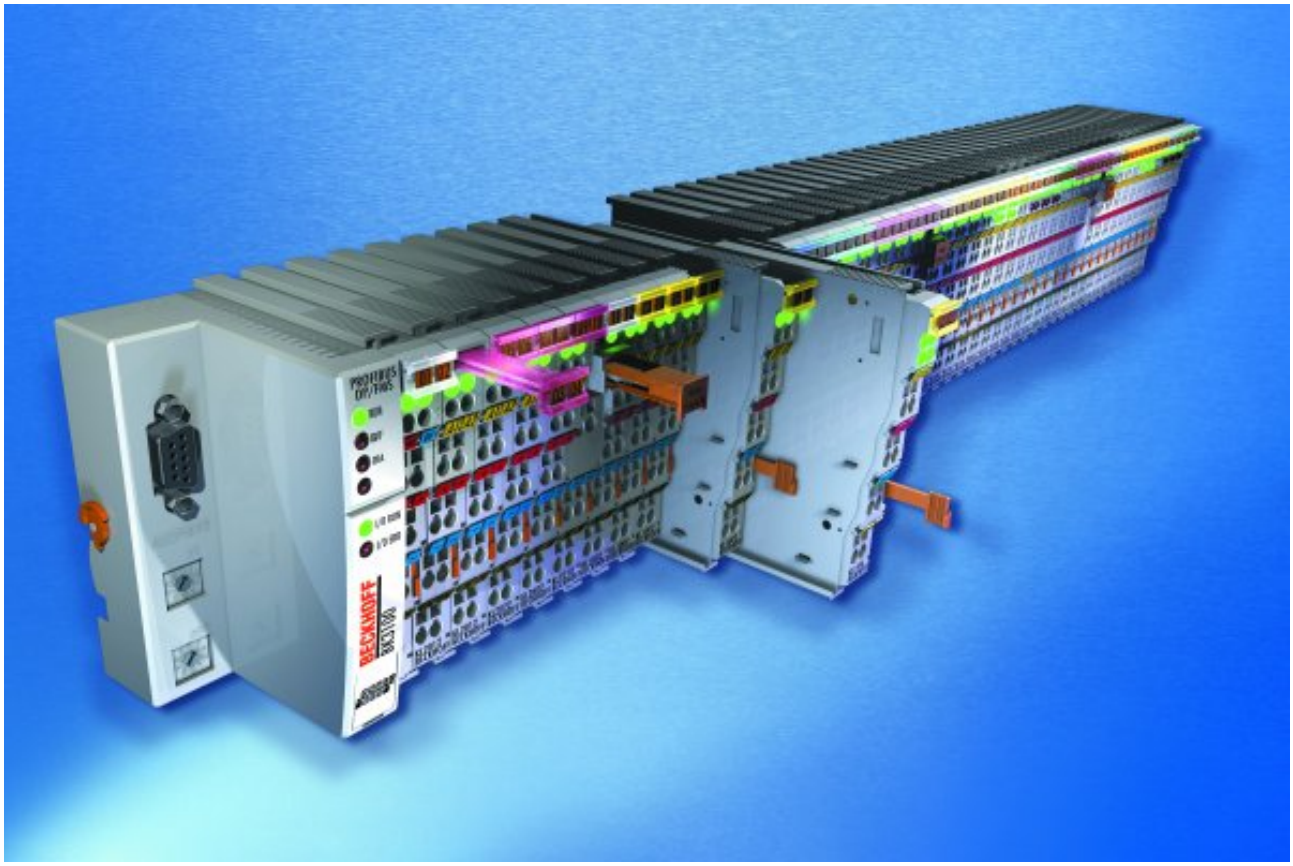

Documentation

KL6224

IO-Link Terminal

Version: 2.0.0
Date: 2017-05-17

BECKHOFF

Table of contents

1 Foreword

1.1 Notes on the documentation

Intended audience

This description is only intended for the use of trained specialists in control and automation engineering who are familiar with the applicable national standards.
It is essential that the documentation and the following notes and explanations are followed when installing and commissioning these components.
It is the duty of the technical personnel to use the documentation published at the respective time of each installation and commissioning.

The responsible staff must ensure that the application or use of the products described satisfy all the requirements for safety, including all the relevant laws, regulations, guidelines and standards.

Disclaimer

The documentation has been prepared with care. The products described are, however, constantly under development.

We reserve the right to revise and change the documentation at any time and without prior announcement.

No claims for the modification of products that have already been supplied may be made on the basis of the data, diagrams and descriptions in this documentation.

Trademarks

Beckhoff®, TwinCAT®, EtherCAT®, Safety over EtherCAT®, TwinSAFE®, XFC® and XTS® are registered trademarks of and licensed by Beckhoff Automation GmbH.
Other designations used in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owners.

Patent Pending

The EtherCAT Technology is covered, including but not limited to the following patent applications and patents: EP1590927, EP1789857, DE102004044764, DE102007017835 with corresponding applications or registrations in various other countries.

The TwinCAT Technology is covered, including but not limited to the following patent applications and patents: EP0851348, US6167425 with corresponding applications or registrations in various other countries.

EtherCAT® is registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany

Copyright

1.2 Safety instructions

Safety regulations

Please note the following safety instructions and explanations!
Product-specific safety instructions can be found on following pages or in the areas mounting, wiring, commissioning etc.

Exclusion of liability

All the components are supplied in particular hardware and software configurations appropriate for the application. Modifications to hardware or software configurations other than those described in the documentation are not permitted, and nullify the liability of Beckhoff Automation GmbH & Co. KG.

Personnel qualification

This description is only intended for trained specialists in control, automation and drive engineering who are familiar with the applicable national standards.

Description of symbols

In this documentation the following symbols are used with an accompanying safety instruction or note. The safety instructions must be read carefully and followed without fail!

 **DANGER**	**Serious risk of injury!** Failure to follow the safety instructions associated with this symbol directly endangers the life and health of persons.
 **WARNING**	**Risk of injury!** Failure to follow the safety instructions associated with this symbol endangers the life and health of persons.
 **CAUTION**	**Personal injuries!** Failure to follow the safety instructions associated with this symbol can lead to injuries to persons.
 **Attention**	**Damage to the environment or devices** Failure to follow the instructions associated with this symbol can lead to damage to the environment or equipment.
 **Note**	**Tip or pointer** This symbol indicates information that contributes to better understanding.

1.3 Documentation issue status

Version	Comment
2.0.0	• Migration
1.0.0	• Product overview updated • Technical data updated • Description of the KS2000 configuration software expanded • Register description updated
0.5	• Preliminary version

Firmware and hardware versions

Documentation version	KL6224	
	Firmware	Hardware
1.0.0	03	02
0.5	00	01

The firmware and hardware versions (delivery state) can be taken from the serial number printed on the side of the terminal.

The KL6224 supports the IO-Link specification 1.1 from firmware version 02.

Syntax of the serial number

Structure of the serial number: WW YY FF HH

WW - week of production (calendar week)
YY - year of production
FF - firmware version
HH - hardware version

Example with ser. no.: 02 13 00 01:

02 - week of production 02
13 - year of production 2013
00 - firmware version 00
01 - hardware version 01

2 Product overview

2.1 Introduction

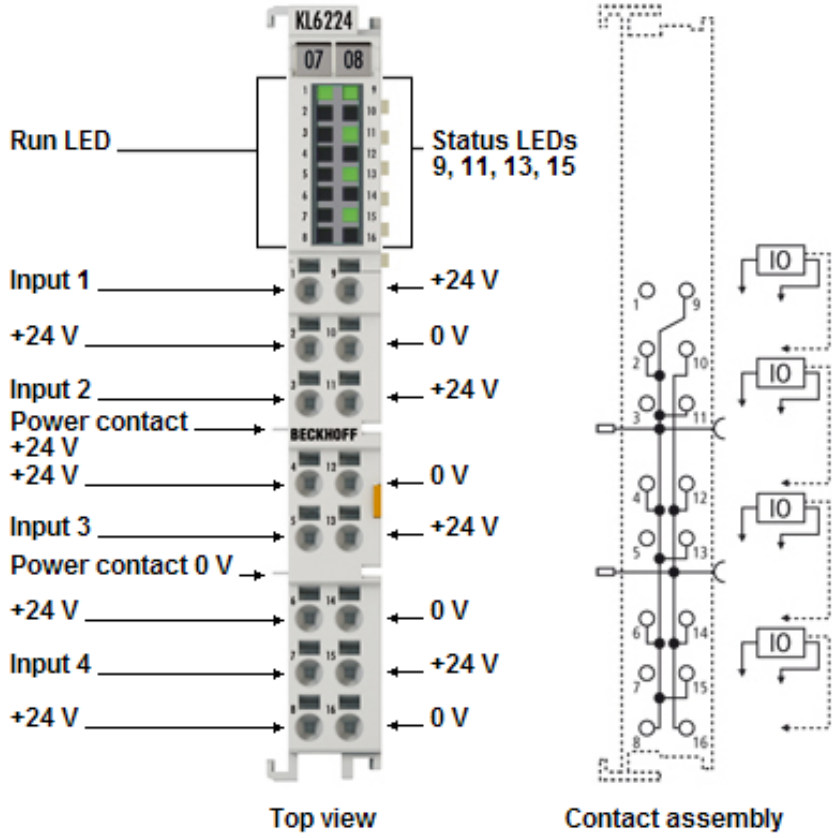

Fig. 1: KL6224

IO-Link

The KL6224 IO-Link terminal enables connection of up to four IO-Link devices. These can be actuators, sensors or a combination of both. A point-to-point connection is used between the terminal and the device.

IO-Link is designed as an intelligent link between the fieldbus level and the sensor, wherein parameterization information can be exchanged bidirectionally via the IO-Link connection. The IO-Link devices can be parameterized via the KS2000 configuration software or from the PLC via register communication.

By default the KL6224 operates as a four-channel input terminal (24 V_{DC}), which communicates with connected IO-Link devices, parameterizes them and adjusts their operating mode as required.

Thanks to the integration of the KL6224 in the HD housing with 16 connection points, each IO-Link device can be operated in 3-wire mode. The direct plug-in technique enables toolless construction.

2.2 Technical data

Technical data	KL6224
Technology	IO-Link
Number of channels	4
IO-Link interfaces	4
Field voltage	24 V_{DC} (via power contacts)
Connection	3-wire, HD terminal [▶ 16]
Data transfer rates	4.8 kBaud, 38.4 kBaud and 230.4 kBaud
Cable length	max. 20 m
Power supply	via K-Bus and power contacts
Supply current for devices	500 mA per device
Current consumption from K-bus	typ. 85 mA
Current consumption from power contacts	not specified
Nominal voltage	24 V_{DC} (-15 %/+20 %)
Electrical isolation	500 V (IO-Link / K-bus)
Weight	app. 60 g
K-bus bit width	depends on the process image
Configuration	via fieldbus parameter interface, DP-V1 or Bus Coupler via Configuration software KS2000
Permissible ambient temperature during operation	0°C ... +55°C
Permissible ambient temperature during storage	-25°C ... +85°C
Permissible relative humidity	95 %, no condensation
Mounting [▶ 12]	on 35 mm mounting rail conforms to EN 60715
Vibration / shock resistance	conforms to EN 60068-2-6 / EN 60068-2-27 see also installation instructions [▶ 14] for enhanced mechanical load capacity
EMC resistance burst / ESD	conforms to EN 61000-6-2 / EN 61000-6-4
Protection class	IP20
Installation position	Variable
Approval	CE, cULus

2.3 IO-Link - Basics

IO-Link represents a communication system for the connection of intelligent sensors and actuators to an automation system in the IEC 61131-9 standard under the designation "Single-drop digital communication interface for small sensors and actuators" (SDCI).

Both the electrical connection data and the communication protocol are standardized and summarized in the IO-Link Spec.

Note	**IO-Link specification** The development of the EL6224 was subject to the IO-Link specification 1.1. At the time of the preparation of this documentation, the IO-Link specification is entering the IEC standardization and will be adopted in extended form as IEC 61131-9. The new designation SDCI will be introduced at the same time. As a member of the respective committee, Beckhoff supports the development of IO-Link and reflects changes to the specification in its products.

An IO-Link system consists of an IO-Link master and one or more IO-Link devices, i.e. sensors or actuators. The IO-Link master provides the interface to the higher-level controller and controls communication with the connected IO-Link devices. The Beckhoff EL6224 IO-Link Master Terminal has four IO-Link ports. One IO-Link device can be connected to each of them. Therefore, IO-Link is not a fieldbus, but a peer-to-peer connection (see Fig. *Peer-to-peer communication IO-Link*).

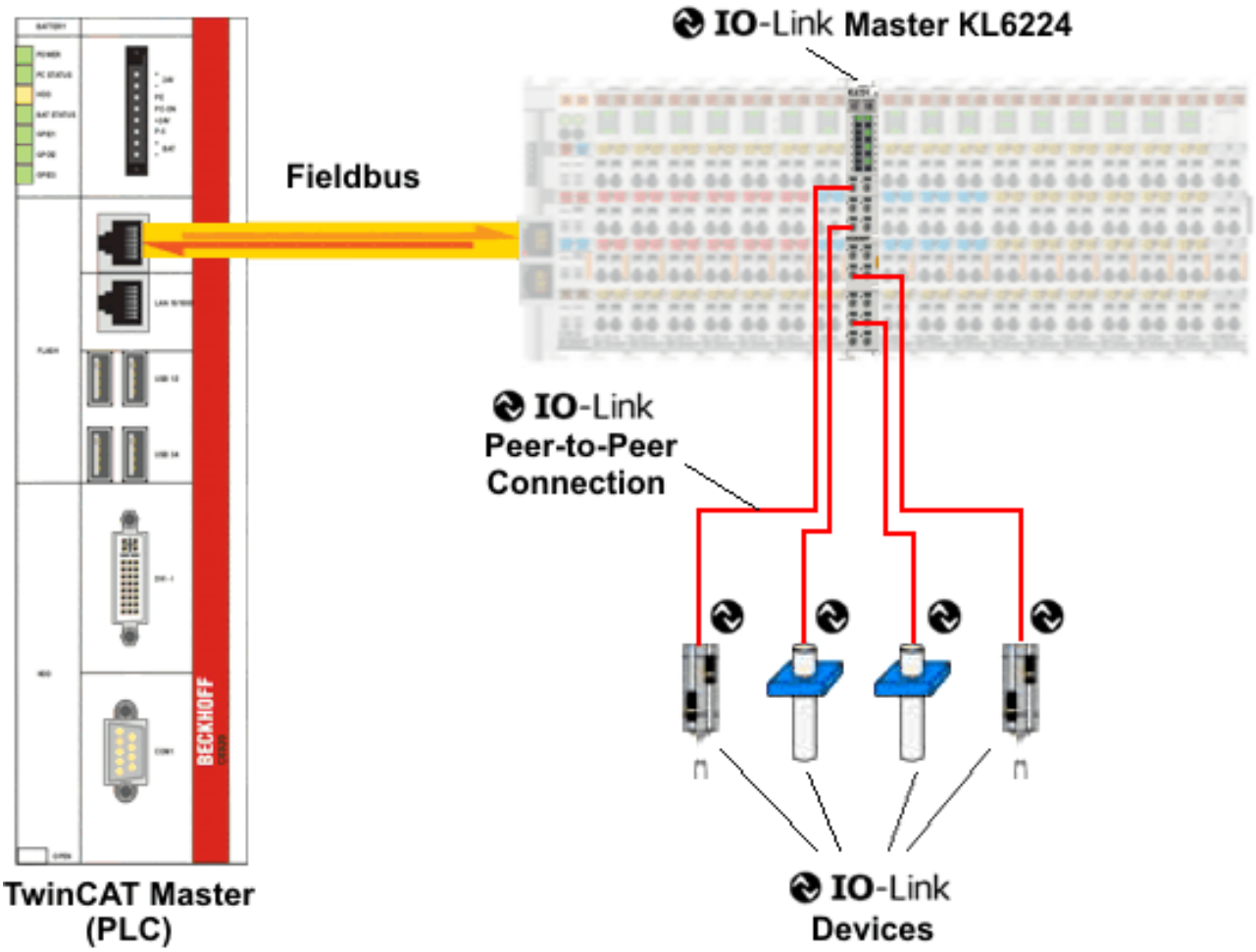

Fig. 2: Peer-to-peer communication IO-Link

The connected IO-Link devices have individual parameter information in the form of an IO device description (IODD), which is set with the KS2000 configuration software (see chapter KS2000 - Settings [▶ 26]).

