

Application



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

Contents

1.	Abou	It this document	
	1.1.	Version	3
	1.2.	Scope	3
	1.3.	Target group	3
	1.4.	Supplementary documents	3
	1.5.	Notice	3
2.	Com	ponents/modules used	4
	2.1.	EUCHNER	4
	2.2.	Others	4
	2.3.	Software	4
3.	Fund	tional description	4
4.	Impo	orting the library	5
5.	Integ	grating the block library	7
	5.1.	Library variable table	7
	5.2.	Inserting the block library	7
5.	Read	ling and writing Electronic-Key data	9
	5.1.	Transferring program to the PLC	9
	5.2.	Reading contents of the Electronic-Key memory using a watch and force table	9
	5.3.	Writing contents of the Electronic-Key memory using a watch and force table	11
6.	Impo	ortant note – please observe carefully!	14

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	www
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

 (\mathbf{i})

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 <u>www.euchner.com</u>. 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 <u>www.euchner.com</u> in the area Downloads/Software/Sample files and Libraries/EKS.

EUCHNER

4. Importing the library

1. Open the Project tab and select Library \rightarrow Show References from the context menu.

EKS_Ethernet_TCP_IP - NX102 - Sysmac Studio

File Edit View Insert	Project Controller Simulation Tools Window Help
X 🗐 🖻 🗴 ⊃	Check All Programs F7 Check Selected Programs Shift+F7
Multiview Explorer	Build Controller F8
NX102 🔻	Rebuild Controller
✓ Configurations and Setu	Abort Build Shift+F8
🔚 EtherCAT	Memory Usage
▶ 🖼 CPU/Expansion R	Online Edit
	Library • Show References
►	Library Setting
Cam Data Setting	Create Library
▶ Event Settings	

Figure 1: Library context menu

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

🔜 Library Reference	—		\times
Library name Name Space Version Author Company Date Created Date Modified Comment	Attache	d Files	ID
■ Notify if the library ID included in the project does not match with the library ID included in	ne referenc	ce library	r file ID.
ОК			

Figure 2: Adding the library

EN



3. Select the library and click on Open.

Refer to Library File			×
\leftarrow \rightarrow \checkmark \uparrow \blacksquare «_P_Applikationen > AP000257 > Bibliothek	ٽ ~	"Bibliothek" durchsuchen	م
Organisieren 👻 Neuer Ordner			
AP000237 ^ Name ^	Änderungsdatum	Тур	Größe
AP000238	14.08.2020 11:13	SLR-Datei	47
AP000240			
AP000244			
AP000247 EKS			
AP000248			
AP000250			
AP000253			
AP000257			
📜 alt			
Bibliothek 🗸 <			>
Dateiname: Library_EKS_TCPIP_V1_0_8_YYYYMMDD.slr	~	Sysmac studio library File	(*.slr) ~
		Ö <u>f</u> fnen	rechen

Figure 3: Selecting the library

4. Complete the library import by clicking on OK.

Library name		Name Space	Version	Author C		Date Created	Date N	lodified	Comment	Attached Files	86698775-9967-4	ID
		1.0.0		OCHINER							110-0105-110	
Programs												
Functions												
▼												
			1.0.0 A	Admin		03/03/2020 10:18:10	08/20/20	20 12:27:33				
Data												
			<u> </u>				-	🗹 Include ti	he referenced libra	ries into the project.		
0								Notify if t	the library ID includ	led in the project doe	es not match with th	e reference lil
snace - Using		_	_	_	_		_	_				
	1							-				
Name	In/Out	Dat	а Туре	E	dge	Initial Value	Retain	Constant	<u> </u>	Comment		
SktConEnable	Input	▼ BOOL		No Edg	e v				Enable Socket cor	nnection.		
EKSReadMode	Input	▼ SINT		No Edg	e v				1 = Manual; 2 = A	lutomatic		
SktWriteMan	Input	▼ BOOL		No Edg	e v				Set bit to write Socket manually. (To read the key)			
IPAddress	Input	▼ STRING[2	256]	No Edg	e v				IP address			
JobFinishedActiveTim	Input	TIME		No Edg	e v				Active time for bit	t JobFinished.		
EKSStartAddressRead	Input			No Edg	e v				Start address for t	he user data.		
EKSNumberOfBytesR	e Input	▼ BYTE		No Edg	e v				Number of bytes	of user data.		
Done	Output	BOOL		No Edg	e v							
EKSKeyDataRead	Output	 Array[01 	23] of Byte	No Edg	e v				EKS user data			
SktOpen	Output	BOOL		No Edg	e v				Socket is open.			
EKSKeyIN	Output	▼ BOOL		No Edg	e v				Key is placed.			
EKSStatusMessage	Output	▼ BOOL		No Edg	e v				A status message	has been received.		
EKSStatusNumber	Output	■ BYIE		No Edg	e v				Status number of	EKS.		
JobFinished	Output	BOOL		No Edg	e v				This bit indicates	that a write process h	has been successfull	
SktError	Output	▼ BOOL		No Edg	e v				Socket error.			
EKSStartAddressWrite	e Input			No Edg	e v				Start address for t	the user data.		
EKSNumberOfBytesW	/ Input			No Edg	e v				Number of bytes	of user data.		
	Input	Array[01	15] of Byte	No Edg	e v				EKS user data			
EKSKeyDataWrite												



