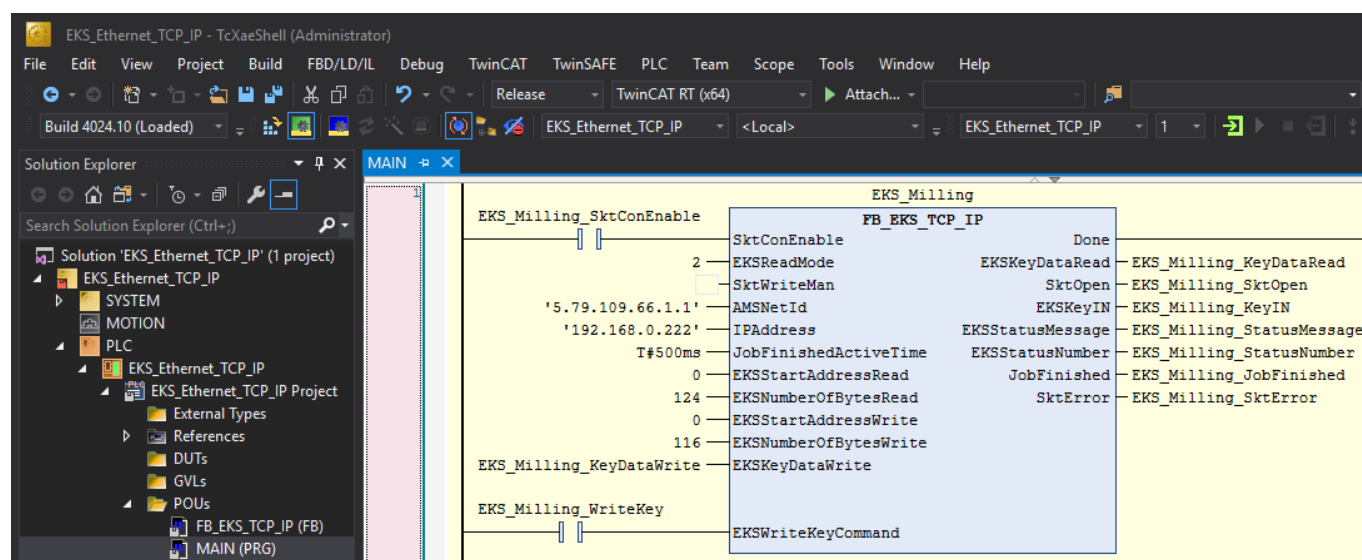


Figure 5: Declaring the instance block



IMPORTANT

On the Electronic-Key read/write with 116 bytes freely programmable, the memory is organized in 4-byte blocks. This means the start address for writing must be given in the range byte number 0 to byte number 112, always in 4-byte steps (byte number 0, 4, 8 ... 112). Also, a multiple of 4-byte-sized blocks must always be written (4, 8, 12 ... 116 bytes)! However, during reading it is possible to access the memory byte-by-byte without the above-mentioned restriction for writing. The Electronic-Key read/write also has a unique 8-byte serial number that is permanently written to the memory during the Electronic-Key production process. The serial number therefore cannot be changed. This serial number is used for secure differentiation of every single Electronic-Key. It is necessary that all 8 bytes are completely evaluated for secure differentiation. The serial number is appended to the freely programmable memory. The serial number can be read by entering the start address byte number 116 and the number of bytes 8.



TIP

The IP address of the Electronic-Key adapter EKS is assigned using the web interface. You will find the related description in the manual, chapter 7.2.

6. Reading and writing Electronic-Key data

6.1. Transferring program to the PLC

Compile the program, load it into the control system and then go online.

6.2. Reading contents of the Electronic-Key memory using a watch and force table

By setting the bit *EKS_Milling_SktConEnable*, the connection to the EKS is established and *SktOpen* becomes *True*. As soon as an Electronic-Key is placed in the Electronic-Key adapter, *EKSKeyIN* changes to *True*.