Parameter data exchange

An intelligent IO-Link device can support parameterization through SPDUs (service protocol data units). The PLC must explicitly query or, when marked as such, send these acyclic service data.

Note	**SPDU access** TwinCAT currently only supports access via ADS.

The so-called SPDU index is used to address the corresponding parameter. The following ranges are available.

Index range	Name
0x00...0x0F	System
0x10...0x1F	Identification
0x20...0x2F	Diagnostic
0x30...0x3F	Communication
0x40...0xFE	Preferred Index
0x0100...0x3FFF	Extended Index
0x4000...0xFFFF	not specified (reserved)

The use and implementation of these ranges is the responsibility of the sensor/actuator manufacturer.

For clarification, just a few of the possible indices are listed here; refer to the documentation for the IO-Link device used.

Index	Name
0x0010	Vendor Name
0x0011	Vendor Text
0x0012	Product Name
0x0013	Product ID
0x0015	Serial Number
0x0016	Hardware Revision
0x0017	Firmware Revision
...	...

3 Mounting and wiring

3.1 Installation on mounting rails

WARNING	**Risk of electric shock and damage of device!** Bring the bus terminal system into a safe, powered down state before starting installation, disassembly or wiring of the Bus Terminals!

Assembly

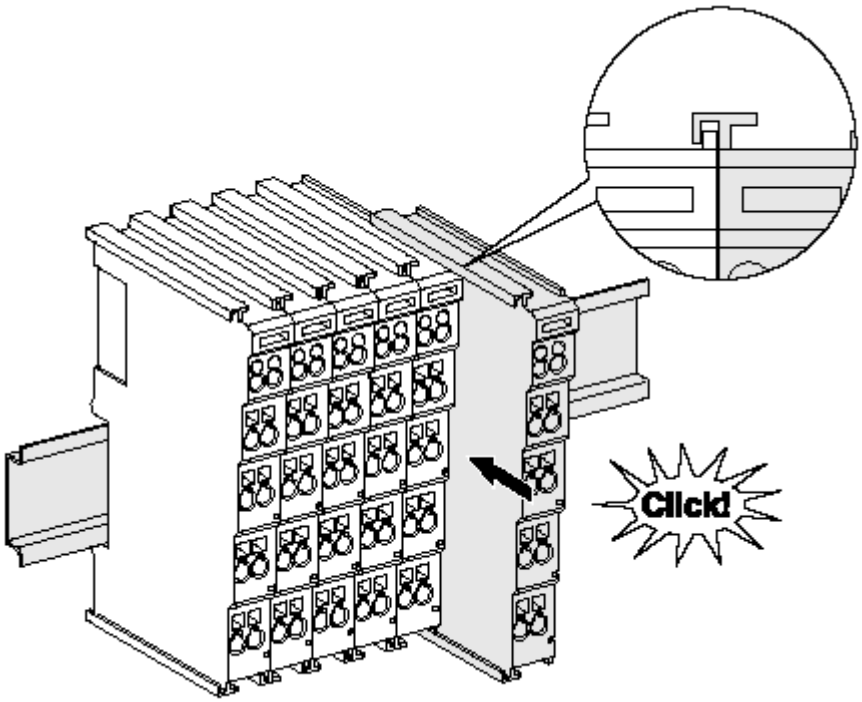

Fig. 3: Attaching on mounting rail

The Bus Coupler and Bus Terminals are attached to commercially available 35 mm mounting rails (DIN rails according to EN 60715) by applying slight pressure:

1. First attach the Fieldbus Coupler to the mounting rail.
2. The Bus Terminals are now attached on the right-hand side of the Fieldbus Coupler. Join the components with tongue and groove and push the terminals against the mounting rail, until the lock clicks onto the mounting rail.
   If the Terminals are clipped onto the mounting rail first and then pushed together without tongue and groove, the connection will not be operational! When correctly assembled, no significant gap should be visible between the housings.

Note	**Fixing of mounting rails** The locking mechanism of the terminals and couplers extends to the profile of the mounting rail. At the installation, the locking mechanism of the components must not come into conflict with the fixing bolts of the mounting rail. To mount the mounting rails with a height of 7.5 mm under the terminals and couplers, you should use flat mounting connections (e.g. countersunk screws or blind rivets).

Disassembly

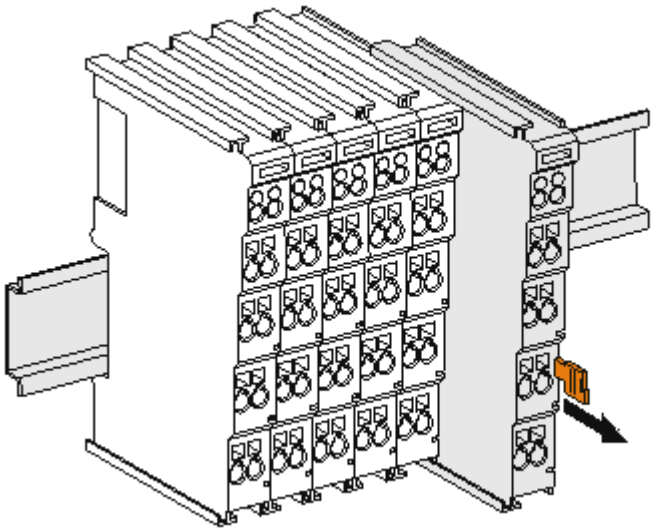

Fig. 4: Disassembling of terminal

Each terminal is secured by a lock on the mounting rail, which must be released for disassembly:

1. Pull the terminal by its orange-colored lugs approximately 1 cm away from the mounting rail. In doing so for this terminal the mounting rail lock is released automatically and you can pull the terminal out of the bus terminal block easily without excessive force.
2. Grasp the released terminal with thumb and index finger simultaneous at the upper and lower grooved housing surfaces and pull the terminal out of the bus terminal block.

Connections within a bus terminal block

The electric connections between the Bus Coupler and the Bus Terminals are automatically realized by joining the components:

- The six spring contacts of the K-Bus/E-Bus deal with the transfer of the data and the supply of the Bus Terminal electronics.
- The power contacts deal with the supply for the field electronics and thus represent a supply rail within the bus terminal block. The power contacts are supplied via terminals on the Bus Coupler (up to 24 V) or for higher voltages via power feed terminals.

Note	**Power Contacts** During the design of a bus terminal block, the pin assignment of the individual Bus Terminals must be taken account of, since some types (e.g. analog Bus Terminals or digital 4-channel Bus Terminals) do not or not fully loop through the power contacts. Power Feed Terminals (KL91xx, KL92xx or EL91xx, EL92xx) interrupt the power contacts and thus represent the start of a new supply rail.

PE power contact

The power contact labeled PE can be used as a protective earth. For safety reasons this contact mates first when plugging together, and can ground short-circuit currents of up to 125 A.

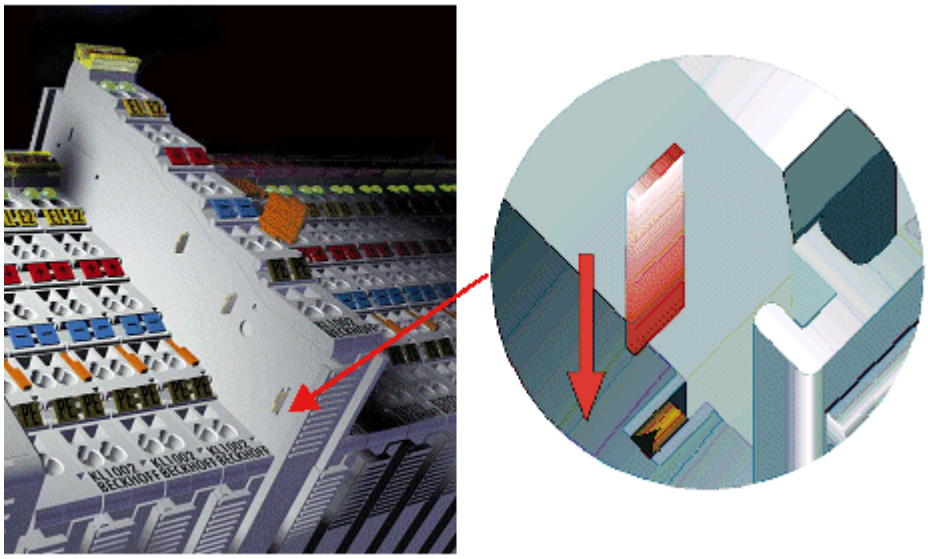

Fig. 5: *Power contact on left side*

Attention	**Possible damage of the device** Note that, for reasons of electromagnetic compatibility, the PE contacts are capacitatively coupled to the mounting rail. This may lead to incorrect results during insulation testing or to damage on the terminal (e.g. disruptive discharge to the PE line during insulation testing of a consumer with a nominal voltage of 230 V). For insulation testing, disconnect the PE supply line at the Bus Coupler or the Power Feed Terminal! In order to decouple further feed points for testing, these Power Feed Terminals can be released and pulled at least 10 mm from the group of terminals.
WARNING	**Risk of electric shock!** The PE power contact must not be used for other potentials!

3.2 Installation instructions for enhanced mechanical load capacity

WARNING	**Risk of injury through electric shock and damage to the device!** Bring the Bus Terminal system into a safe, de-energized state before starting mounting, disassembly or wiring of the Bus Terminals!

Additional checks

The terminals have undergone the following additional tests:

Verification	Explanation
Vibration	10 frequency runs in 3 axes
	6 Hz < f < 60 Hz displacement 0.35 mm, constant amplitude
	60.1 Hz < f < 500 Hz acceleration 5 *g*, constant amplitude
Shocks	1000 shocks in each direction, in 3 axes
	25 *g*, 6 ms

Additional installation instructions

For terminals with enhanced mechanical load capacity, the following additional installation instructions apply:

- The enhanced mechanical load capacity is valid for all permissible installation positions
- Use a mounting rail according to EN 60715 TH35-15
- Fix the terminal segment on both sides of the mounting rail with a mechanical fixture, e.g. an earth terminal or reinforced end clamp
- The maximum total extension of the terminal segment (without coupler) is:
  64 terminals (12 mm mounting with) or 32 terminals (24 mm mounting with)
- Avoid deformation, twisting, crushing and bending of the mounting rail during edging and installation of the rail
- The mounting points of the mounting rail must be set at 5 cm intervals
- Use countersunk head screws to fasten the mounting rail
- The free length between the strain relief and the wire connection should be kept as short as possible. A distance of approx. 10 cm should be maintained to the cable duct.

3.3 Connection system

 **WARNING**	**Risk of electric shock and damage of device!** Bring the bus terminal system into a safe, powered down state before starting installation, disassembly or wiring of the Bus Terminals!

Overview

The Bus Terminal system offers different connection options for optimum adaptation to the respective application:

- The terminals of KLxxxx and ELxxxx series with standard wiring include electronics and connection level in a single enclosure.
- The terminals of KSxxxx and ESxxxx series feature a pluggable connection level and enable steady wiring while replacing.
- The High Density Terminals (HD Terminals) include electronics and connection level in a single enclosure and have advanced packaging density.

Standard wiring

Fig. 6: Standard wiring

The terminals of KLxxxx and ELxxxx series have been tried and tested for years.
They feature integrated screwless spring force technology for fast and simple assembly.

Pluggable wiring

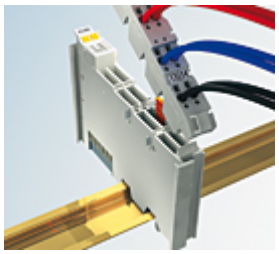

Fig. 7: Pluggable wiring

The terminals of KSxxxx and ESxxxx series feature a pluggable connection level.
The assembly and wiring procedure for the KS series is the same as for the KLxxxx and ELxxxx series.
The KS/ES series terminals enable the complete wiring to be removed as a plug connector from the top of the housing for servicing.
The lower section can be removed from the terminal block by pulling the unlocking tab.
Insert the new component and plug in the connector with the wiring. This reduces the installation time and eliminates the risk of wires being mixed up.

The familiar dimensions of the terminal only had to be changed slightly. The new connector adds about 3 mm. The maximum height of the terminal remains unchanged.

A tab for strain relief of the cable simplifies assembly in many applications and prevents tangling of individual connection wires when the connector is removed.

Conductor cross sections between 0.08 mm^2 and 2.5 mm^2 can continue to be used with the proven spring force technology.

The overview and nomenclature of the product names for KSxxxx and ESxxxx series has been retained as known from KLxxxx and ELxxxx series.

High Density Terminals (HD Terminals)

Fig. 8: *High Density Terminals*

The Bus Terminals from these series with 16 connection points are distinguished by a particularly compact design, as the packaging density is twice as large as that of the standard 12 mm Bus Terminals. Massive conductors and conductors with a wire end sleeve can be inserted directly into the spring loaded terminal point without tools.

Note	**Wiring HD Terminals** The High Density (HD) Terminals of the KLx8xx and ELx8xx series doesn't support steady wiring.