5. Integrating the block library

5.1. Library variable table

	Variable	Use	Data type	Description
Enter Function Block F8_EKS_TCP_IP	FB_EKS_TCP_IP	-	FB_EKS_TCP_IP	Instance for the FB
SktCosEnable Done Enter Variable EKSKeyDataRead -Enter Variable	SktConEnable	Input	BOOL	Activates TCP/IP socket connection
Enter Variable SktWriteMan SktOpen -Enter Variable	EKSReadMode	Input	SINT	Mode for the request 1= manual; 2= automatic
Enter Variable—JobFinishedActiveTime EKSStatusMessage—Enter Variable	SktWriteMan	Input	BOOL	Trigger for reading the data in $EKSReadMode = 1$
Enter Variable EXSStartAddressRead EXSStatusNumber Enter Variable Enter Variable ESNumberOf®ytesRead JobFinished Enter Variable	IPAddress	Input	STRING[256]	EKS IP address
Enter Variable – BCSStartAddressWrite SktEnor – Enter Variable Enter Variable – BCSNumberOfBytesWrite Forer Variable – BCSNumberOfBytesWrite	JobFinishedActiveTime	Input	TIME	Time value indicating how long the <i>JobFinished</i> bit is to remain active after the write process
Enter Variabile BCSWitteKeyCommand	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[0115] of BYTE	Data to be written
	EKSWriteKeyCommand	Input	BOOL	Command for writing Electronic-Key memory
	Done	Output	BOOL	-
	EKSKeyDataRead	Output	Array[0123] of BYTE	Reply with the user data from the EKS Electronic-Key
	SktOpen	Output	BOOL	Socket connection opened
	EKSKeylN	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 (*SectionO*), 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) \rightarrow Start address for the user data used
EKSNumberOfBytesRead	Variable (value from 1 to 124) \rightarrow Number of bytes to be read
EKSStartAddressWrite	Variable (value from 0 to 112) \rightarrow Start address for the user data to be written

EUCHNER

 Variable
 Value

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

 Table 2:
 Input variables

 Image: EKS_Ethernet_TCP_IP - NX102 - Sysmac Studio (32bit)

 File
 Edit View Insert Project Controller Simulation Tools Window Help





$\mathbf{(i)}$)
\sim	

 (\mathbf{i})

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 Electron-ic-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 \bigtriangleup . Then open the Controller \rightarrow Transfer... \rightarrow To Controller... tab.

EKS_Ethernet_TCP_IP - NX102 - Sysmac Studio

File Edit View Insert Project	Controller Simulation Tools	Window Help	
X 🖞 🛱 🖮 ちぐ 🖻	Communications Setup Change Device		🔺 🔌 63 🖗 🆡 🖬 🕻
Multiview Explorer	Online	Ctrl+W	
NX102 -	Offline	Ctrl+Shift+W	
✓ Configurations and Setup	Synchronize	Ctrl+M	EKS_Milling ER EKS TCP IP
諁 EtherCAT	Transfer	•	To Controller Ctrl+T
► 🖻 CPU/Expansion Racks	Mode	•	From Controller Ctrl+Shift+T
🛹 I/O Map	Monitor		2'— IPAddress S
🕞 🕨 🖾 Controller Setup	0 M 11 1		



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*.