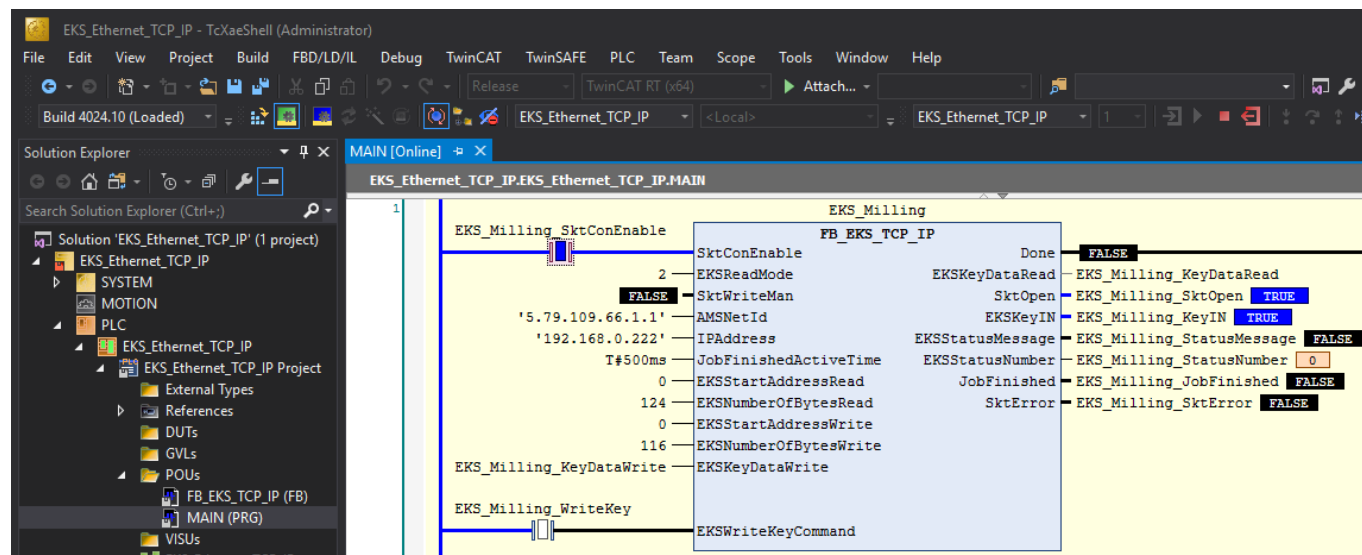



Figure 6: Instance block online

By setting the parameters on the input *EKSReadMode* with the value 2, the Electronic-Key data are automatically read after the Electronic-Key has been placed. This can be monitored in a watch and force table.

Watch 1			
Expression	Application	Type	Value
MAIN.EKS_Milling_KeyDataRead	EKS_Ethernet_TCP...	ARRAY [0..123] OF BYTE	
EKS_Milling_KeyDataRead[0]		BYTE	16#01
EKS_Milling_KeyDataRead[1]		BYTE	16#02
EKS_Milling_KeyDataRead[2]		BYTE	16#03
EKS_Milling_KeyDataRead[3]		BYTE	16#04
EKS_Milling_KeyDataRead[4]		BYTE	16#05
EKS_Milling_KeyDataRead[5]		BYTE	16#06

Figure 7: Reading Electronic-Key data in a watch and force table

6.3. Writing contents of the Electronic-Key memory using a watch and force table

The watch and force table has been prepared such that data can also be written to the Electronic-Key. Values were prepared to this end, as shown in the screenshot below. Values entered into the *Prepared value* column must be confirmed by selecting the *Write values*  button. The values are then adopted into the *Value* column.