Ultrasonically "bonded" (ultrasonically welded) conductors

Note	**Ultrasonically “bonded" conductors** It is also possible to connect the Standard and High Density Terminals with ultrasonically "bonded" (ultrasonically welded) conductors. In this case, please note the tables concerning the wire-size width [▶ 17] below!

Wiring

Terminals for standard wiring ELxxxx/KLxxxx and for pluggable wiring ESxxxx/KSxxxx

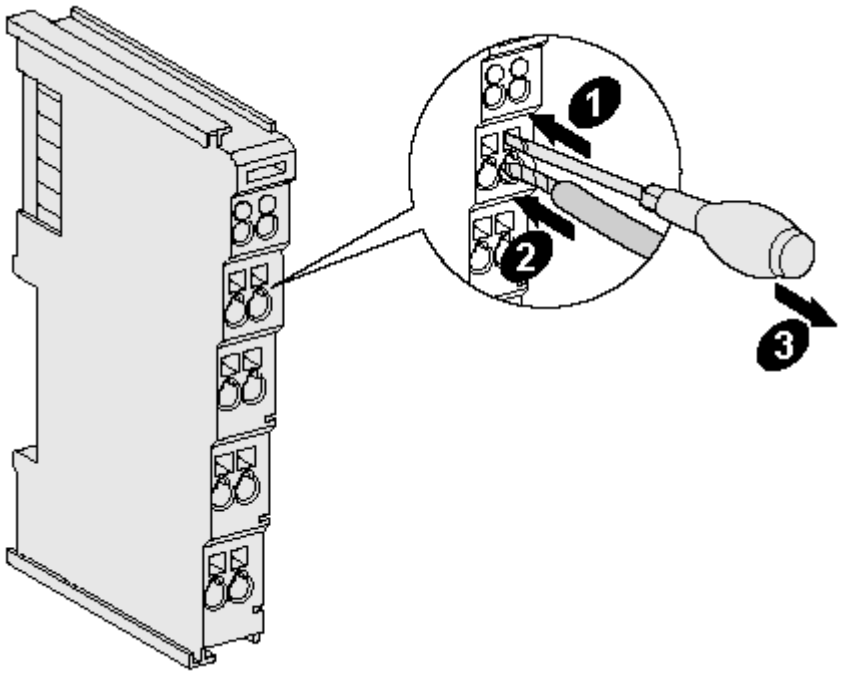

Fig. 9: Mounting a cable on a terminal connection

Up to eight connections enable the connection of solid or finely stranded cables to the Bus Terminals. The terminals are implemented in spring force technology. Connect the cables as follows:

1. Open a spring-loaded terminal by slightly pushing with a screwdriver or a rod into the square opening above the terminal.
2. The wire can now be inserted into the round terminal opening without any force.
3. The terminal closes automatically when the pressure is released, holding the wire securely and permanently.

Terminal housing	ELxxxx, KLxxxx	ESxxxx, KSxxxx
Wire size width	0.08 ... 2,5 mm²	0.08 ... 2.5 mm²
Wire stripping length	8 ... 9 mm	9 ... 10 mm

High Density Terminals ELx8xx, KLx8xx (HD)

The conductors of the HD Terminals are connected without tools for single-wire conductors using the direct plug-in technique, i.e. after stripping the wire is simply plugged into the contact point. The cables are released, as usual, using the contact release with the aid of a screwdriver. See the following table for the suitable wire size width.

Terminal housing	High Density Housing
Wire size width (conductors with a wire end sleeve)	0.14... 0.75 mm²
Wire size width (single core wires)	0.08 ... 1.5 mm²
Wire size width (fine-wire conductors)	0.25 ... 1.5 mm²
Wire size width (ultrasonically “bonded" conductors)	only 1.5 mm² (see notice [▶ 16]!)
Wire stripping length	8 ... 9 mm

Shielding

 Note	**Shielding** Analog sensors and actors should always be connected with shielded, twisted paired wires.

3.4 Connection

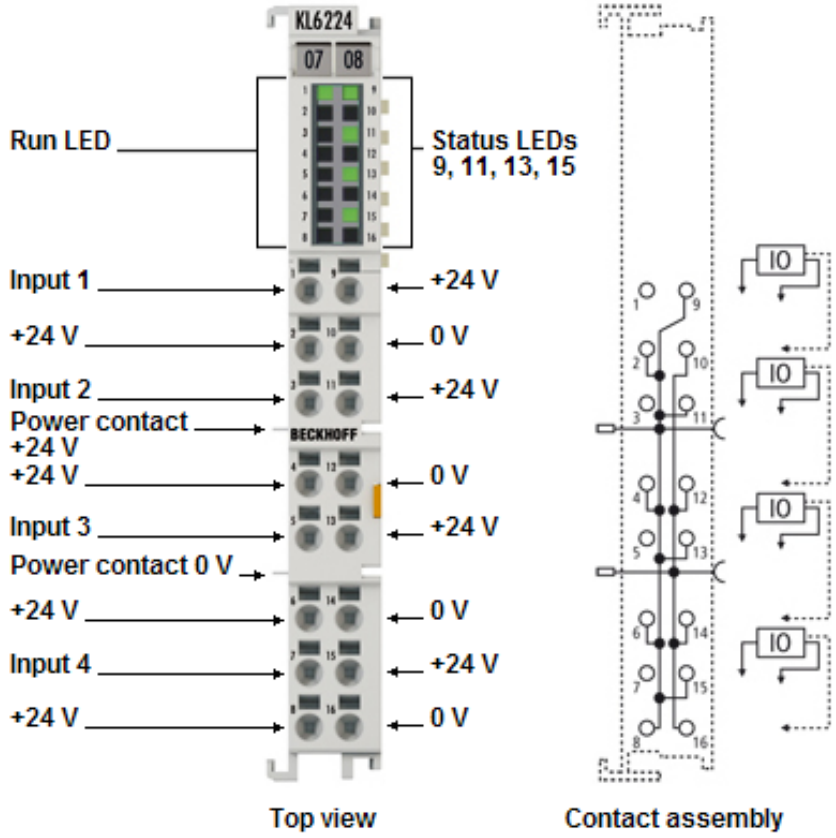

Fig. 10: KL6224 connection

Connection

Terminal point		Description
Name	No.	
Input 1	1	Input 1
+ 24 V	2	+ 24 V
Input 2	3	Input 2
+ 24 V	4	+ 24 V
Input 3	5	Input 3
+ 24 V	6	+ 24 V
Input 4	7	Input 4
+ 24 V	8	+ 24 V
+ 24 V	9	+ 24 V
0 V	10	0 V
+ 24 V	11	+ 24 V
0 V	12	0 V
+ 24 V	13	+ 24 V
0 V	14	0 V
+ 24 V	15	+ 24 V
0 V	16	0 V

3.5 LED display

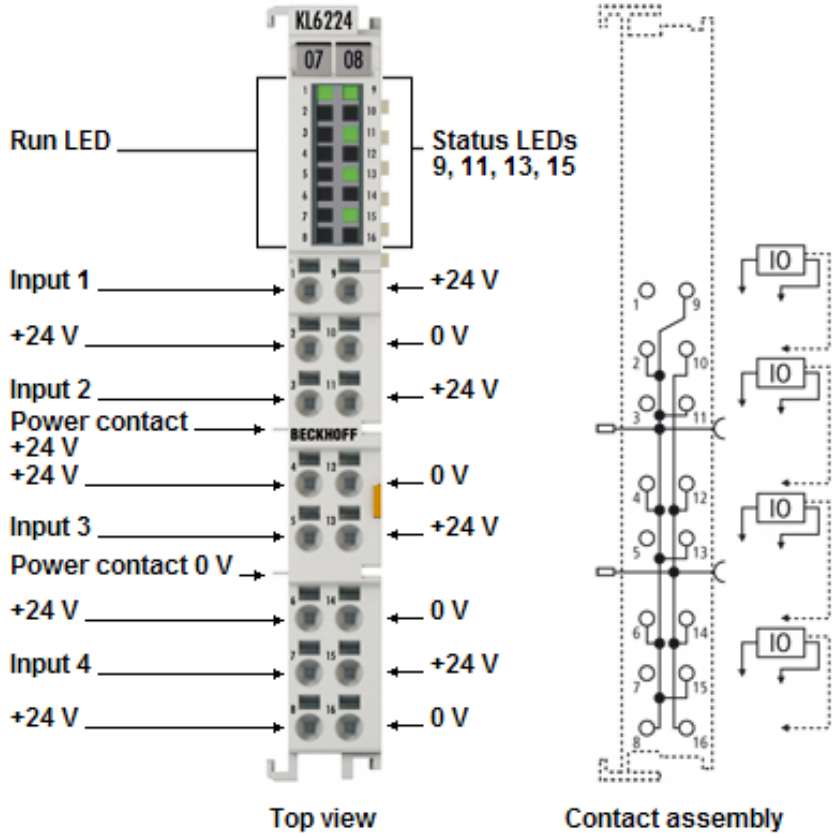

Fig. 11: LEDs

LEDs

LED	Color	Meaning	
RUN	green	These LEDs indicate the terminal's operating state:	
		off	no data transfer on the K-bus
		flashes	data transmission on the K-bus
State Ch. 1 - 4	green	on / off	Status of the signal line (if configured as STD in / out)
		flashes briefly twice	establishing IO-Link communication
		permanently flashing	IO-Link communication established and in operation

4 KS2000 Configuration Software

4.1 KS2000 - Introduction

The KS2000 configuration software permits configuration, commissioning and parameterization of bus couplers, of the affiliated bus terminals and of Fieldbus Box Modules. The connection between bus coupler / Fieldbus Box Module and the PC is established by means of the serial configuration cable or the fieldbus.

Fig. 12: KS2000 configuration software

Configuration

You can configure the Fieldbus stations with the Configuration Software KS2000 offline. That means, setting up a terminal station with all settings on the couplers and terminals resp. the Fieldbus Box Modules can be prepared before the commissioning phase. Later on, this configuration can be transferred to the terminal station in the commissioning phase by means of a download. For documentation purposes, you are provided with the breakdown of the terminal station, a parts list of modules used and a list of the parameters you have modified. After an upload, existing fieldbus stations are at your disposal for further editing.

Parameterization

KS2000 offers simple access to the parameters of a fieldbus station: specific high-level dialogs are available for all bus couplers, all intelligent bus terminals and Fieldbus Box modules with the aid of which settings can be modified easily. Alternatively, you have full access to all internal registers of the bus couplers and intelligent terminals. Refer to the register description for the meanings of the registers.

Commissioning

The KS2000 software facilitates commissioning of machine components or their fieldbus stations: Configured settings can be transferred to the fieldbus modules by means of a download. After a *login* to the terminal station, it is possible to define settings in couplers, terminals and Fieldbus Box modules directly *online*. The same high-level dialogs and register access are available for this purpose as in the configuration phase.

The KS2000 offers access to the process images of the bus couplers and Fieldbus Box modules.

- Thus, the coupler's input and output images can be observed by monitoring.
- Process values can be specified in the output image for commissioning of the output modules.

All possibilities in the *online mode* can be used in parallel with the actual fieldbus mode of the terminal station. The fieldbus protocol always has the higher priority in this case.

4.2 Configuration of the KL6224

Connect the configuration interface of your fieldbus coupler with the serial interface of your PC via the configuration cable and start the *KS2000* Configuration Software.

Click on the *Login* button. The configuration software will now load the information for the connected fieldbus station.
In the example shown, this is

- a BK9000 Ethernet Coupler
- a KL1xx2 Digital Input Terminal
- a KL6224 IO-Link terminal
- a KL9010 bus end terminal

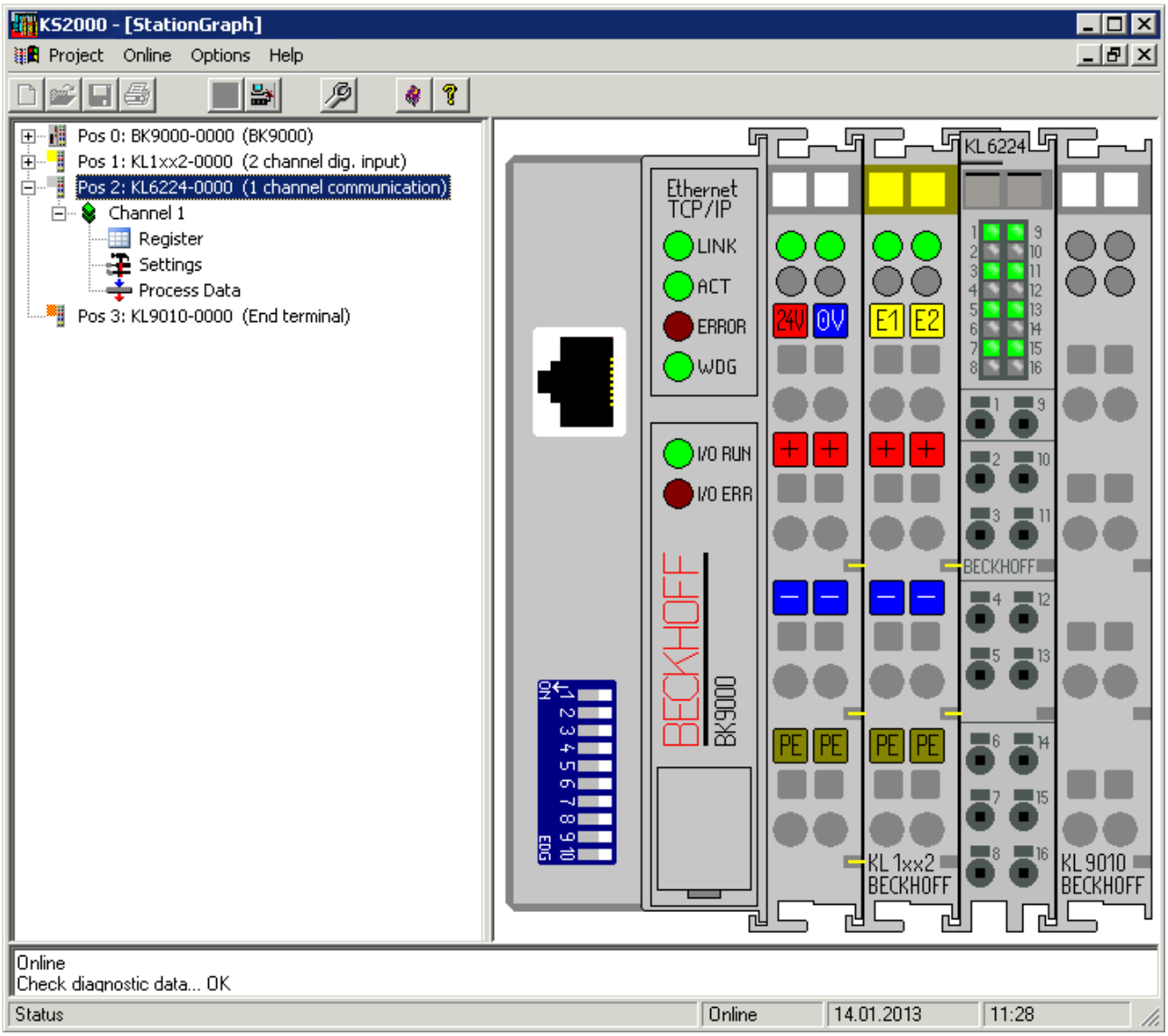

Fig. 13: Display of the fieldbus station in KS2000

The left-hand KS2000 window displays the terminals of the fieldbus station in a tree structure.
The right-hand KS2000 window contains a graphic display of the fieldbus station terminals.