KS_Ethernet_TCP_IP - NX102 - Sysmac Studio (32bit)				
File Edit View Insert Project Controller Simulation Tools Window Help				
X ឿ 値 さ さ 包 占 く ※	[2] 토 읊 봄 ❷ 武 ▲ ≫ & ☆ ኈ ኈ ㅇ 맢 맘 ፲ @ Q ♡ ♡			
Multiview Explorer	n0 - Program0 ×			
NX102 ▼ Variat ► Configurations and Setup	Ies EKS_Milling_SktConEnable EKS_Milling_SktConEnable SktConEnable Done			
Programming POUs Programs Program0 L Section0 L Functions L Function Blocks T Data L Data Types L Global Variables F Tasks	SktConEnable Done 2 EKSReadMode EKSKeyDataRead EKS_Milling_KeyDataRead Enter Variable SktWriteMan SktOpen EKS_Milling_SktOpen (True) '192.168.0.222' IPAddress EKSKeyIN EKS_Milling_KeyIN (True) 192.168.0.222' IPAddress EKSStatusMessage EKS_Milling_StatusMessage (False) 0 EKSStartAddressRead EKSStatusNumber EKS_Milling_StatusNumber (00) 124- EKSNumberOfBytesRead JobFinished EKS_Milling_StatusNumber (00) 124- EKSNumberOfBytesRead JobFinished EKS_Milling_SktError (False) 0- EKSStartAddressWrite SktError EKS_Milling_SktError (False) 116 EKSNumberOfBytesWrite EKS_Milling_KeyDataWrite			
	(False) EKS_Milling_WriteKey— EKSWriteKeyCommand			



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	l Modify	Cc 🗠
NX102	Program0.EKS_Milling_KeyDataRead[0123]			
	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		\sim
<		-		>

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.

Watch Output Area				×
Device name	Name	Online value	Modify	$ $ \wedge
NX102	Program0.EKS_Milling_KeyDataWrite[0115]			
	EKS_Milling_KeyDataWrite[0]	E (16#45)	E	
	EKS_Milling_KeyDataWrite[1]	K (16#4B)	K	
	EKS_Milling_KeyDataWrite[2]	S (16#53)	S	
	EKS_Milling_KeyDataWrite[3]	(16#20)		
	EKS_Milling_KeyDataWrite[4]	E (16#45)	E	
	EKS_Milling_KeyDataWrite[5]	T (16#54)	Т	
	EKS_Milling_KeyDataWrite[6]	H (16#48)	Н	
	EKS_Milling_KeyDataWrite[7]	E (16#45)	E	
	EKS_Milling_KeyDataWrite[8]	R (16#52)	R	
	EKS_Milling_KeyDataWrite[9]	N (16#4E)	N	
	EKS_Milling_KeyDataWrite[10]	E (16#45)	E	
	EKS_Milling_KeyDataWrite[11]	T (16#54)	Т	
	EKS_Milling_KeyDataWrite[12]	(16#20)		
	EKS_Milling_KeyDataWrite[13]	T (16#54)	Т	
	EKS_Milling_KeyDataWrite[14]	C (16#43)	С	
	EKS_Milling_KeyDataWrite[15]	P (16#50)	Р	
	EKS_Milling_KeyDataWrite[16]	/ (16#2F)	/	
	EKS_Milling_KeyDataWrite[17]	I (16#49)	1	
	EKS_Milling_KeyDataWrite[18]	P (16#50)	Р	
	EKS_Milling_KeyDataWrite[19]	00		
<				

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

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.

-	Section0 -	Program0		
	Variables			
Rung	0	0 EKS_Milling_SktConEnable		EKS_Milling FB_EKS_TCP_II SktConEnable
l Com			2—	EKSReadMode E
ment			Enter Variable-	SktWriteMan
List			'192.168.0.222' 	IPAddress
			T#500ms—	JobFinishedActiveTime EK
			0—	EKSStartAddressRead Ek
			124—	EKSNumberOfBytesRead
			0—	EKSStartAddressWrite
			116—	EKSNumberOfBytesWrite
			EKS_Milling_KeyDataWrite—	EKSKeyDataWrite
			(True) EKS_Milling_WriteKey—	EKSWriteKeyCommand

Figure 11: Command for writing Electronic-Key

EUCHNER

Watch Input Area 🗸 🗖 🕽				🛨 🛨 🗙
Device name	Name	Online value	Modify	Cc 🗠
NX102	Program0.EKS_Milling_KeyDataRead[0123]			
	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		\sim
<				>

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

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®

Copyright: © EUCHNER GmbH + Co. KG, 08/2020

Subject to technical modifications; no responsibility is accepted for the accuracy of this information. $% \label{eq:sub_constraint}$