EUCHNER

Application

Integration of EKS with TCP/IP Interface in OMRON Sysmac Studio®

EN

Contents

1. About this document

1.1. Version

Version	Date	Change/addition	Chapter
01-08/20	8/21/2020	Prepared	All

1.2. Scope

The purpose of this document is the integration and configuration of the EKS with TCP/IP interface in OMRON Sysmac Studio®.

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.

EN

2. Components/modules used

2.1. EUCHNER

Description	Order number / item
EKS with TCP/IP interface	100401 / EKS-A-IEX-G01-ST02/03
	099265 / EKS-A-IEXA-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
NX102-1120 CPU UNIT	NX102-1120

2.3. Software

Description	Version
OMRON Sysmac Studio®	Version Lite Edition V1.27

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 library. The library handles the establishment of the communication between the control system and EKS as well as sending and receiving the TCP/IP communication telegrams.

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

4. Importing the library

1. Open the *Project* tab and select *Library* → *Show References* from the context menu.

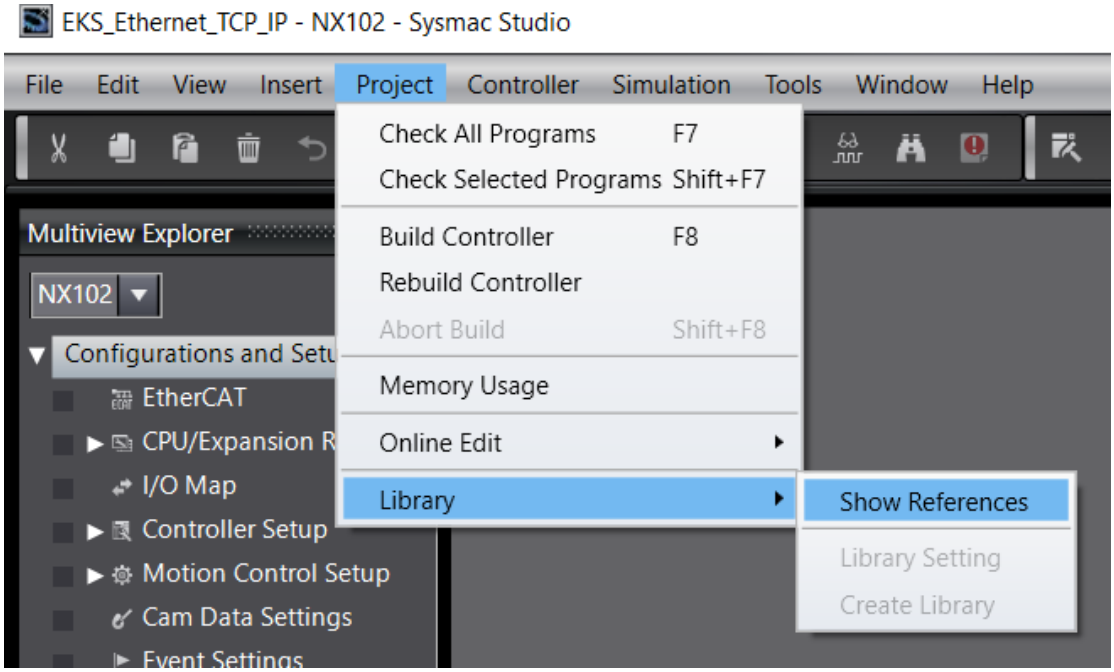


Figure 1: Library context menu

2. Click on the + symbol to add a new library.

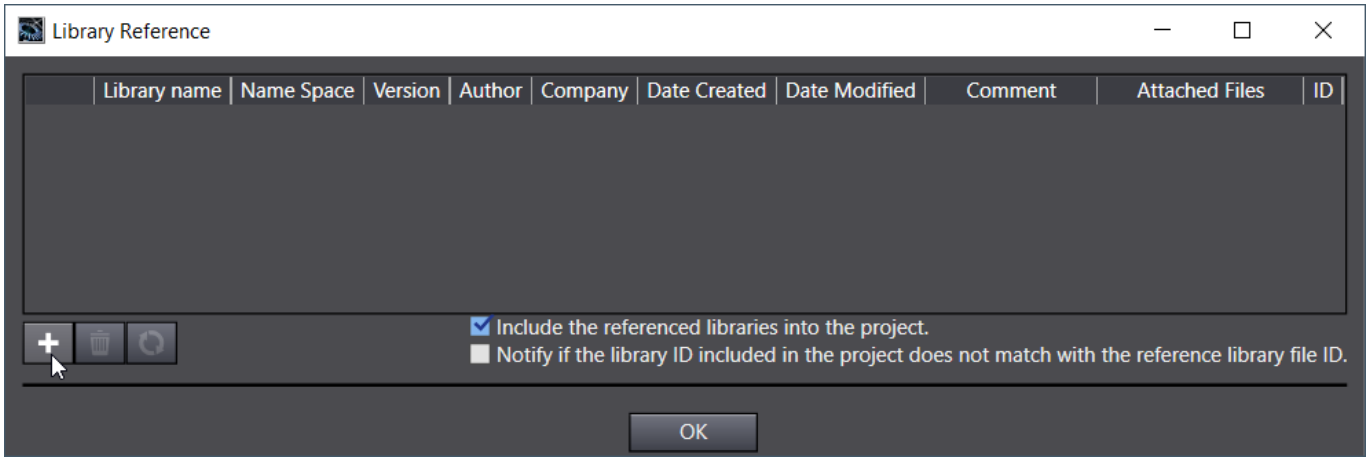


Figure 2: Adding the library

EN

3. Select the library and click on *Open*.

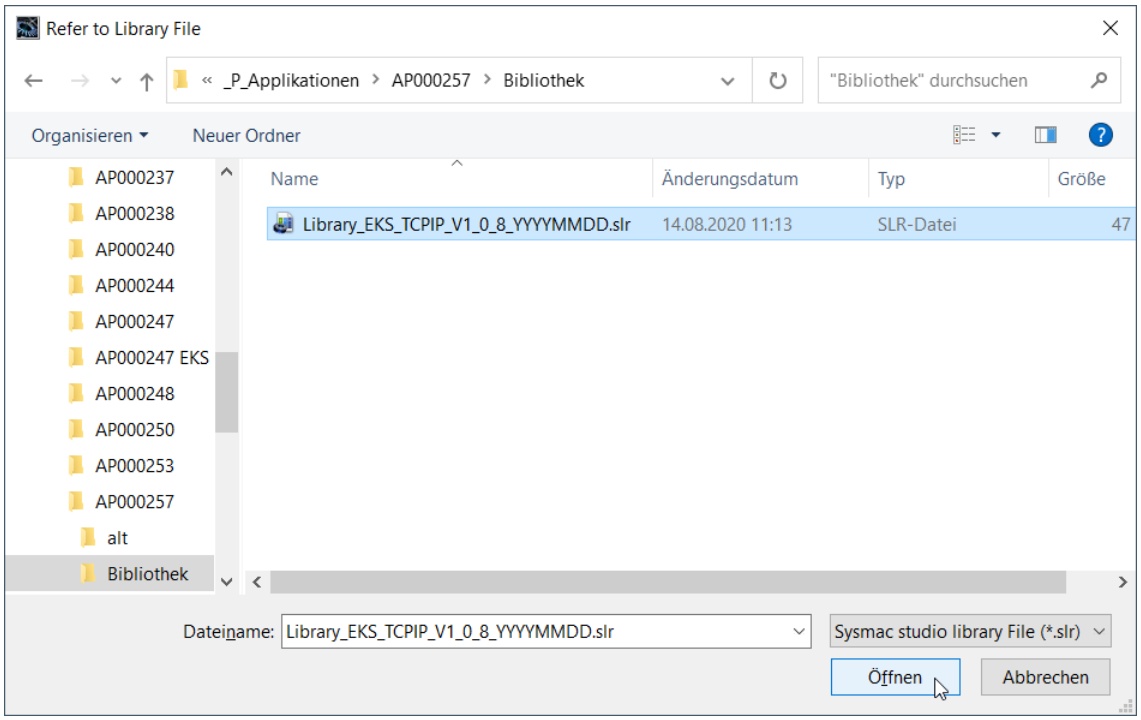


Figure 3: Selecting the library

4. Complete the library import by clicking on *OK*.

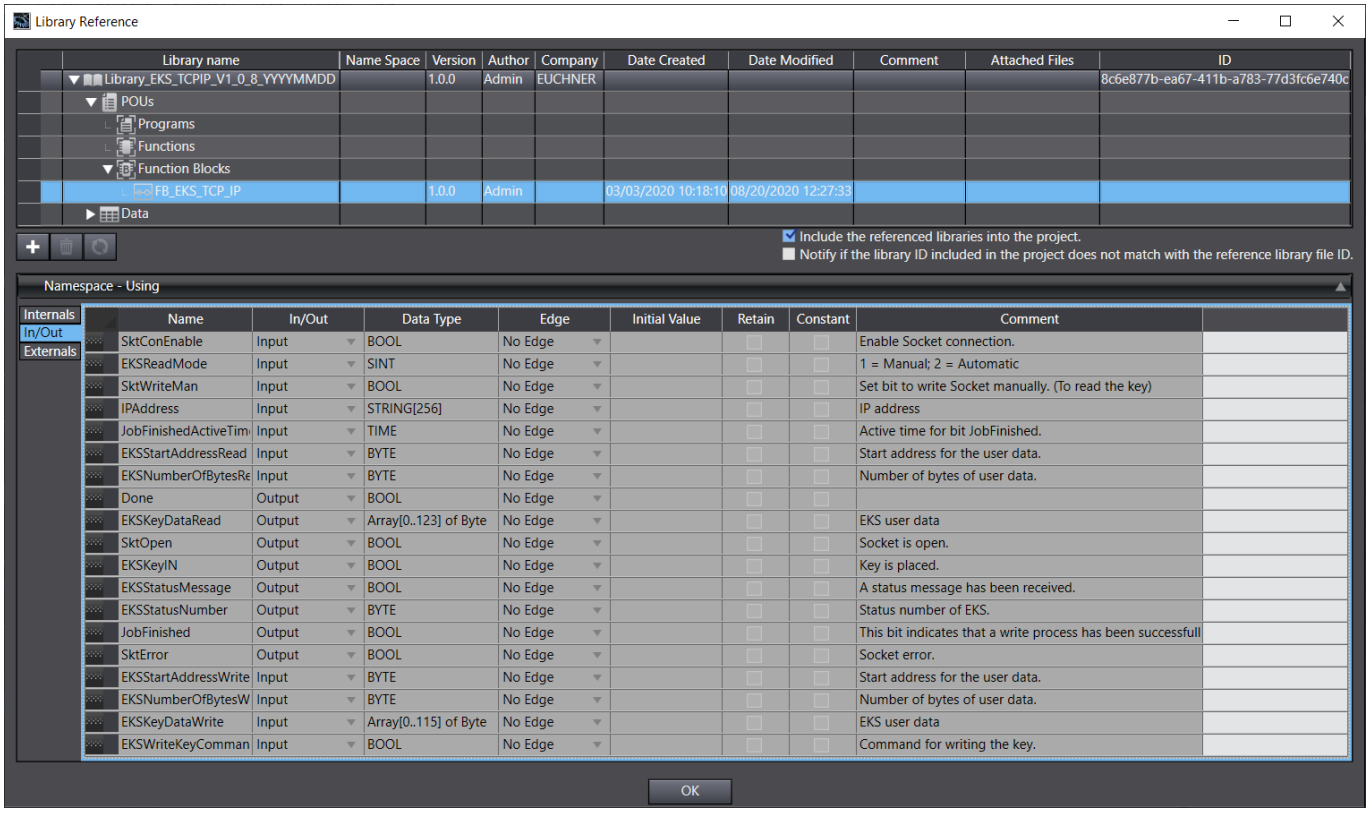
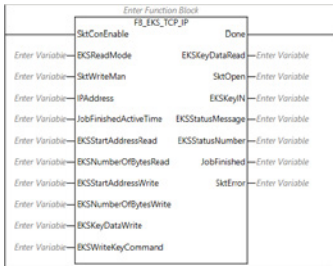


Figure 4: Completing the import

5. Integrating the block library

5.1. Library 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	SINT	Mode for the request 1= manual; 2= automatic