In the tree structure of the left-hand window, click on the plus-sign next to the terminal whose parameters you wish to change (item 1 in the example).

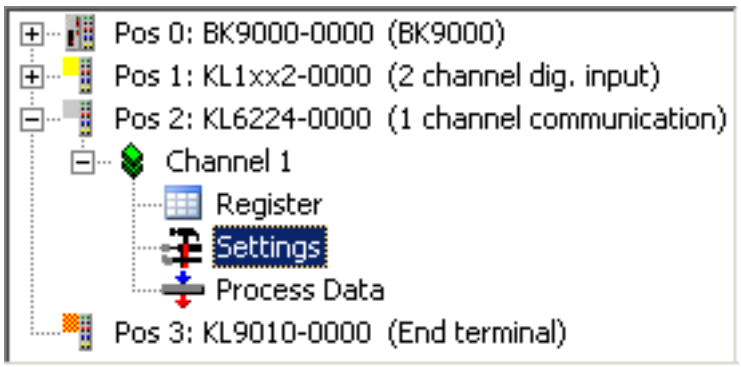

Fig. 14: KS2000 branches for channel 1 of the KL6224

For the KL6224, the branches *Register*, *Settings* and *ProcData* are displayed:

- Register [▶ 25] permits direct access to the registers of the KL6224.
- Dialog boxes for the parameterization of the KL6224 can be found under Settings [▶ 26].
- ProcData displays the process data of the KL6224.

4.3 Register

You can access the registers of the KL6224 directly under *Register*. The meaning of the register is explained in the register overview [▶ 39].

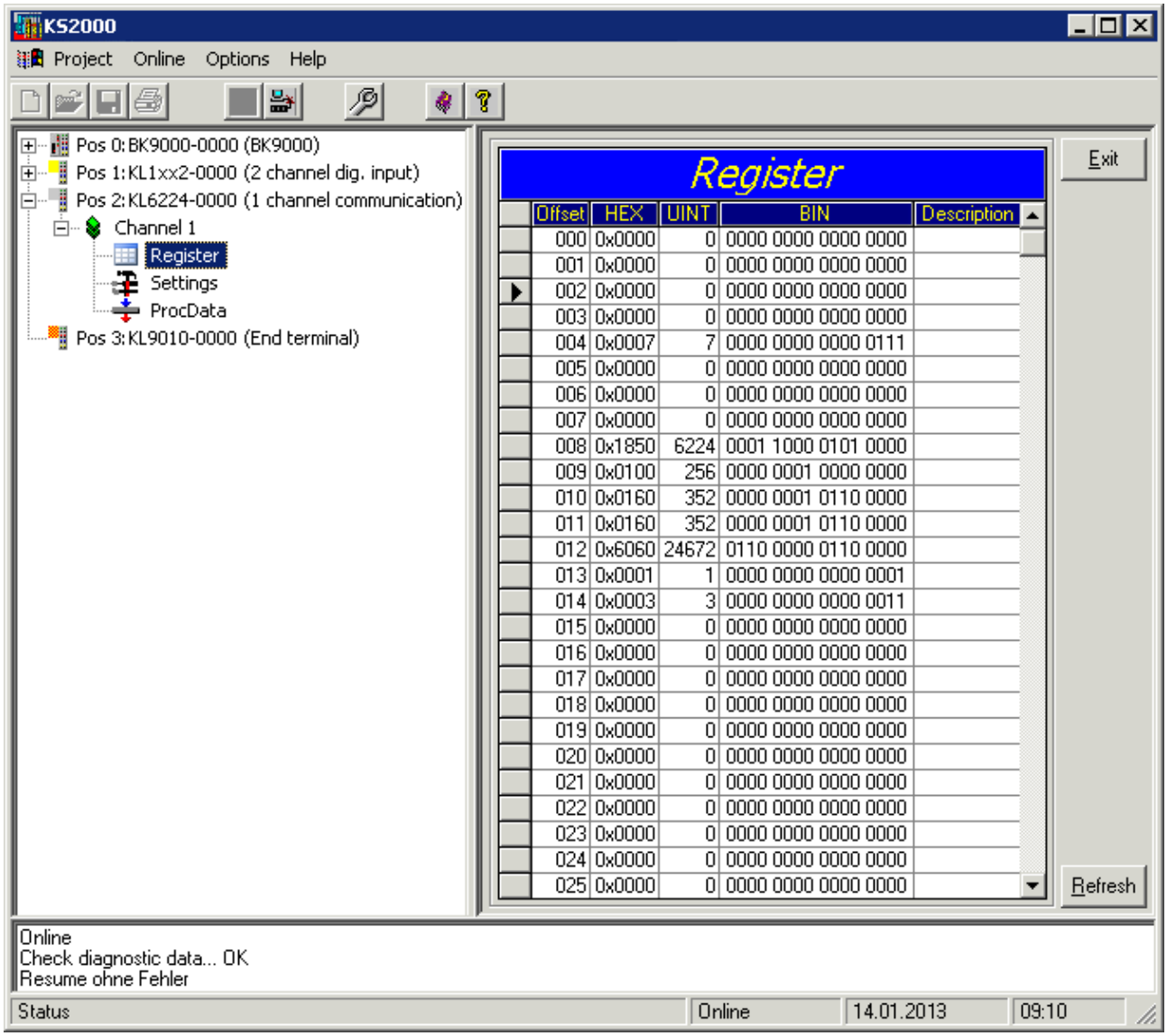

Fig. 15: Register view in KS2000

The screenshot shows the registers of the KL6224.

4.4 Settings

Start the KS2000 Configuration Software.

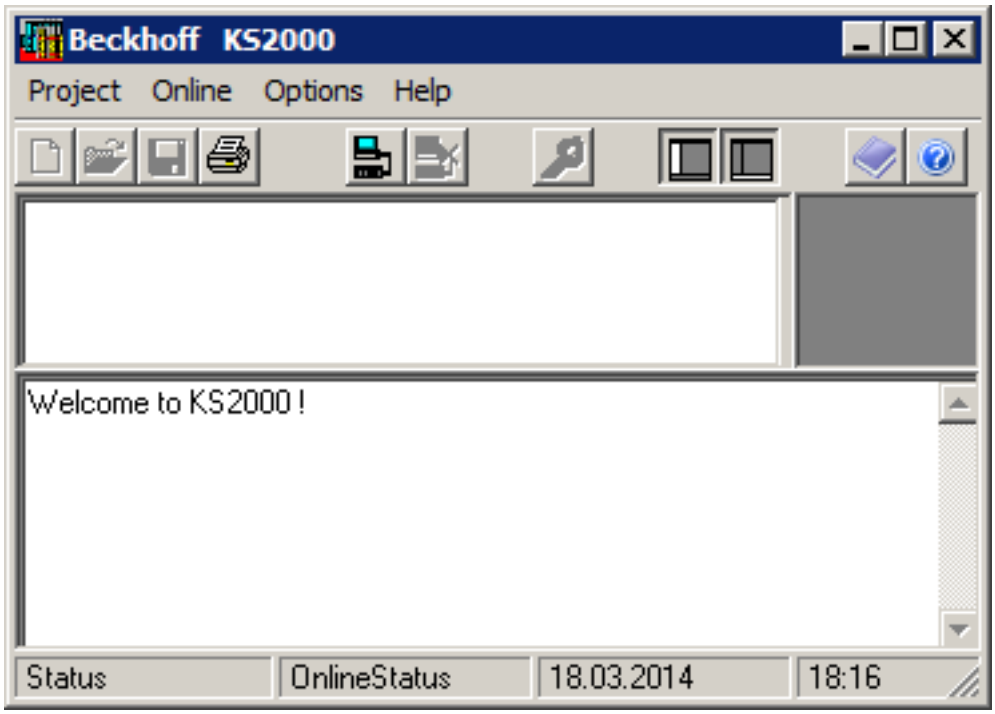

Fig. 16: KS2000 - Starting the configuration software

Communication channel

The communication between KS2000 and the KL6224 can take place either via the fieldbus (**via ADS**) or via a serial cable (**via COM** interface).

Click on *Options/Communication channel* to select the communication channel.

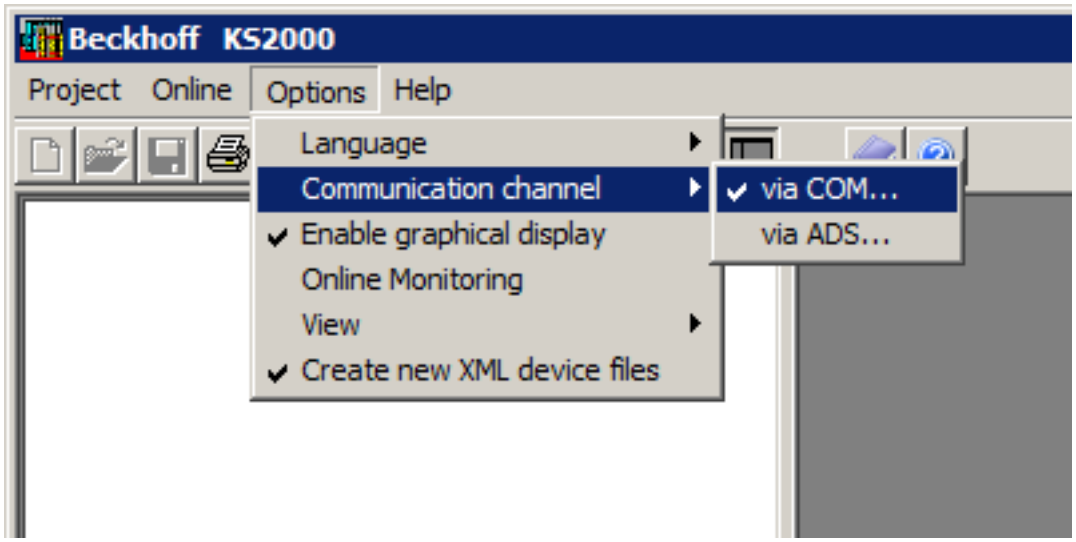

Fig. 17: KS2000 - Selecting the communication channel

Login

Click on *Login*. 

The fieldbus station is displayed as a tree structure.

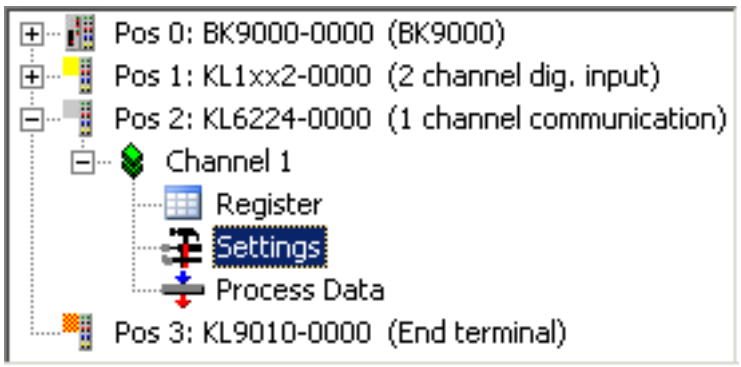

Fig. 18: KS2000 - Display of the fieldbus station

Insert the IO-Link devices

There are three options to integrate an IO-Link device:

1. Automatic scan [▶ 27] of the IO-Link ports, [Scan devices] button
2. Manual insertion [▶ 28] via catalog, [Catalog] button
3. Import the device description [▶ 30], [Import Device Description] button

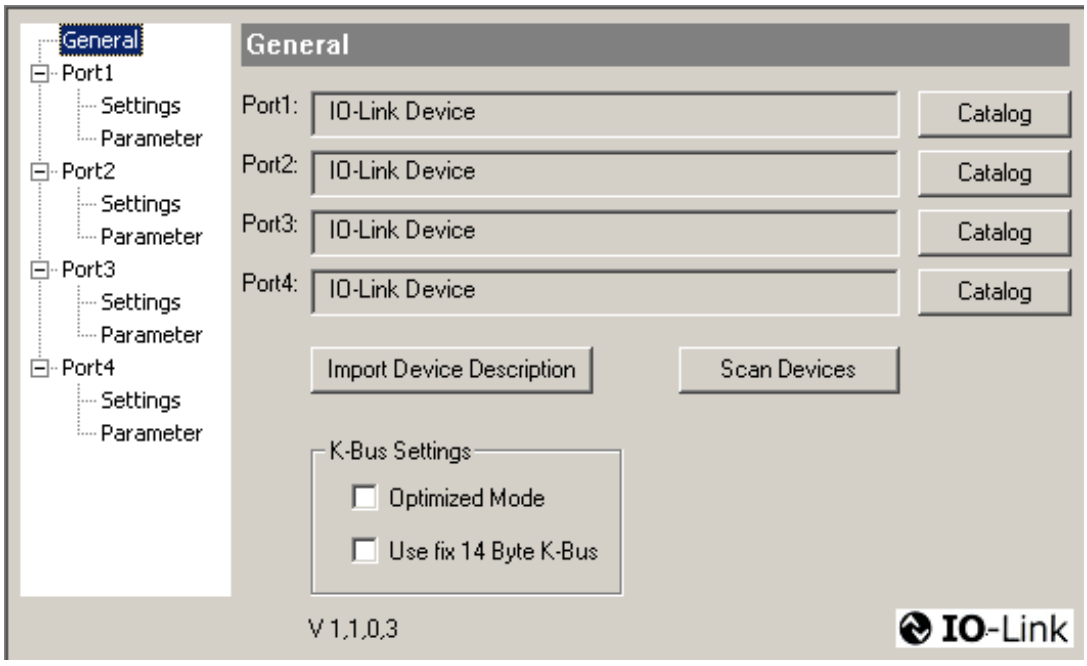

Fig. 19: Insert the IO-Link devices

Automatic scan of the IO-Link ports

Connect the IO-Link sensor to the KL6224.

Switch on the Bus Coupler with the KL6224.

Click on *Login*  to connect to the fieldbus station and read the modified KL6224 process image.

In the *General* dialog click on the [Scan devices] button.

