User Guide English

ProfiNet App

KATHREIN

Content

1. About This Guide

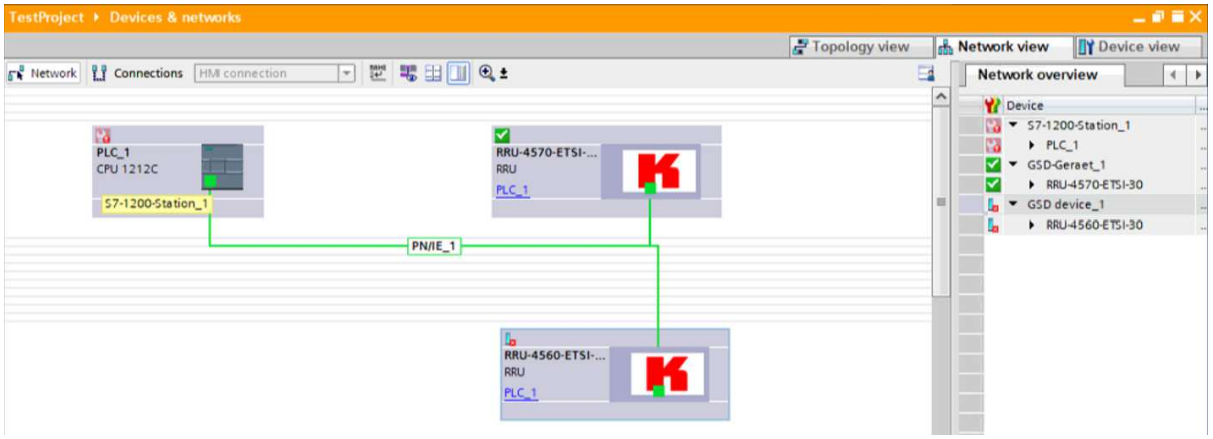
This document describes how to configure the ProfiNet app.

2. Explanation of Symbols

Since the app has no display in the ReaderStart SW, there are also no symbols

3. Function

The ProfiNet App connects the Kathrein readers of the X5xx series with a Simatic S7. The ProfiNet App runs on the Linux part of the reader and communicates with the Simatic S7 controller. For the integration of the reader please use the GSD file released by Kathrein.



4. Specification ProfiNet-IO Protocol

4.1. Slot setup

The profinet-io app provides four slots for Status and control data, GPIO data (card 0 and 1), KRAI LEDs (port 1 to 4) and Reader integrated LEDs.

Cyclic IO data is used to control these slots and present the current data.

Slot	Subslot	Length (byte)	Direction	Function
1	1	4	In/Out	Status and control data
2	1	1	In/Out	GPIO data (card 0)
	2	1	In/Out	GPIO data (card 1)
3	1	1	Out	KRAI LEDs (port 1)
	2	1	Out	KRAI LEDs (port 2)
	3	1	Out	KRAI LEDs (port 3)
	4	1	Out	KRAI LEDs (port 4)
4	1	1	Out	Reader integrated LEDs
	2	1	Out	Reader integrated LEDs

4.2. Slot data

4.2.1. Status and control data – Slot 1, Subslot 1, 4 Byte, In/Out

IN		
Bit	Function	Comment
0	Buys	
1	Done	
2 ... 29	Reserved	
30	Command cancelled	Indicates, that the current command was deleted from the command queue
31	All commands cancelled	Indicates, that all commands were deleted from the command queue

Out		
Bit	Function	Comment
0 ... 29	Reserved	
30	Command cancelled	Indicates, that the current command should be deleted from the command queue
31	All commands cancelled	Indicates, that all commands should be deleted from the command queue

4.2.2. GPIO data – Slot 2, Subslot 1 (card 0) and 2 (card 1), 1 Byte, In/Out

IN		
Bit	Function	Comment
0	GPIO data input 0 (card X)	If set, input 0 is logical high
1	GPIO data input 1 (card X)	If set, input 1 is logical high
2	GPIO data input 2 (card X)	If set, input 2 is logical high
3	GPIO data input 3 (card X)	If set, input 3 is logical high
4 ...7	Reserved	

Out		
Bit	Function	Comment
0	GPIO data output 0 (card X)	Set, to enable output 0
1	GPIO data output 1 (card X)	Set, to enable output 1
2	GPIO data output 2 (card X)	Set, to enable output 2
3	GPIO data output 3 (card X)	Set, to enable output 3
4 ...7	Reserved	

4.2.3. KRAI LEDs – Slot 3, Subslot 1 (port 1) to 4 (port 4), 1 Byte, Out

Out		
Bit	Function	Comment
0	GPIO KRAI LED 0 (card X)	Set, to enable LED 0
1	GPIO KRAI LED 1 (card X)	Set, to enable LED 1
2	GPIO KRAI LED 2 (card X)	Set, to enable LED 2
3	Status LED 4	Set, to enable LED 3
4 ...7	Reserved	

4.2.4. Reader integrated LEDs – Slot 4, Subslot 1 to 2, 1 Byte, Out

Out	(Subslot 1)	
Bit	Function	Comment
0	Status LED 1	Set, to enable LED 1
1	Status LED 2	Set, to enable LED 2
2	Status LED 3	Set, to enable LED 3
3	Status LED 4	Set, to enable LED 4
4 ...7	Reserved	

Out	(Subslot 2)	
Bit	Function	Comment
0	Status LED 5	Set, to enable LED 5
1	Status LED 6	Set, to enable LED 6
2	Status LED 7	Set, to enable LED 7
3	Status LED 8	Set, to enable LED 8
4	Status LED 9	Set, to enable LED 9
5	Status LED 10	Set, to enable LED 10
6	Status LED 11	Set, to enable LED 11
7	Status LED 12	Set, to enable LED 12

4.3. Acyclic commands

To read or write tags and to configure the reader, acyclic commands are used. Each command consists of a command header and the command data. If the command data exceeds a maximum number of bytes, it must be split into several blocks. For commands sent to the reader, the maximal data length is 256 byte per block, for commands received from the reader, it is 1024 byte. All fields are transferred MSB first. The command structure is shown below.