SktWriteMan	Input	BOOL	Trigger for reading the data in *EKSReadMode* = 1
IPAddress	Input	STRING[256]	EKS IP address
JobFinishedActiveTime	Input	TIME	Time value indicating how long the *JobFinished* bit is to remain active after the write process
EKSStartAdressRead	Input	BYTE	Start address for the Electronic-Key data to be requested
EKSNumberOfBytesRead	Input	BYTE	Amount of Electronic-Key data to be requested
EKSStartAdressWrite	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: Library variable table

5.2. Inserting the block library

1. Open a program (e.g. *Section0*) and drag the block from the *Toolbox* to a new rung using drag & drop.

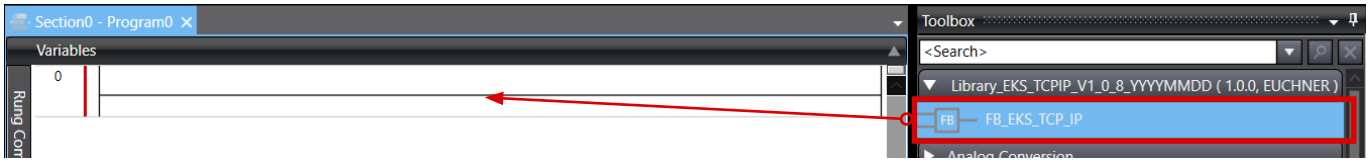


Figure 5: Adding the block to the main program

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 (*Section0*), 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 rung as a contact (*Examine On*).