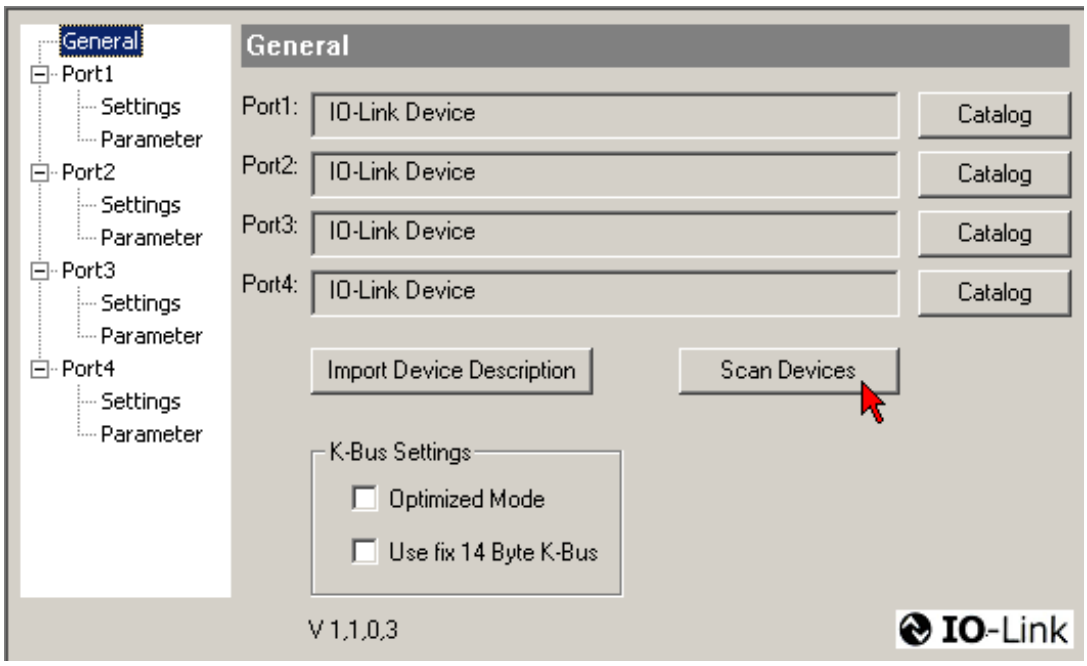

Fig. 20: Automatic scan of the IO-Link ports

Note	**Communication Mode** For scanning, the Communication mode must not be set to Communication (Port1/Settings).

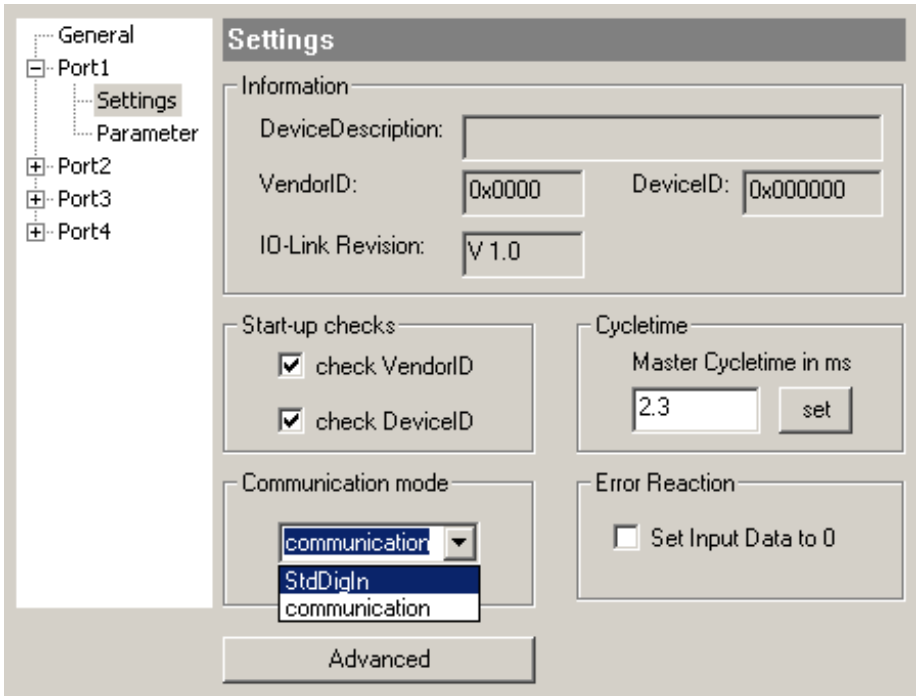

Fig. 21: Display of the detected IO-Link devices

The detected IO-Link devices are displayed, and the required process data are created.

Manual insertion via catalog

In the *General* dialog click on the [Catalog] button.

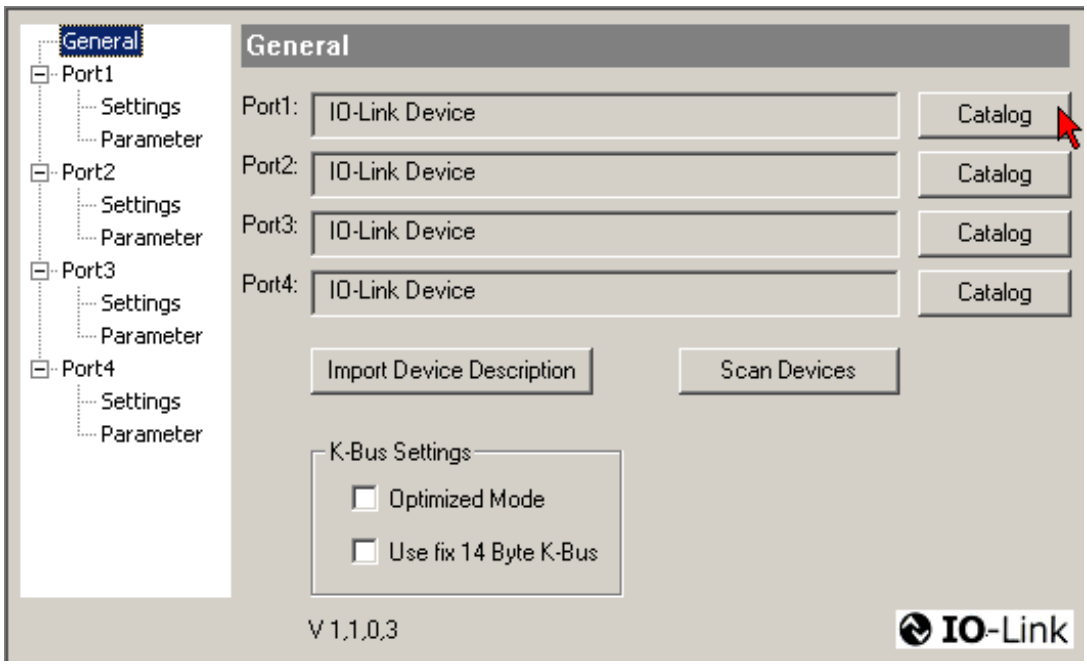

Fig. 22: Manual insertion via catalog

Here you can

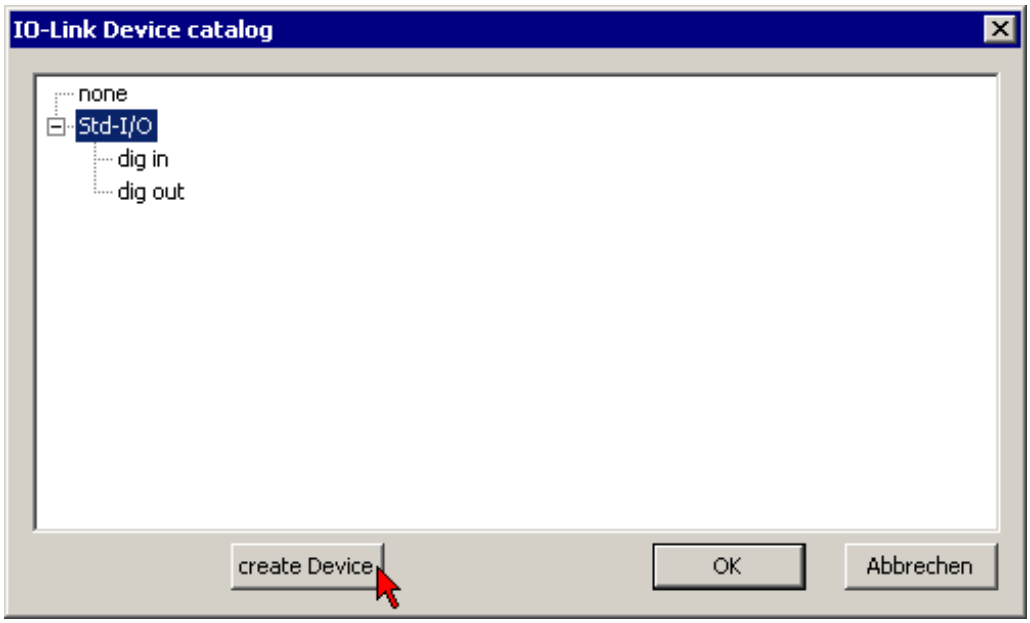

Fig. 23: Calling the *create Device* dialog

use the *create Device* dialog to manually create an IO-Link device

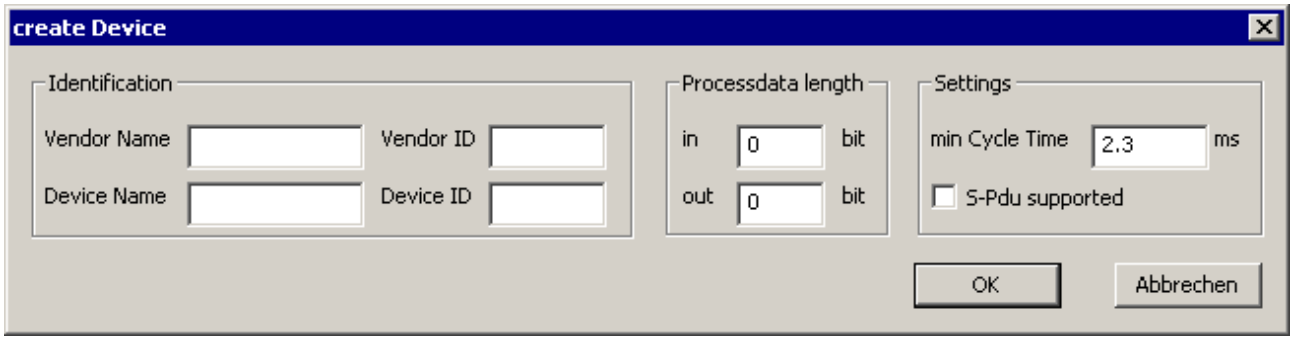

Fig. 24: The *create Device* dialog

with the main communication parameters.

Import the device description

In the *General* dialog click on the [Import Device Description] button.

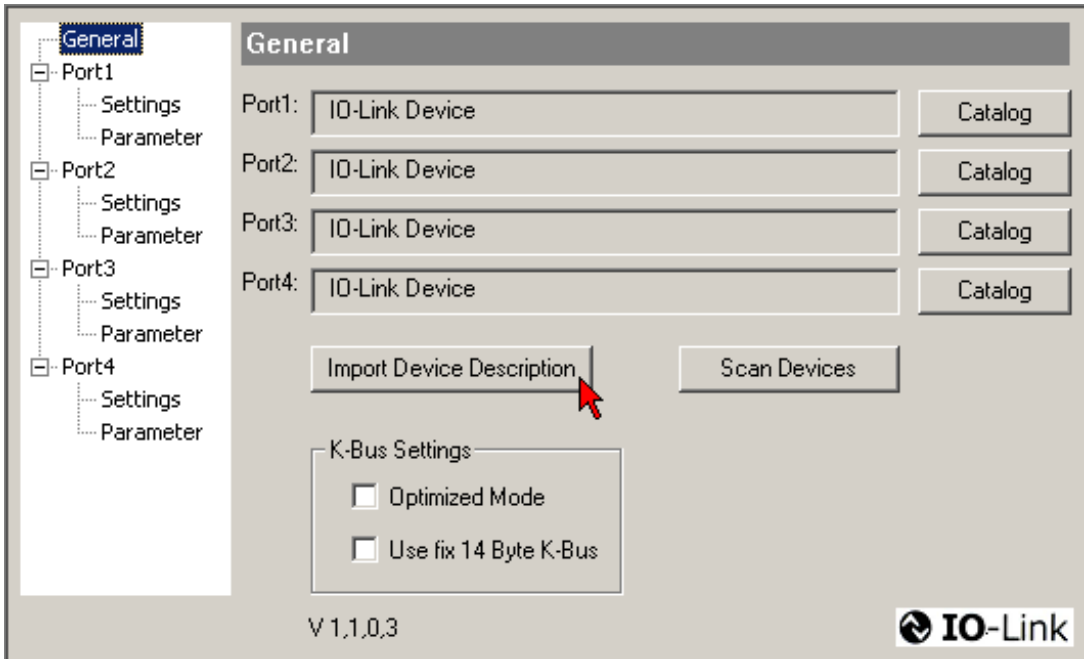

Fig. 25: Import the device description

The XML files for the IO-Link devices are stored in the IO-Link folder of the KS2000 (e.g. under Windows 7 in the folder C:\Program Files (x86)\KS2000_V4\IOLink)