Watch 1				
Expression	Application	Type	Value	Prepared value
MAIN.EKS_Milling_KeyDataRead	EKS_Ethernet_TCP...	ARRAY [0..123] OF BYTE		
MAIN.EKS_Milling_KeyDataWrite	EKS_Ethernet_TCP...	ARRAY [0..115] OF BYTE		
EKS_Milling_KeyDataWrite[0]		BYTE	16#0A	16#0A
EKS_Milling_KeyDataWrite[1]		BYTE	16#0B	16#0B
EKS_Milling_KeyDataWrite[2]		BYTE	16#0C	16#0C
EKS_Milling_KeyDataWrite[3]		BYTE	16#0D	16#0D
EKS_Milling_KeyDataWrite[4]		BYTE	16#0E	16#0E
EKS_Milling_KeyDataWrite[5]		BYTE	16#0F	16#0F

Figure 8: Writing Electronic-Key data in a watch and force table

Running the *EKSWriteKeyCommand* write command will write the modified data in the Electronic-Key memory. This bit must then be reset. An edge is sufficient to run the command.

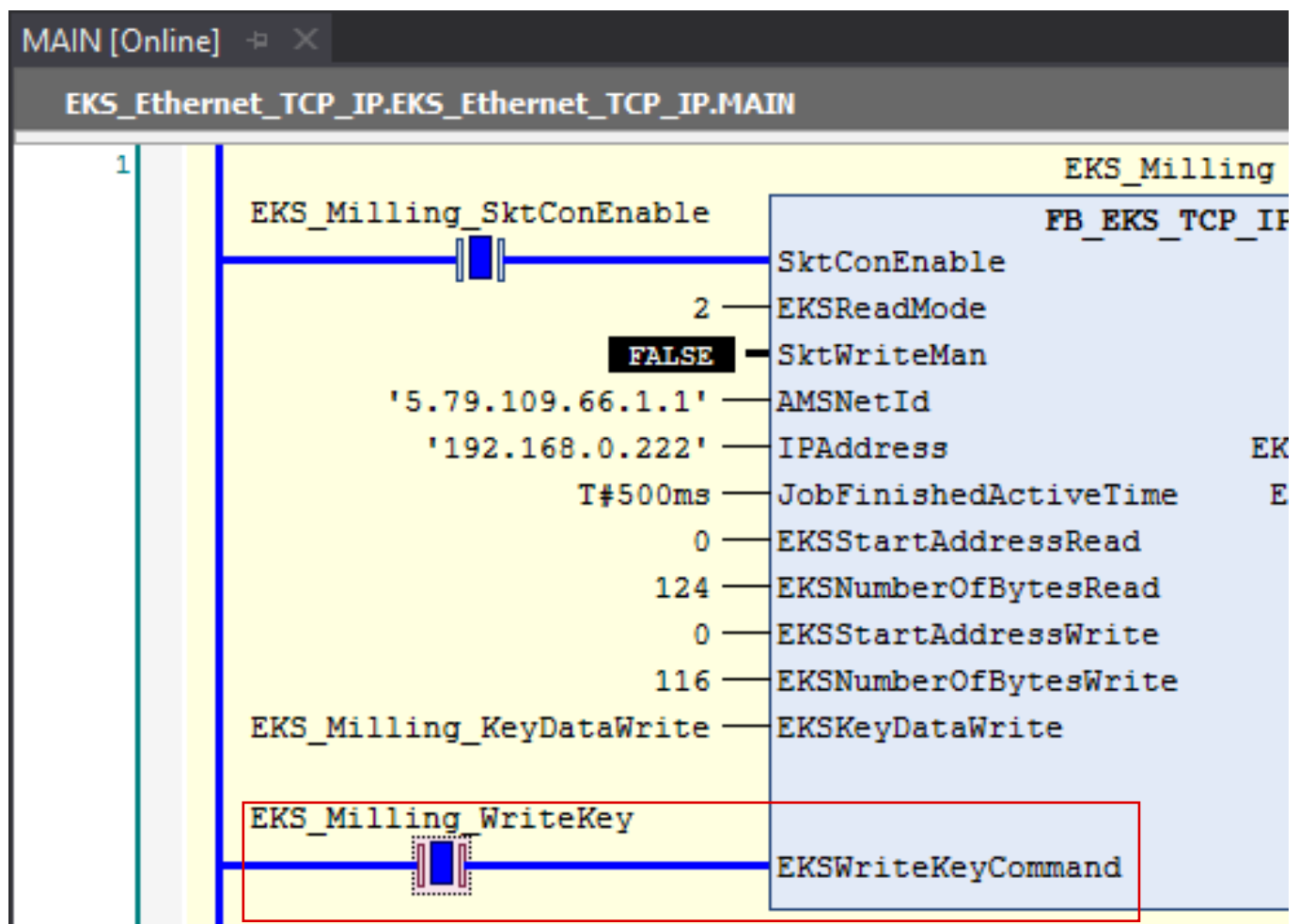


Figure 9: Command for writing Electronic-Key








Watch 1			
Expression	Application	Type	Value
 MAIN.EKS_Milling_KeyDataRead	EKS_Ethernet_TCP...	ARRAY [0..123] OF BYTE	
 EKS_Milling_KeyDataRead[0]		BYTE	16#0A
 EKS_Milling_KeyDataRead[1]		BYTE	16#0B
 EKS_Milling_KeyDataRead[2]		BYTE	16#0C
 EKS_Milling_KeyDataRead[3]		BYTE	16#0D
 EKS_Milling_KeyDataRead[4]		BYTE	16#0E
 EKS_Milling_KeyDataRead[5]		BYTE	16#0F

Figure 10: Reading Electronic-Key data in a watch and force table, updated

7. Important note – please observe carefully!

This document is intended for a design engineer who possesses the requisite knowledge in safety engineering and knows the applicable standards, e.g. through training for qualification as a safety engineer. Only with the appropriate qualification is it possible to integrate the example provided into a complete safety chain.

The example represents only part of a complete safety chain and does not fulfill any safety function on its own. In order to fulfill a safety function, the energy switch-off function for the danger zone and the software must also be considered in the safety evaluation, for example.

The applications provided are only examples for solving certain safety tasks for protecting safety doors. The examples cannot be comprehensive due to the application-dependent and individual protection goals within a machine/installation.

If questions concerning this example remain open, please contact us directly.