Variable	Value
EKSReadMode	1 = manual mode; the Electronic-Key data are received by triggering the variable *SktWriteMan* 2 = automatic mode; the reception of the Electronic-Key data is triggered by reading the Electronic-Key status *KeyIN* (see manual)
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 *JobFinished* 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

EN

Variable	Value
EKSNumberOfBytesWrite	Variable (value from 4 to 116) → Number of bytes to be written

Table 2: Input variables

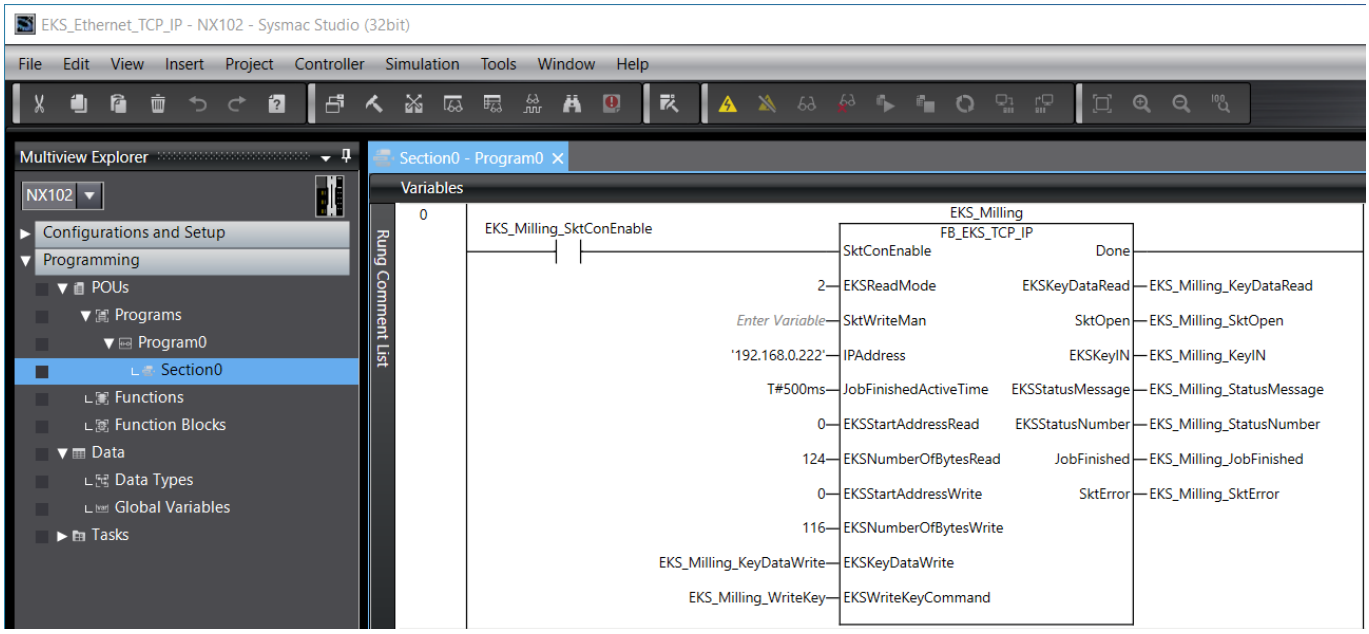


Figure 6: Declaring the instance block

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 distinction of every single Electronic-Key. It is necessary that all 8 bytes are completely evaluated for secure distinction. 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.

5. Reading and writing Electronic-Key data

5.1. Transferring program to the PLC

Transfer the program by going online  . Then open the *Controller* → *Transfer...* → *To Controller...* tab.