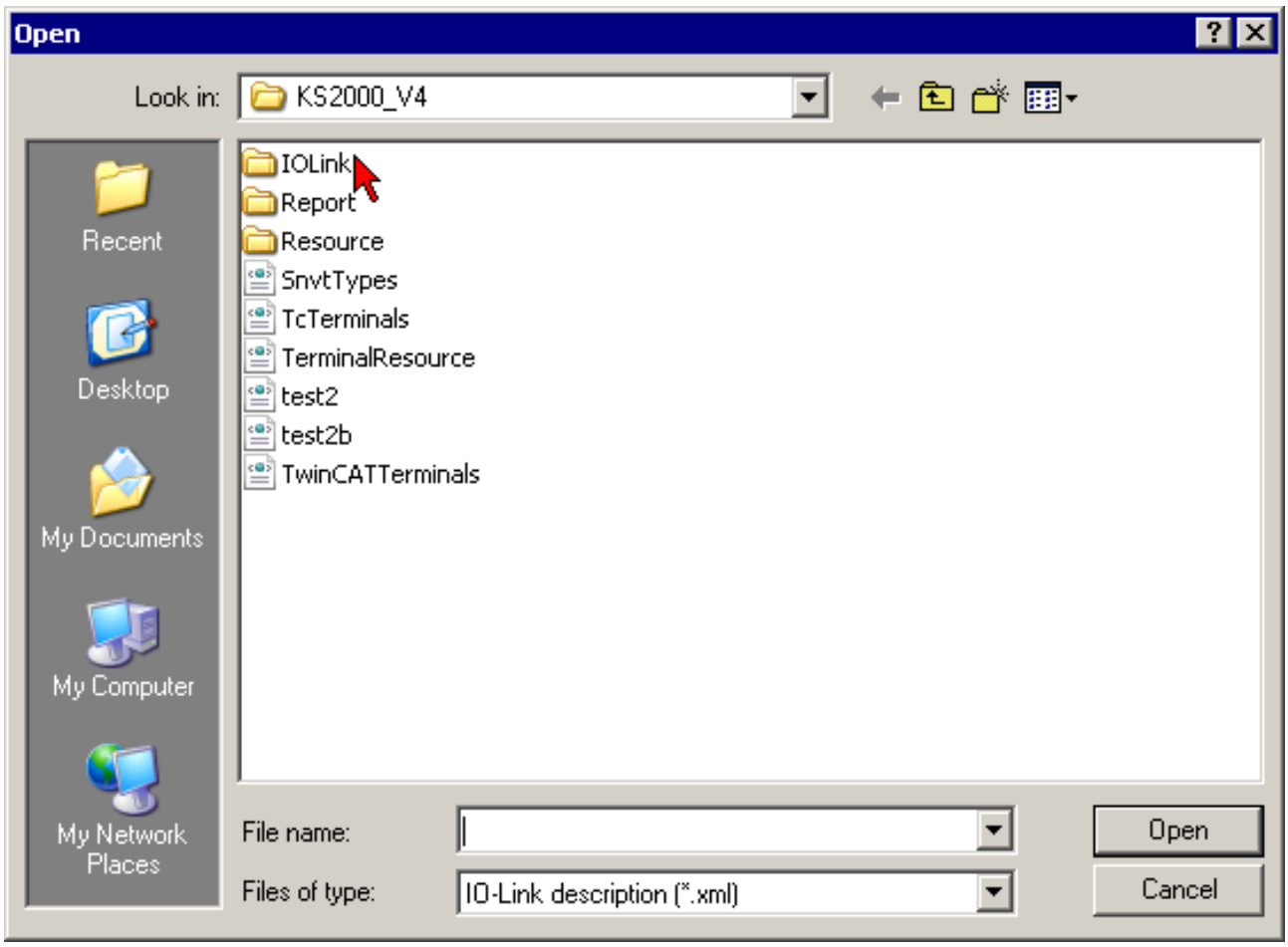

Fig. 26: Selecting the XML file

Select the XML file for the required sensor and open it.

Now continue scanning the IO-Link ports (see above [▶ 27]).

The detected IO-Link devices are displayed, and the required process data are created.

Note	**Always import IODD** Always read in existing IODDs before scanning or manually inserting IO-Link devices, in order to obtain further sensor-specific information.

K-bus interface

Please note: When changing IO-Link devices the following message may occur.

Fig. 27: The K-bus interface of the KL6224 has changed

This message indicates that the K-bus interface of the KL6224 has changed, because the connection of a further IO-Link device has changed the process image of the KL6224.
However, this message does not affect the integration of IO-Link devices.

IO-Link port settings

In the *General* dialog click on *Settings* for the required port.

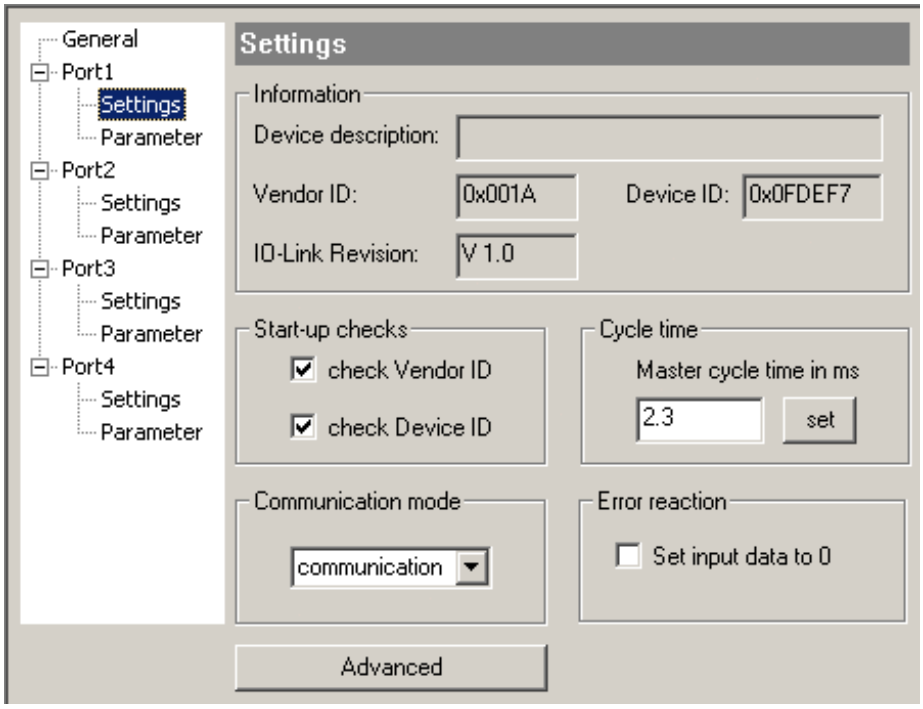

Fig. 28: IO-Link port settings?

StartUpChecks

This parameter can be used to specify that the vendor ID and the device ID should be checked when the IO-Link device starts up.

CycleTime

Specifies the cycle time for the IO-Link master

Communication mode

An IO-Link device can be operated in different modes. The default mode for IO-Link devices is Communication.

Error reaction

If this checkbox is checked, the input data are set to 0 in the event of an error.

Advanced

Click on [Advanced] to open the dialog for the advanced settings.

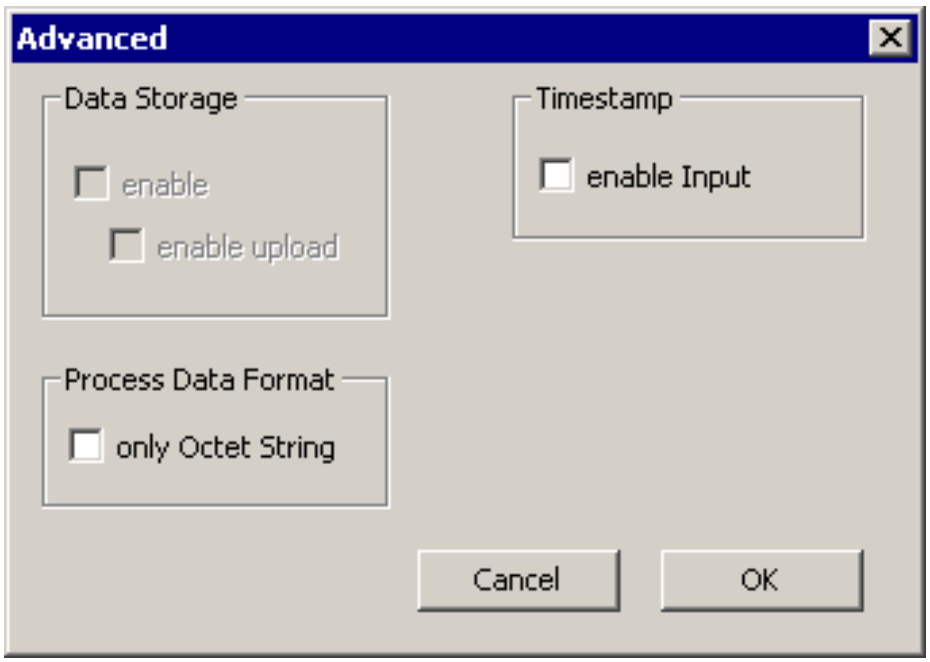

Fig. 29: Advanced settings

Data Storage

Please note the sensor versions:

- V1.0: Data Storage is not supported
- V1.1: Data Storage is supported: in delivery state (default) data (sensor parameters) are stored.

Process Data Format

Here you can adjust the process data format.

If the checkbox *only Octet String* is checked, complex data types (process data) are created as octet string in the interest of simplification.

5 Access from the user program

5.1 Process image

The process image of the KL6224 consists of a 6-byte parameter data block and a 6-, 8-, 18-, 30- or 42-byte process data block.
The result is a 12-, 14-, 24-, 36- or 48-byte process image.

The size of the process image can be set via the KS2000 configuration software, or fieldbus-specific via the Bus Coupler. It is stored in register R27 of the KL6224.
After changing the process image the Bus Coupler must be restarted for the modified process image to take effect.

Please refer to chapter Firmeware version of the Bus Couplers [▶ 52] to find out which process images your Bus Coupler supports.

Description of the process images

Note

Selection of the process image

The process image should not be larger than necessary for operating your IO-Link devices! This saves bandwidth for the higher-level fieldbus and the K-bus. Since only 12 bytes can be transferred to the KL6224 in a K-bus cycle, the update rate of the other terminals is also optimally fast with minimal selection.

12-byte process image

The 12-byte process image consists of a 6-byte parameter data block and a 6-byte process data block.

Output data (PLC -> KL6224)

Parameter data block (6 bytes)						Process data block (6 bytes)					
Byte 0	**Byte 1**	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**	**Byte 8**	**Byte 9**	**Byte 10**	**Byte 11**
CB0	CB1	ParaOut0	ParaOut1	ParaOut2	ParaOut3	DataOut0	DataOut1	DataOut2	DataOut3	DataOut4	DataOut5

Input data (KL6224 -> PLC)

Parameter data block (6 bytes)						Process data block (6 bytes)					
Byte 0	**Byte 1**	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**	**Byte 8**	**Byte 9**	**Byte 10**	**Byte 11**
SB0	SB1	ParaIn0	ParaIn1	ParaIn2	ParaIn3	StatusCh1	StatusCh2	StatusCh3	StatusCh4	DataIn0	DataIn1

Key

CB n: Control byte n of the KL6224
SB n: Status byte n of the KL6224
Para Out n: Output parameter, byte n
Para In n: Input parameter, byte n
StatusCH n: Status information of IO-Link device n
DataOut n: Output data byte n
DataIn n: Input data byte n

12-byte process image, compressed mode

In compressed mode the data are compacted in order to save space (e.g. for CANopen Bus Coupler). It can be activated via the parameter Master Control (register R41 on register page 4) [▶ 46].
If compressed mode is active, it applies to all IO-Link ports.

The compressed mode is also available for the larger process images (14, 24, 36 and 48 bytes).
For the larger process images the structure of the mixed data block is exactly as shown here, and the process data block then includes the further data bytes from DataOut2/DataIn2.

Output data (PLC -> KL6224)

Mixed data block (6 bytes)						Process data block (6 bytes)					
Byte 0	**Byte 1**	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**	**Byte 8**	**Byte 9**	**Byte 10**	**Byte 11**
CB0	CB1	ParaOut0	ParaOut1	ParaOut2[1] or DataOut[2]	ParaOut3[1] or DataOut[2]	DataOut2	DataOut3	DataOut4	DataOut5	DataOut6	DataOut7

Input data (KL6224 -> PLC)

Mixed data block (6 bytes)						Process data block (6 bytes)					
Byte 0	**Byte 1**	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**	**Byte 8**	**Byte 9**	**Byte 10**	**Byte 11**
SB0[1] or StatusCh1/2[2]	SB1[1] or StatusCh3/4[2]	ParaIn0	ParaIn1	ParaIn3[1] or DataIn0[2]	ParaIn4[1] or DataIn1[2]	DataIn2	DataIn3	DataIn4	DataIn5	DataIn6	DataIn7

[1]) for register communication (see chapter Control and status byte [▶ 38])
[2]) for process data mode

Key

CB n: Control byte n of the KL6224
SB n: Status byte n of the KL6224
Para Out n: Output parameter, byte n
Para In n: Input parameter, byte n
StatusCh1/2: Status information of IO-Link device n (4-bit low = port 1 / 4-bit high = port 2)
StatusCh3/4: Status information of IO-Link device n (4-bit low = port 3 / 4-bit high = port 4)
DataOut n: Output data byte n
DataIn n: Input data byte n

14-byte process image

The 14-byte process image consists of a 6-byte parameter data block and an 8-byte process data block.

Output data (PLC -> KL6224)

Parameter data block (6 bytes)						Process data block (8 bytes)							
Byte 0	**Byte 1**	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**	**Byte 8**	**Byte 9**	**Byte 10**	**Byte 11**	**Byte 12**	**Byte 13**
CB0	CB1	ParaOut0	ParaOut1	ParaOut2	ParaOut3	DataOut0	DataOut1	DataOut2	DataOut3	DataOut4	DataOut5	DataOut6	DataOut7

Input data (KL6224 -> PLC)

Parameter data block (6 bytes)						Process data block (8 bytes)							
Byte 0	**Byte 1**	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**	**Byte 8**	**Byte 9**	**Byte 10**	**Byte 11**	**Byte 12**	**Byte 13**
SB0	SB1	ParaIn0	ParaIn1	ParaIn2	ParaIn3	StatusCh1	StatusCh2	StatusCh3	StatusCh4	DataIn0	DataIn1	DataIn2	DataIn3

Key

See 12-byte process image.

24-byte process image (default)

The 24-byte process image consists of a 6-byte parameter data block and an 18-byte process data block. This setting is enabled in the delivery state of the KL6224.

Output data (PLC -> KL6224)

Parameter data block (6 bytes)						Process data block (18 bytes)					
Byte 0	**Byte 1**	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**	**Byte 8**	**Byte 9**	**Byte 10**	**Byte 11**
CB0	CB1	ParaOut0	ParaOut1	ParaOut2	ParaOut3	DataOut0	DataOut1	DataOut2	DataOut3	DataOut4	DataOut5

Process data block (18 bytes, continuation)											
Byte 12	Byte 13	Byte 14	Byte 15	Byte 16	Byte 17	Byte 18	Byte 19	Byte 20	Byte 21	Byte 22	Byte 23
DataOut 6	DataOut 7	DataOut 8	DataOut 9	DataOut 10	DataOut 11	DataOut 12	DataOut 13	DataOut14	DataOut15	DataOut16	DataOut17

Input data (KL6224 -> PLC)

Parameter data block (6 bytes)						Process data block (18 bytes)					
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9	Byte 10	Byte 11
SB0	SB1	ParaIn0	ParaIn1	ParaIn2	ParaIn3	StatusCh1	StatusCh2	StatusCh3	StatusCh4	DataIn0	DataIn1

Process data block (18 bytes, continuation)											
Byte 12	Byte 13	Byte 14	Byte 15	Byte 16	Byte 17	Byte 18	Byte 19	Byte 20	Byte 21	Byte 22	Byte 23
DataIn2	DataIn3	DataIn4	DataIn5	DataIn6	DataIn7	DataIn8	DataIn9	DataIn10	DataIn11	DataIn12	DataIn13

Key

See 12-byte process image.

36-byte process image

The 36-byte process image consists of a 6-byte parameter data block and a 30-byte process data block.