According to the Machinery Directive 2006/42/EC, the design engineer of a machine or installation has the obligation to perform a risk assessment and take measures to reduce the risk. While doing this, the engineer must comply with the applicable national and international safety standards. Standards generally represent the current state-of-the-art. Therefore, the design engineer should continuously inform himself about changes in the standards and adapt his considerations to them. Relevant standards for functional safety include EN ISO 13849 and EN 62061. This application must be regarded only as assistance for the considerations about safety measures.

The design engineer of a machine/installation has the obligation to assess the safety technology himself. The examples must not be used for an assessment, because only a small excerpt of a complete safety function was considered in terms of safety engineering here.

In order to be able to use the safety switch applications correctly on safety doors, it is indispensable to observe the standards EN ISO 13849-1, EN ISO 14119 and all relevant C-standards for the respective machine type. Under no circumstances does this document replace the engineer's own risk assessment, and it cannot serve as the basis for a fault assessment.

In particular in relation to a fault exclusion, it must be noted that a fault can be excluded only by the machine's or installation's design engineer and this action requires justification. A general fault exclusion is not possible. More information about fault exclusion can be found in EN ISO 13849-2.

Changes to products or within assemblies from third-party suppliers used in this example can lead to the function no longer being ensured or the safety assessment having to be adapted. In any event, the information in the operating instructions on the part of EUCHNER, as well as on the part of third-party suppliers, must be used as the basis before this application is integrated into an overall safety function. If contradictions should arise between the operating instructions and this document, please contact us directly.

Use of brand names and company names

All brand names and company names stated are the property of the related manufacturer. They are used only for the clear identification of compatible peripheral devices and operating environments in relation to our products.

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

Edition:
AP000261-01-09/20
Title:
Application EKS
Integration of EKS with TCP/IP Interface in BECKHOFF
TwinCat3®

Copyright:
© EUCHNER GmbH + Co. KG, 09/2020

Subject to technical modifications; no responsibility is accepted for the accuracy of this information.