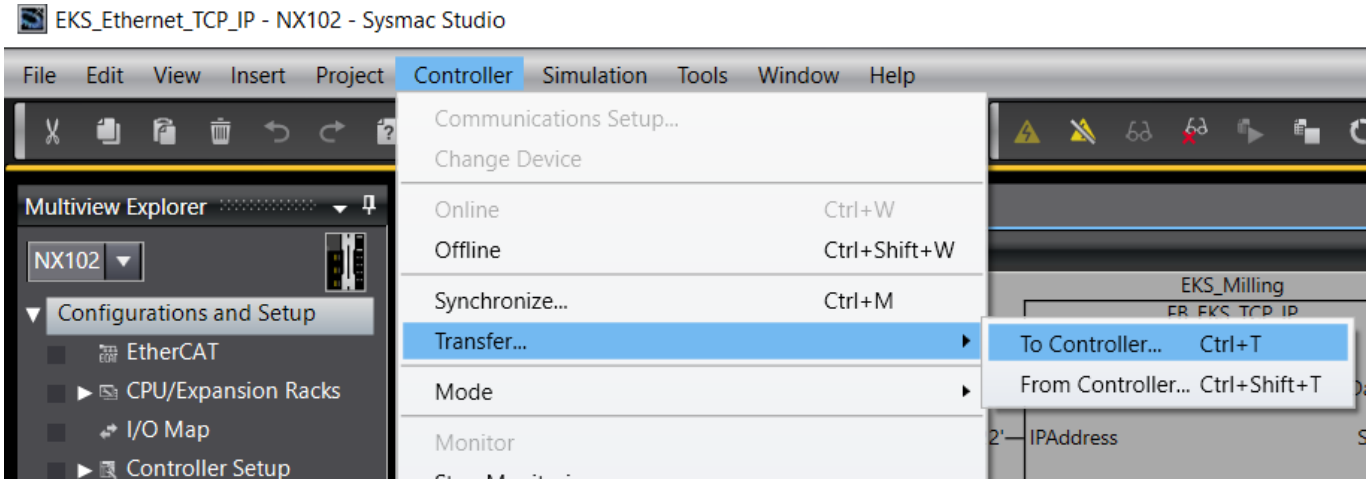


Figure 7: Transferring program to the PLC

5.2. 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*.

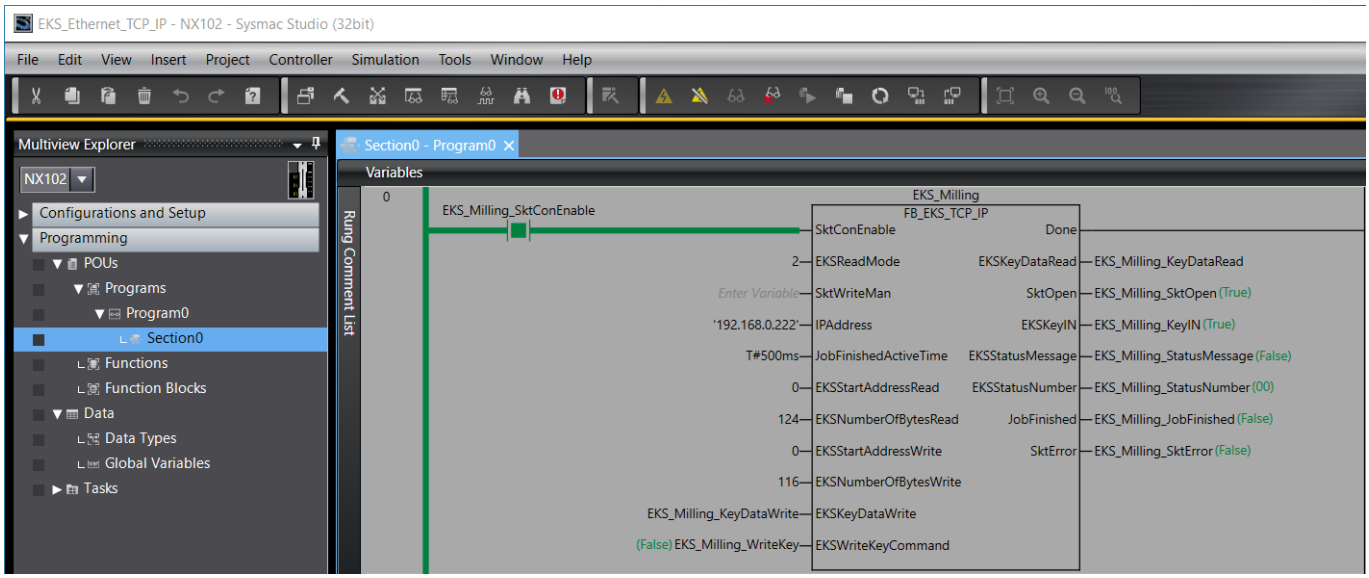
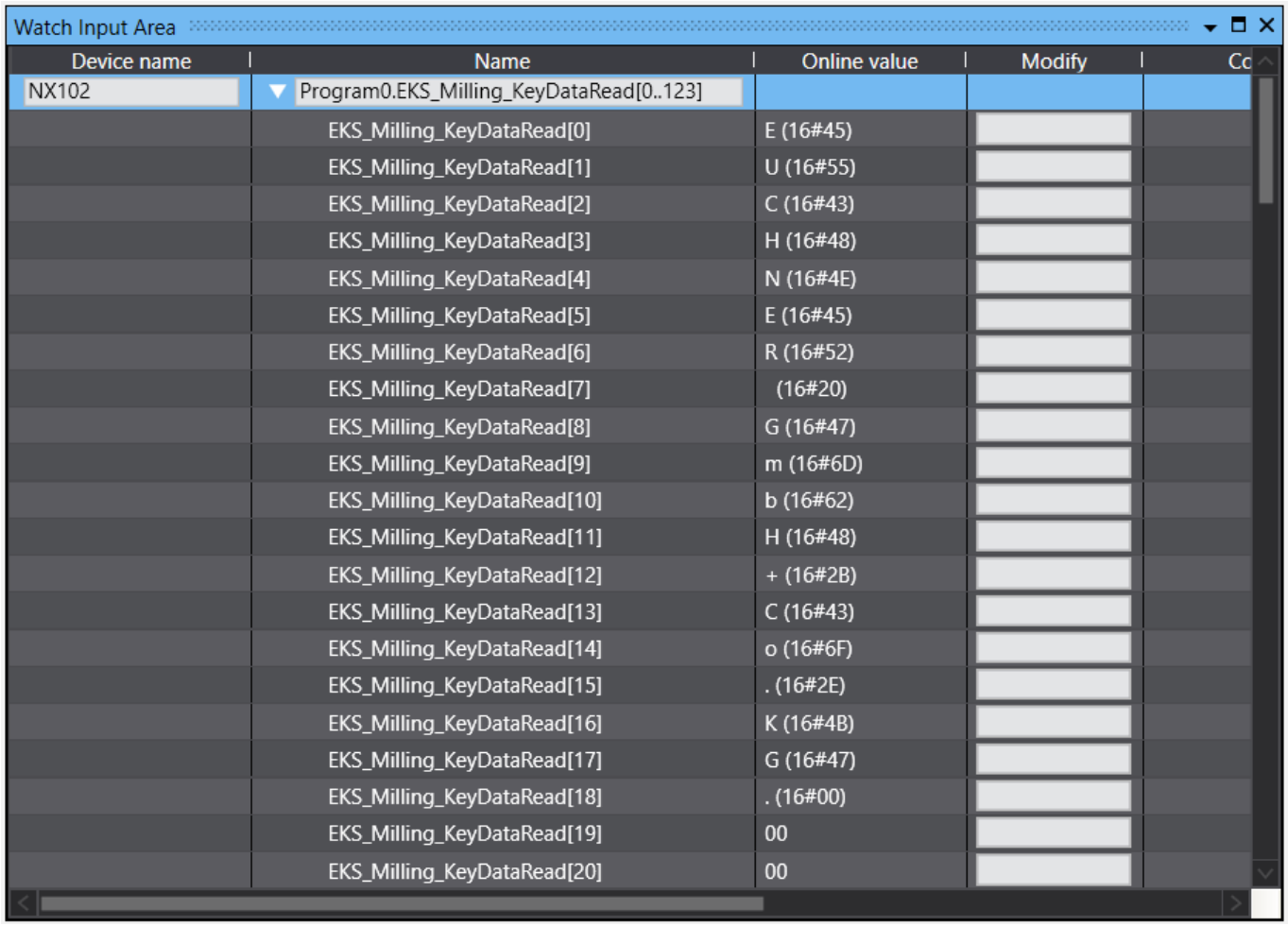