Output data (PLC -> KL6224)

Parameter data block (6 bytes)						Process data block (30 bytes)					
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9	Byte 10	Byte 11
CB0	CB1	ParaOut0	ParaOut 1	ParaOut 2	ParaOut3	DataOut0	DataOut1	DataOut2	DataOut3	DataOut4	DataOut25

Process data block (30 bytes, continuation)											
Byte 12	Byte 13	Byte 14	Byte 15	Byte 16	Byte 17	Byte 18	Byte 19	Byte 20	Byte 21	...	Byte 35
DataOut 6	DataOut 7	DataOut 8	DataOut 9	DataOut 10	DataOut11	DataOut12	DataOut13	DataOut14	DataOut15	...	DataOut29

Input data (KL6224 -> PLC)

Parameter data block (6 bytes)						Process data block (30 bytes)					
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9	Byte 10	Byte 11
SB0	SB1	ParaIn0	ParaIn1	ParaIn2	ParaIn3	StatusCh1	StatusCh2	StatusCh3	StatusCh4	DataIn0	DataIn1

Process data block (30 bytes, continuation)											
Byte 12	Byte 13	Byte 14	Byte 15	Byte 16	Byte 17	Byte 18	Byte 19	Byte 20	Byte 21	...	Byte 35
DataIn2	DataIn3	DataIn4	DataIn5	DataIn6	DataIn7	DataIn8	DataIn9	DataIn10	DataIn11	...	DataIn13

Key

See 12-byte process image.

48-byte process image

The 48-byte process image consists of a 6-byte parameter data block and a 42-byte process data block.

Output data (PLC -> KL6224)

Parameter data block (6 bytes)						Process data block (42 bytes)					
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9	Byte 10	Byte 11
CB0	CB1	ParaOut0	ParaOut 1	ParaOut2	ParaOut3	DataOut0	DataOut1	DataOut2	DataOut3	DataOut4	DataOut25

Process data block (42 bytes, continuation)											
Byte 12	**Byte 13**	**Byte 14**	**Byte 15**	**Byte 16**	**Byte 17**	**Byte 18**	**Byte 19**	**Byte 20**	**Byte 21**	**...**	**Byte 48**
DataOut 6	DataOut 7	DataOut 8	DataOut 9	DataOut 10	DataOut 11	DataOut12	DataOut13	DataOut14	DataOut15	...	DataOut41

Input data (KL6224 -> PLC)

Parameter data block (6 bytes)						Process data block (42 bytes)					
Byte 0	**Byte 1**	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**	**Byte 8**	**Byte 9**	**Byte 10**	**Byte 11**
SB0	SB1	ParaIn0	ParaIn1	ParaIn2	ParaIn3	StatusCh1	StatusCh2	StatusCh3	StatusCh4	DataIn0	DataIn1

Process data block (42 bytes, continuation)											
Byte 12	**Byte 13**	**Byte 14**	**Byte 15**	**Byte 16**	**Byte 17**	**Byte 18**	**Byte 19**	**Byte 20**	**Byte 21**	**...**	**Byte 48**
DataIn2	DataIn3	DataIn4	DataIn5	DataIn6	DataIn7	DataIn8	DataIn9	DataIn10	DataIn11	...	DataIn37

Key

See 12-byte process image.

5.2 Control and status byte

Register communication

Control byte (for register communication)

The control byte (CB) is located in the output image [▶ 34], and is transmitted from the controller to the terminal.

Bit	CB.7	CB.6	CB.5	CB.4	CB.3	CB.2	CB.1	CB.0
Name	RegAccess	R/W	Reg. no.					

Key

Bit	Name	Description	
CB.7	RegAccess	1_{bin}	Register communication switched on
CB.6	R/W	0_{bin}	Read access
		1_{bin}	Write access
CB.5 to CB.0	Reg. no.	Register number: Enter the number of the register [▶ 39] that you want to read or write here.	

Status byte (for register communication)

The status byte (SB) is located in the input image [▶ 34], and is transmitted from terminal to the controller.

Bit	SB.7	SB.6	SB.5	SB.4	SB.3	SB.2	SB.1	SB.0
Name	RegAccess	R/W	Reg. no.					

Key

Bit	Name	Description	
SB.7	RegAccess	1_{bin}	Acknowledgement for register access
SB.6	R	0_{bin}	Read access
SB.5 to SB.0	Reg. no.	Number of the register that was read or written.	

Process data mode

Note

Control and status byte in process data mode

In process data mode the control and status bytes of the KL6224 currently have no additional function.

Control byte in process data mode

The control byte (CB) is located in the output image [▶ 34], and is transmitted from the controller to the terminal.

Bit	CB.7	CB.6	CB.5	CB.4	CB.3	CB.2	CB.1	CB.0
Name	RegAccess	-	-	-	-	-	-	-

Key

Bit	Name	Description	
CB.7	RegAccess	0_{bin}	Register communication off (process data mode)
CB.6 to CB.0	-	0_{bin}	reserved

Status byte in process data mode

The status byte (SB) is located in the input image [▶ 34], and is transmitted from terminal to the controller.

Bit	SB.7	SB.6	SB.5	SB.4	SB.3	SB.2	SB.1	SB.0
Name	RegAccess	-	-	-	-	-	-	-

Key

Bit	Name	Description	
SB.7	RegAccess	0_{bin}	Acknowledgement for process data mode
SB.6 to SB.0	Error	0_{bin}	reserved

5.3 Register overview

The registers are used for parameterization of the terminal and can be read or written via the register communication [▶ 41].

Register	Comment	Default value		R/W	Memory
R0 to R3	reserved	0x0000	0_{dec}	-	-
R4 [▶ 40]	Register page [▶ 45]	0x0004	4_{dec}	R/W	
R5 to R7	reserved	0x0000	0_{dec}	-	-
R8 [▶ 40]	Terminal description	0x1850	6224_{dec}	R	ROM
R9 [▶ 40]	Firmware version	e.g. 0x0100	e.g. 256_{dec}	R	ROM
R10 [▶ 40]	Multiplex shift register	0x0160	352_{dec}	R	ROM
R11 [▶ 40]	Signal channels	0x0160	352_{dec}	R	ROM
R12 [▶ 40]	Minimum data length	0x6060	24672_{dec}	R	ROM
R13 [▶ 40]	Data structure of the Bus Terminal	0x0001	1_{dec}	R	ROM
R14	reserved	-	-	-	-
R15 [▶ 40]	Alignment register	-	-	R/W	RAM
R16 to R26	reserved	0x0000	0_{dec}	-	-
R27	reserved	0x0001	1_{dec}	R/W	SEEROM/RAM
R28 to R30	reserved	0x0000	0_{dec}	-	-
R31 [▶ 40]	Code word register (not used)	0x0000	0_{dec}	R/W	RAM
R32 to R63 [▶ 40]	Register for displaying the register pages [▶ 45]	-	-	-	-

5.4 Register description

The registers are used for parameterization of the terminal and can be read or written via the register communication [▶ 41].

Registers 0 to 31 always have the same meaning. The contents of registers 32 to 63 are specified via the register page selection register (R4 [▶ 40]).

R4: Register page selection register

This register specifies which register page [▶ 45] is displayed in registers R32 to R63 of the KL6224 (default: 0x0000).
The register pages provide access to the IO-Link configuration, diagnostics and parameterization of the KL6224.

R8: Terminal description

Register R8 contains the terminal identifier in hexadecimal coding: 0x1850 (6224_{dec})

R9: Firmware version

Register R9 contains the firmware revision level of the terminal in hexadecimal coding, e. g. 0x0100 (256_{dec}).

R10: Shift register length

0x0160

R11: Number of signal channels

0x0160

R12: Minimum data length

0x6060

R13: Data structure of the Bus Terminal

Register R13 contains the data structure of the Bus Terminal.

R15: Alignment register

R31: Code word register

The code word register of the KL6224 has no function for the user, since all settings that are specified via registers R32 to R63 (register pages) immediately take effect in the KL6224.

Register pages

In registers R32 to R64 the KL6224 shows the IO-Link configuration, diagnostics and parameterization of the register page [▶ 45] selected with register R4 [▶ 40].

Register page 2 (RP2)

Write 2 into the register page selection register R4 [▶ 40] to facilitate access to register RP2.R32.

RP2.R32

This register specifies the size of the process image of the KL6224.

Value	Process image	Default
0002_{hex}	12-byte process image [▶ 34] (6 bytes parameter interface and 6 bytes IO-Link process data)	0004_{hex}
0004_{hex}	24-byte process image [▶ 35]	
0005_{hex}	36-byte process image [▶ 36]	
0006_{hex}	48-byte process image [▶ 36]	
further	reserved	

After changing the process image the Bus Coupler must be restarted for the modified process image to take effect.

Register pages 4 to 9

See chapter Register pages for IO-Link parameters [▶ 45].

5.5 Examples of Register Communication

The numbering of the bytes in the examples corresponds to the display without word alignment.

5.5.1 Example 1: reading the firmware version from Register 9 of a terminal

Output Data

Byte 0: Control byte	Byte 1: DataOUT1, high byte	Byte 2: DataOUT1, low byte
0x89 (1000 1001_{bin})	0xXX	0xXX

Explanation:

- Bit 0.7 set means: Register communication switched on.
- Bit 0.6 not set means: reading the register.
- Bits 0.5 to 0.0 specify the register number 9 with 00 1001_{bin}.
- The output data word (byte 1 and byte 2) has no meaning during read access. To change a register, write the required value into the output word.

Input Data (answer of the bus terminal)

Byte 0: Status byte	Byte 1: DataIN1, high byte	Byte 2: DataIN1, low byte
0x89	0x33	0x41

Explanation:

- The terminal returns the value of the control byte as a receipt in the status byte.
- The terminal returns the firmware version 0x3341 in the input data word (byte 1 and byte 2). This is to be interpreted as an ASCII code:
  - ASCII code 0x33 represents the digit 3
  - ASCII code 0x41 represents the letter A
    The firmware version is thus 3A.

5.5.2 Example 2: Writing to an user register

Note	**Code word** In normal mode all user registers are read-only with the exception of Register 31. In order to deactivate this write protection you must write the code word (0x1235) into Register 31. If a value other than 0x1235 is written into Register 31, write protection is reactivated. Please note that changes to a register only become effective after restarting the terminal (power-off/power-on).

I. Write the code word (0x1235) into Register 31.

Output Data

Byte 0: Control byte	Byte 1: DataOUT1, high byte	Byte 2: DataOUT1, low byte
0xDF (1101 1111_{bin})	0x12	0x35

Explanation:

- Bit 0.7 set means: Register communication switched on.
- Bit 0.6 set means: writing to the register.
- Bits 0.5 to 0.0 specify the register number 31 with 01 1111_{bin}.
- The output data word (byte 1 and byte 2) contains the code word (0x1235) for deactivating write protection.

Input Data (answer of the bus terminal)

Byte 0: Status byte	Byte 1: DataIN1, high byte	Byte 2: DataIN1, low byte
0x9F (1001 1111_{bin})	0xXX	0xXX

Explanation:

- The terminal returns a value as a receipt in the status byte that differs only in bit 0.6 from the value of the control byte.
- The input data word (byte 1 and byte 2) is of no importance after the write access. Any values still displayed are invalid!

II. Read Register 31 (check the set code word)

Output Data

Byte 0: Control byte	Byte 1: DataOUT1, high byte	Byte 2: DataOUT1, low byte
0x9F (1001 1111_{bin})	0xXX	0xXX

Explanation:

- Bit 0.7 set means: Register communication switched on.
- Bit 0.6 not set means: reading the register.
- Bits 0.5 to 0.0 specify the register number 31 with 01 1111_{bin}.
- The output data word (byte 1 and byte 2) has no meaning during read access.

Input Data (answer of the bus terminal)

Byte 0: Status byte	Byte 1: DataIN1, high byte	Byte 2: DataIN1, low byte
0x9F (1001 1111_{bin})	0x12	0x35

Explanation:

- The terminal returns the value of the control byte as a receipt in the status byte.

- The terminal returns the current value of the code word register in the input data word (byte 1 and byte 2).

III. Write to Register 32 (change contents of the feature register)

Output Data

Byte 0: Control byte	Byte 1: DataIN1, high byte	Byte 2: DataIN1, low byte
0xE0 (1110 0000_{bin})	0x00	0x02

Explanation:

- Bit 0.7 set means: Register communication switched on.
- Bit 0.6 set means: writing to the register.
- Bits 0.5 to 0.0 indicate register number 32 with 10 0000_{bin}.
- The output data word (byte 1 and byte 2) contains the new value for the feature register.

CAUTION

Observe the register description!

The value of 0x0002 given here is just an example! The bits of the feature register change the properties of the terminal and have a different meaning, depending on the type of terminal. Refer to the description of the feature register of your terminal (chapter *Register description*) regarding the meaning of the individual bits before changing the values.

Input Data (answer of the bus terminal)

Byte 0: Status byte	Byte 1: DataIN1, high byte	Byte 2: DataIN1, low byte
0xA0 (1010 0000_{bin})	0xXX	0xXX

Explanation:

- The terminal returns a value as a receipt in the status byte that differs only in bit 0.6 from the value of the control byte.
- The input data word (byte 1 and byte 2) is of no importance after the write access. Any values still displayed are invalid!

IV. Read Register 32 (check changed feature register)

Output Data

Byte 0: Control byte	Byte 1: DataOUT1, high byte	Byte 2: DataOUT1, low byte
0xA0 (1010 0000_{bin})	0xXX	0xXX

Explanation:

- Bit 0.7 set means: Register communication switched on.
- Bit 0.6 not set means: reading the register.
- Bits 0.5 to 0.0 indicate register number 32 with 10 0000_{bin}.
- The output data word (byte 1 and byte 2) has no meaning during read access.

Input Data (answer of the bus terminal)

Byte 0: Status byte	Byte 1: DataIN1, high byte	Byte 2: DataIN1, low byte
0xA0 (1010 0000_{bin})	0x00	0x02

Explanation:

- The terminal returns the value of the control byte as a receipt in the status byte.
- The terminal returns the current value of the feature register in the input data word (byte 1 and byte 2).

V. Write Register 31 (reset code word)

Output Data

Byte 0: Control byte	Byte 1: DataOUT1, high byte	Byte 2: DataOUT1, low byte
0xDF (1101 1111_{bin})	0x00	0x00

Explanation:

- Bit 0.7 set means: Register communication switched on.
- Bit 0.6 set means: writing to the register.
- Bits 0.5 to 0.0 specify the register number 31 with 01 1111_{bin}.
- The output data word (byte 1 and byte 2) contains 0x0000 for reactivating write protection.