Field	Block length	Block counter	Status	Reserved	ID	Command data
Bytes	2	2	1	2	1	n

First element is the block length, which is the size (in byte) of this block excluding the two byte of the block length itself.

The block counter is needed, if the command is split into several parts. For the first block, it is set to ([number of parts] – 1) and is decreased for every following block. Thus, if a command is not split, this field is set to zero.

The bits of the status field are described below.

Bit	Function
0 ... 6	Reserved
7	Indicates first block

The ID defines the command the reader should perform.

ID	Function
0x01	SyncBulkGetEPCs
0x02	SyncReadData
0x03	SyncWriteEPC
0x04	SyncWriteData
0x05	GetParameterByConfigId
0x06	SetParameterByConfigId
0x07	

After the ID, command specific data is following. The format and length of this data is described in the chapters below.

4.3.1. Tag data

Every command that sends information in the command answer uses a common format to represent a tag. The tag data format is shown below. It has a fixed size of 76 bytes. Please note that some parameters are only provided, if the "Extended Result Flag" is configured in the reader (see Communication protocol (KBRP) for details).

Parameter	Length (byte)	Comment
Antenna	1	Antenna property
RSSI	1	Raw value without unit
Reserved	2	
Timestamp	4	Unix time in seconds
Frequency	3	Frequency in kHz
Tag phase	2	Tag phase in 1/10 degree steps
EPC length	1	[EPC] length in bytes
EPC	62	Only the first [EPC length] bytes are valid

4.3.2. SyncBulkGetEPCs

4.3.2.1. Request

Parameter	Length (byte)	Comment
Time	2	Time to read tags in ms
Reserved	1	

4.3.2.2. Answer

Parameter	Length (byte)	Comment
Result flag	1	see Communication protocol (KBRP)
Tag count	2	Number of tags following
Tag data	N * 76	N; Number of tags, see Tag data

4.3.3. SyncReadData

4.3.3.1. Request

Parameter	Length (byte)	Comment
Single/specific	1	If > 0, [EPC] is used to target specific tag
Until end of bank	1	If > 0, try to read up to [Memory length] bytes
Reserved	1	
EPC length	1	[EPC] length in bytes
EPC	62	Only the first [EPC length] bytes are valid
Password	4	Access password
Memory bank	1	0: RFU, 1: EPC, 2: TID, 3: USR
Memory pointer	4	Byte address
Memory length	4	Number of bytes to read
Reserved	1	

4.3.3.2. Answer

Parameter	Length (byte)	Comment
Result flag	1	See Communication protocol (KBRP)
Tag data	76	See Tag data
Tag error code	1	See Communication protocol (KBRP)
Data length	4	[Data] length in bytes
Data	n	

4.3.4. SyncWriteEPC

4.3.4.1. Request

Parameter	Length (byte)	Comment
Single/specific	1	If > 0, [Old EPC] is used to target specific tag
Reserved	1	
Old EPC length	1	[Old EPC] length in bytes
Old EPC	62	Only the first [Old EPC length] bytes are valid
Password	4	Access password
Reserved	1	
New EPC length	1	[New EPC] length in bytes
New EPC	62	Only the first [New EPC length] bytes are valid

4.3.4.2. Answer

Parameter	Length (byte)	Comment
Result flag	1	See Communication protocol (KBRP)
Tag data	76	See Tag data
Tag error code	1	See Communication protocol (KBRP)

4.3.5. SyncWriteData

4.3.5.1. Request

Parameter	Length (byte)	Comment
Single/specific	1	If > 0, [EPC] is used to target specific tag
Reserved	1	
EPC length	1	[EPC] length in bytes
EPC	62	Only the first [EPC length] bytes are valid
Password	4	Access password
Memory bank	1	0: RFU, 1: EPC, 2: TID, 3: USR
Memory pointer	4	Byte address
Reserved	1	
Data length	4	[Data] length in byte
Data	n	

4.3.5.2. Answer

Parameter	Length (byte)	Comment
Result flag	1	See Communication protocol (KBRP)
Tag data	76	See Tag data
Tag error code	1	See Communication protocol (KBRP)

4.3.6. GetParameterByConfigId

4.3.6.1. Request

Parameter	Length (byte)	Comment
Config ID	4	

4.3.6.2. Answer

Parameter	Length (byte)	Comment
Result flag	1	See Communication protocol (KBRP)
Config ID	4	
Value	4	

4.3.7. SetParameterByConfigId

4.3.7.1. Request

Parameter	Length (byte)	Comment
Config ID	4	
Value	4	

4.3.7.2. Answer

Parameter	Length (byte)	Comment
Result flag	1	See Communication protocol (KBRP)
Config ID	4	

Electronic equipment is not domestic waste – in accordance with directive 2002/96/EC OF THE EUROPEAN PARLIAMENT AND THE COUNCIL dated 27th January 2003 concerning used electrical and electronic appliances, it must be disposed of properly. At the end of its service life, take this unit for disposal at a designated public collection point.