Figure 8: Instance block online

EN

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 Input Area

Device name	Name	Online value	Modify	Co
NX102	Program0.EKS_Milling_KeyDataRead[0..123]			
	EKS_Milling_KeyDataRead[0]	E (16#45)		
	EKS_Milling_KeyDataRead[1]	U (16#55)		
	EKS_Milling_KeyDataRead[2]	C (16#43)		
	EKS_Milling_KeyDataRead[3]	H (16#48)		
	EKS_Milling_KeyDataRead[4]	N (16#4E)		
	EKS_Milling_KeyDataRead[5]	E (16#45)		
	EKS_Milling_KeyDataRead[6]	R (16#52)		
	EKS_Milling_KeyDataRead[7]	(16#20)		
	EKS_Milling_KeyDataRead[8]	G (16#47)		
	EKS_Milling_KeyDataRead[9]	m (16#6D)		
	EKS_Milling_KeyDataRead[10]	b (16#62)		
	EKS_Milling_KeyDataRead[11]	H (16#48)		
	EKS_Milling_KeyDataRead[12]	+ (16#2B)		
	EKS_Milling_KeyDataRead[13]	C (16#43)		
	EKS_Milling_KeyDataRead[14]	o (16#6F)		
	EKS_Milling_KeyDataRead[15]	. (16#2E)		
	EKS_Milling_KeyDataRead[16]	K (16#4B)		
	EKS_Milling_KeyDataRead[17]	G (16#47)		
	EKS_Milling_KeyDataRead[18]	. (16#00)		
	EKS_Milling_KeyDataRead[19]	00		
	EKS_Milling_KeyDataRead[20]	00		

Figure 9: *Reading* Electronic-Key data in a watch and force table

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

A further 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 *Modify* column must be confirmed using the enter key.

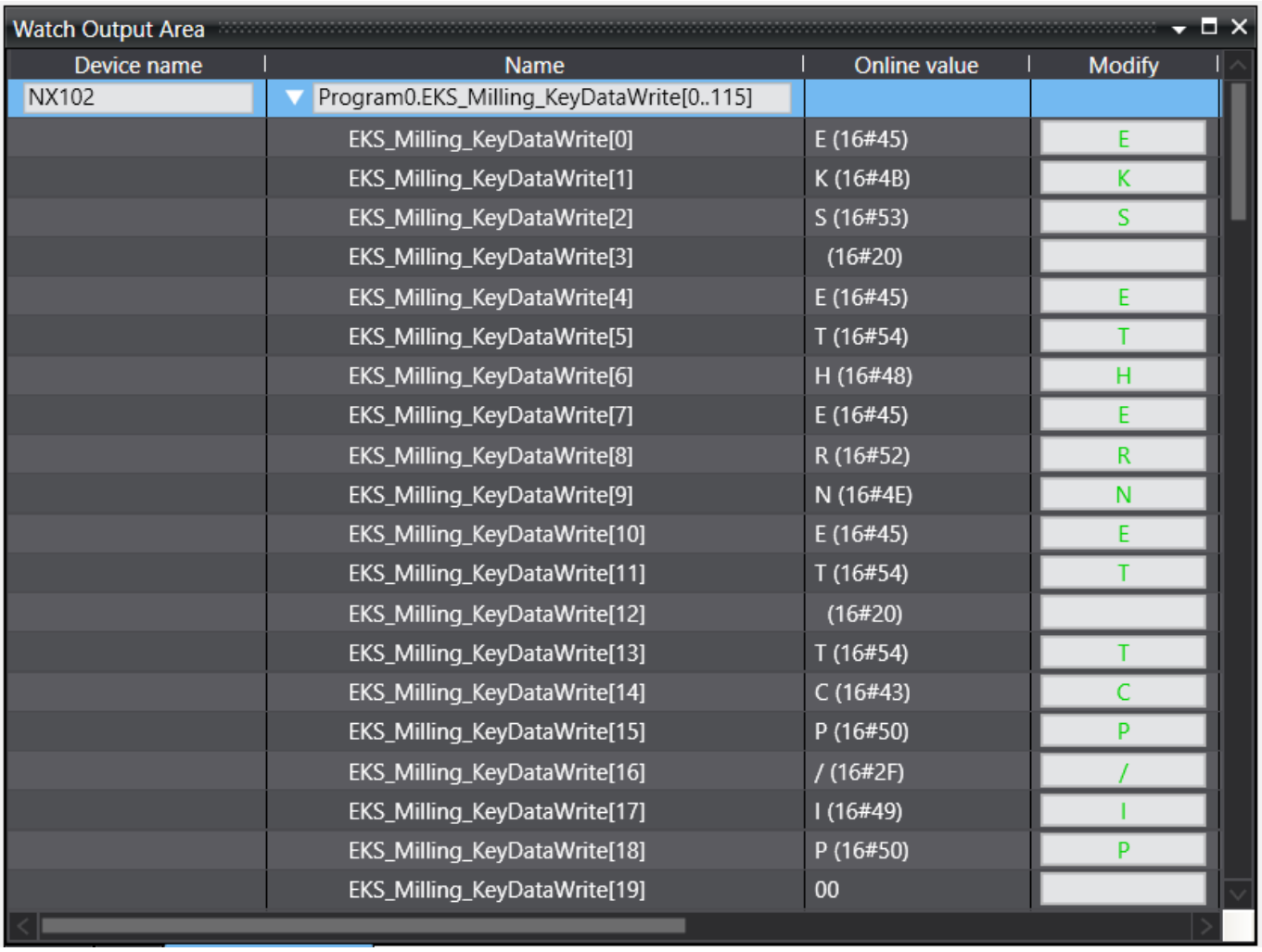


Figure 10: *Writing* Electronic-Key data in a watch and force table

EN

Running the *WriteKeyCommand* 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.

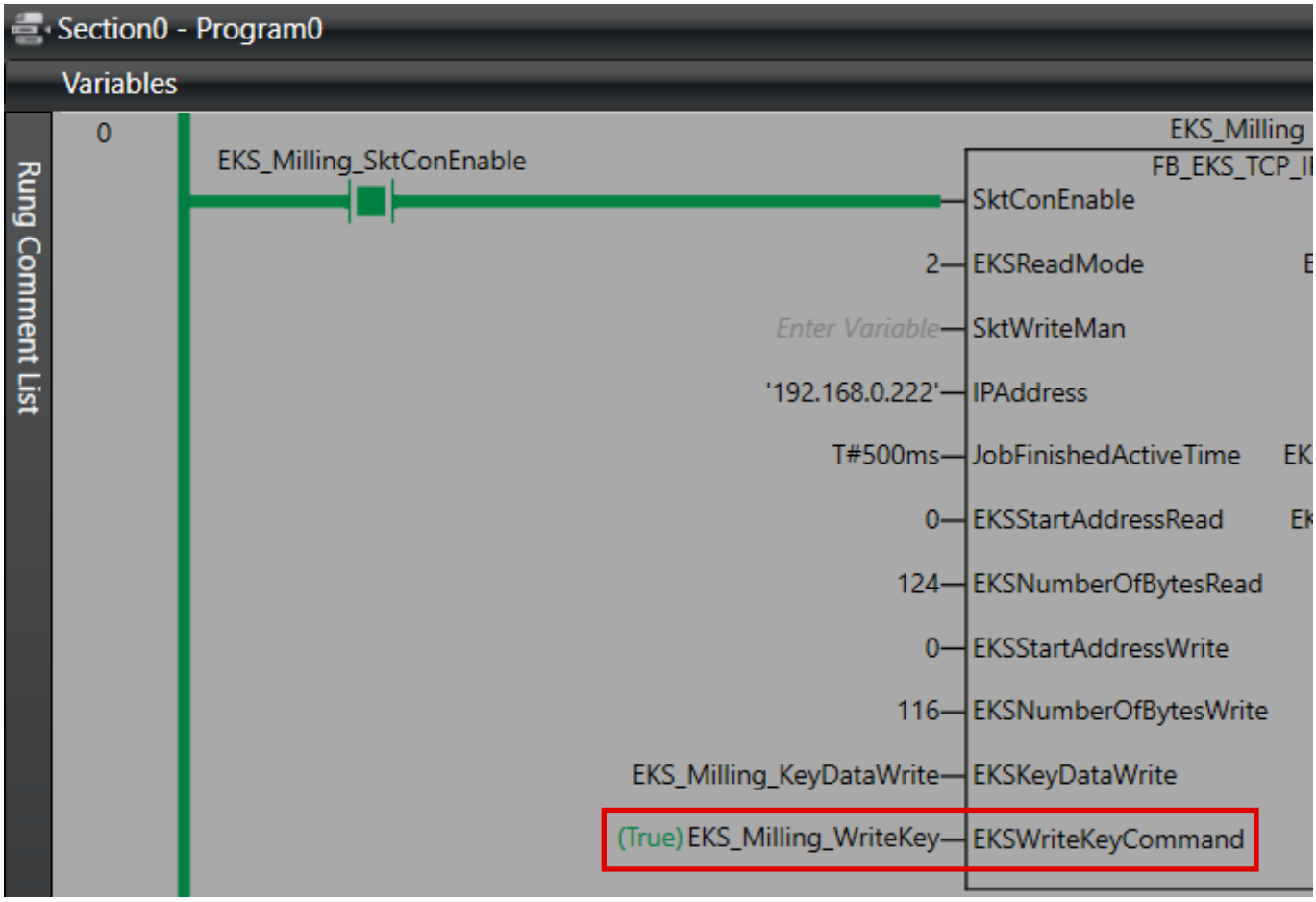


Figure 11: Command for writing Electronic-Key

Watch Input Area

Device name	Name	Online value	Modify	Co
NX102	Program0.EKS_Milling_KeyDataRead[0..123]			
	EKS_Milling_KeyDataRead[0]	E (16#45)		
	EKS_Milling_KeyDataRead[1]	K (16#4B)		
	EKS_Milling_KeyDataRead[2]	S (16#53)		
	EKS_Milling_KeyDataRead[3]	(16#20)		
	EKS_Milling_KeyDataRead[4]	E (16#45)		
	EKS_Milling_KeyDataRead[5]	T (16#54)		
	EKS_Milling_KeyDataRead[6]	H (16#48)		
	EKS_Milling_KeyDataRead[7]	E (16#45)		
	EKS_Milling_KeyDataRead[8]	R (16#52)		
	EKS_Milling_KeyDataRead[9]	N (16#4E)		
	EKS_Milling_KeyDataRead[10]	E (16#45)		
	EKS_Milling_KeyDataRead[11]	T (16#54)		
	EKS_Milling_KeyDataRead[12]	(16#20)		
	EKS_Milling_KeyDataRead[13]	T (16#54)		
	EKS_Milling_KeyDataRead[14]	C (16#43)		
	EKS_Milling_KeyDataRead[15]	P (16#50)		
	EKS_Milling_KeyDataRead[16]	/ (16#2F)		
	EKS_Milling_KeyDataRead[17]	I (16#49)		
	EKS_Milling_KeyDataRead[18]	P (16#50)		
	EKS_Milling_KeyDataRead[19]	00		
	EKS_Milling_KeyDataRead[20]	00		

Figure 12: *Reading* Electronic-Key data in a watch and force table, updated

EN

6. 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:
AP000257-01-08/20
Title:
Application EKS
Integration of EKS with TCP/IP Interface in OMRON Sysmac Studio®