Input Data (answer of the bus terminal)

Byte 0: Status byte	Byte 1: DataIN1, high byte	Byte 2: DataIN1, low byte
0x9F (1001 1111_{bin})	0xXX	0xXX

Explanation:

- The terminal returns a value as a receipt in the status byte that differs only in bit 0.6 from the value of the control byte.
- The input data word (byte 1 and byte 2) is of no importance after the write access. Any values still displayed are invalid!

5.6 Register pages for IO-Link parameters

The parameters of the KL6224 can be accessed via register communication or via the KS2000 [▶ 21] configuration software. The register model of the terminals is used as a basis. The KL6224 has 64 registers (words). Registers 0 to 31 always have the same meaning. The contents of registers 32 to 63 are specified via the register page selection register (R4 [▶ 40]).

A register page has 32 registers (64 bytes). The following register pages are used:

- Register page 4 [▶ 46]: Parameter(s) for channel 1 (IO-Link device 1)
- Register page 5: Parameter(s) for channel 2 (IO-Link device 2)
- Register page 6: Parameter(s) for channel 3 (IO-Link device 4)
- Register page 7: Parameter(s) for channel 4 (IO-Link device 4)
- Register page 8 [▶ 48]: CMD/status interface and acyclic data for IO-Link devices
- Register page 9 [▶ 48]: acyclic data

Assignment of the IO-Link parameters to the register page and the terminal register

Register page 4 (RP4): Parameter(s) for channel 1 (IO-Link device 1)

KL6224 registers	Description	Name		Comment			
RP4.R32	IO-Link master configuration: The master parameters for operating an IO-Link device are entered here (similar to object 0x8000 of the EL6224).	IO-Link DeviceID LowWord		Device ID of the IO-Link device			
RP4.R33		IO-Link DeviceID HighWord					
RP4.R34		IO-Link VendorID LowWord		Vendor ID of the IO-Link device			
RP4.R35		IO-Link VendorID HighWord					
RP4.R36		LowByte	IO-Link revision: Bit 0...3: MinorRev Bit 4...7: MajorRev	ID of the specification version based on which the IO-Link device communicates.			
		HighByte	FrameCapability: Bit 0: SPDU Bit 1: Type1 Bit 7: PHY1	The Frame Capability indicates certain functionalities of the IO-Link device (e.g. SPDU supported).			
RP4.R37		LowByte	minCycleTime: Bit 6, 7: Time Base Bit 0...5: Multiplier	See table below (time base [▶ 46])			
		HighByte	OffsetTime	reserved			
RP4.R38		LowByte	ProcDataLenght Input (in IO-Link format)	Bit 7	Byte	0_{bin}	Length is counted in bits
						1_{bin}	Length is counted in bytes
				Bit 6	SIO	1_{bin}	The device supports the standard IO mode
				Bit 0...4	Length		Length of the process data
		HighByte	ProcDataLenght Output (in IO-Link format)	see LowByte			
RP4.R39		CompatibleId		Currently not used			
RP4.R40		-		reserved			
RP4.R41		MasterControl		0	Channel 1 is inactive		
				1	Channel 1 is digital input		
				2	Channel 1 is digital output		
				3	Channel 1 is IO-Link port and communicates via the IO-Link protocol		
				4	reserved		
				5	Channel 1 is IO-Link port and communicates via the IO-Link protocol with compressed mode [▶ 34]. Note: If compressed mode is active for channel 1, it is active for all channels that operated as IO-Link ports.		

Time Base

Time Base	Time base meaning	Calculation	Min. Cycle Time
00_{bin}	0.1 ms	Multiplier x Time Base	0.0...6.3 ms
01_{bin}	0.4 ms	6.4 ms + Multiplier x Time Base	6.4...31.6 ms
10_{bin}	1.6 ms	32.0 ms + Multiplier x Time Base	32.0...132.8 ms
11_{bin}	6.4 ms	134.4 ms + Multiplier x Time Base	134.4...537.6 ms

KL6224 registers	Description	Name		Comment			
RP4.R42	IO-Link actual data: The actual data of the connected IO-Link device is displayed here. This is used for comparison with the master parameters described above (similar to object 0x9000 of the EL6224).	IO-Link DeviceID LowWord		Device ID of the IO-Link device			
RP4.R43		IO-Link DeviceID HiWord					
RP4.R44		IO-Link VendorID LowWord		Vendor ID of the IO-Link device			
RP4.R45		IO-Link VendorID HiWord					
RP4.R46		LowByte	IO-Link revision Bit 0...3: MinorRev Bit 4...7: MajorRev	ID of the specification version based on which the IO-Link device communicates.			
		HighByte	FrameCapability: Bit 0: SPDU Bit 1: Type1 Bit 7: PHY1	The Frame Capability indicates certain functionalities of the IO-Link device (e.g. SPDU supported).			
RP4.R47		LowByte	minCycleTime: Bit 6, 7: Time Base Bit 0...5: Multiplier	See table below (time base [▶ 47])			
		HighByte	OffsetTime	reserved			
RP4.R48		LowByte	ProcDataLenght Input (in IO-Link format)	Bit 7	Byte	0_{bin}	Length is counted in bits
						1_{bin}	Length is counted in bytes
				Bit 6	SIO	1_{bin}	The device supports the standard IO mode
				Bit 0...4	Length		Length of the process data
		HighByte	ProcDataLenght Output (in IO-Link format)	see LowByte			
RP4.R49		-		reserved			
RP4.R50		-		reserved			
RP4.R51		-		reserved			

Time Base	Time base meaning	Calculation	Min. Cycle Time
00_{bin}	0.1 ms	Multiplier x Time Base	0.0...6.3 ms
01_{bin}	0.4 ms	6.4 ms + Multiplier x Time Base	6.4...31.6 ms
10_{bin}	1.6 ms	32.0 ms + Multiplier x Time Base	32.0...132.8 ms
11_{bin}	6.4 ms	134.4 ms + Multiplier x Time Base	134.4...537.6 ms

KL6224 registers	Description	Name		Comment	
RP4.R52	IO-Link diagnostics	LowByte	IO-Link state The value of the IO-Link state corresponds to a state from the IO-Link master state machine (similar to object 0xA000 of the EL6224).	0	Inactive
				1	DigInput
				2	DigOutput
				3	EstablishComm
				4	InitMaster
				5	InitDevice
				6	reserved
				7	reserved
				8	Operate
				9	Stop
		HighByte	LostFrames	This parameter counts the number of lost IO-Link telegrams. This value is deleted whenever IO-Link starts up, otherwise it is incremented continuously.	

Register page 5 (RP5): Parameter(s) for channel 2 (IO-Link device 2)

Structure like register page 4 [▶ 46]

Register page 6 (RP6): Parameter(s) for channel 3 (IO-Link device 3)

Structure like register page 4 [▶ 46]

Register page 7 (RP7): Parameter(s) for channel 4 (IO-Link device 4)

Structure like register page 4 [▶ 46]

Register page 8 (RP8): CMD/status interface and acyclic data for IO-Link device

KL6224 registers	Description	Name		Comment
RP8.R32	Cmd	TACYCLICKBUSCMD_CMD_START		0x00000001
		TACYCLICKBUSCMD_CMD_ACK		0x00000002
		TACYCLICKBUSCMD_CMD_READ_CH1		0x00000010
		TACYCLICKBUSCMD_CMD_READ_CH2		0x00000020
		TACYCLICKBUSCMD_CMD_READ_CH3		0x00000030
		TACYCLICKBUSCMD_CMD_READ_CH4		0x00000040
		TACYCLICKBUSCMD_CMD_WRITE_CH1		0x00000100
		TACYCLICKBUSCMD_CMD_WRITE_CH2		0x00000200
		TACYCLICKBUSCMD_CMD_WRITE_CH3		0x00000300
		TACYCLICKBUSCMD_CMD_WRITE_CH4		0x00000400
		TACYCLICKBUSCMD_CMD_SCAN_CH1		0x00001000
		TACYCLICKBUSCMD_CMD_SCAN_CH2		0x00002000
		TACYCLICKBUSCMD_CMD_SCAN_CH3		0x00003000
		TACYCLICKBUSCMD_CMD_SCAN_CH4		0x00004000
RP8.R33	Status	TACYCLICKBUSCMD_STATUS_IDLE		0x00000000
		TACYCLICKBUSCMD_STATUS_BUSY		0x00000001
		TACYCLICKBUSCMD_STATUS_DATAREADY		0x00000002
		TACYCLICKBUSCMD_STATUS_DATAERROR		0x00000003
RP8.R34		SPDU / ISDU SubIdx		SPDU / ISDN index
RP8.R35		LowByte	Length	Length of the data
		HighByte	SPDU / ISDU SubIdx	SPDU / ISDN subindex
RP8.R36	Acyclic data for the IO-Link device			
...				
RP8.R61				

Register page 9 (RP9): Further acyclic data for the IO-Link device

KL6224 registers	Description	Value	Comment
RP9.R0	Further acyclic data for the IO-Link device		
...			
RP9.R61			

Process of an acyclic query

The process of an acyclic query to an IO-Link device is as follows:

1. Reading of register page 8 R33 (status) to ascertain whether status is Idle
2. If status = Idle, write TACYCLICKBUSCMD_CMD_START to register page 8 R32 (Cmd)
3. Write acyclic data into the buffer from register page 8, R36, as required
4. Start the service by writing the corresponding services in register page 8 R32 (Cmd)
5. Read register page 8 R33 (status) until no longer busy
6. Read acyclic data from the buffer from register page 8, R36, as required
7. Conclude the service by writing TACYCLICKBUSCMD_CMD_ACK to register page 8 R32 (Cmd)

5.7 Accessing IO-Link parameters

The parameters of the KL6224 IO-Link master terminal can be accessed via the parameter data block. Write and read access are documented below, including examples.

Writing a parameter

The following sequence should be used for writing a parameter:

First check whether the previous access was fully completed. To do this, assess status byte 1: Bits 4 to 7 should be 0_{bin}. If this is not the case, control bytes 0 and 1 should be set to 0, until bits 4 to 7 in status byte 1 are set to 0_{bin}.

Write access (PLC->KL6224): parameter data block

Byte	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Name	Control byte 0	Control byte 1	ParaOut0	ParaOut1	ParaOut2	ParaOut3
Value	$01_{bin}A_5A_4\ A_3A_2A_1A_0$	$0100_{bin}\ A_9A_8A_7A_6$	$P_0...P_7$	$P_8...P_{15}$	$P_{16}...P_{23}$	$P_{24}...P_{31}$

A_0 to A_9: bits of the parameter address
P_0 to P_{31}: bits of the parameter value

Response to write access (KL6224->PLC): parameter data block

Byte	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Name	Status byte 0	Status byte 1	ParaIn0	ParaIn1	ParaIn2	ParaIn3
Value	$xxxx\ xxxx_{bin}$	$01F1\ xxx1_{bin}$	Error code	Error code	Error code	Error code

x: Bits can take on any value
F: error bit.
$F=0_{bin}$: Write access was successful.
$F=1_{bin}$: Write access was not successful. Bytes 2 to 5 contain an error code providing information about the cause of the error.

The write sequence is completed by setting the control bytes to zero:

Conclusion of write access (PLC->KL6224): parameter data block

Byte	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Name	Control byte 0	Control byte 1	ParaOut0	ParaOut1	ParaOut2	ParaOut3
Value	$0000\ 0000_{bin}$	$0000\ 0000_{bin}$	x	x	x	x

x: The parameter values are not evaluated if the control bytes are 0x00.

Example

The list of currently projected slaves (LPS) is to be written. The IO-Link master is to communicate specifically with the IO-Link slaves with node numbers 1, 2, 3, 4, 12, 16, 17 and 30. In other words, the value 0x4003101E (0100 0000 0000 0011 0001 0000 0001 1110_{bin}) is to be written to parameter 0xA8 (1010 1000_{bin}).

Write access (PLC->KL6224): parameter data block

Byte	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Name	Control byte 0	Control byte 1	ParaOut0	ParaOut1	ParaOut2	ParaOut3
Value	0110 1000_{bin} (0x68)	0100 0010_{bin} (0x42)	0x1E	0x10	0x03	0x40

Therefore the byte sequence 0x68 42 1E 10 03 40 has to be written in the parameter data block for the KL6224.

The terminal responds with the following data:

Response to write access (KL6224->PLC): parameter data block

Byte	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Name	Status byte 0	Status byte 1	ParaIn0	ParaIn1	ParaIn2	ParaIn3
Value	xxxx $xxxx_{bin}$	0101 $xxx1_{bin}$	0	0	0	0

The write sequence is completed with this byte sequence: 0x00 00 00 00 00 00

Reading a parameter

The following sequence should be used for reading a parameter:

First check whether the previous access was fully completed. This requires status 1 to be evaluated - bits 4 to 7 should be 0. If this is not the case, control bytes 0 and 1 should be set to 0, until bits 4 to 7 in status 1 are set to 0.

Read access (PLC->KL6224): parameter data block

Byte	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Name	Control byte 0	Control byte 1	ParaOut0	ParaOut1	ParaOut2	ParaOut3
Value	00_{bin} A_5A_4 $A_3A_2A_1A_0$	0100_{bin} $A_9A_8A_7A_6$	x	x	x	x

A_0 to A_9: bits of the parameter address

x: The parameter values are not evaluated

Response to read access (KL6224->PLC): parameter data block

Byte	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Name	Status byte 0	Status byte 1	ParaIn0	ParaIn1	ParaIn2	ParaIn3
Value	xxxx $xxxx_{bin}$	01F1 $xxx0_{bin}$	$P_0...P_7$	$P_8...P_{15}$	$P_{16}...P_{23}$	$P_{24}...P_{31}$

x: Bits can take on any value
F: error bit.
F=0_{bin}: Read access was successful. Bytes 2 to 5 contain the parameter value.
F=1_{bin}: Read access was not successful. Bytes 2 to 5 contain an error code providing information about the cause of the error.
P_0 to P_{31}: bits of the parameter value or error code

The read sequence is completed by setting the control bytes to zero:

Conclusion of read access (PLC->KL6224): parameter data block

Byte	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Name	Control byte 0	Control byte 1	ParaOut0	ParaOut1	ParaOut2	ParaOut3
Value	0000 0000_{bin}	0000 0000_{bin}	x	x	x	x

x: The parameter values are not evaluated if the control bytes are 0x00.

Example

The list of detected slaves (LDS) is to be determined. This requires the parameter 0xB0 (1011 0000_{bin}) to be read.

Read access (PLC->KL6224): parameter data block

Byte	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Name	Control byte 0	Control byte 1	ParaOut0	ParaOut1	ParaOut2	ParaOut3
Value	0011 0000_{bin} (0x30)	0100 0010_{bin} (0x42)	0x00	0x00	0x00	0x00

Therefore the byte sequence 0x30 42 00 00 00 00 has to be written in the parameter data block for the KL6224. The terminal responds with the following data:

Response to write access (KL6224->PLC): parameter data block

Byte	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Name	Status byte 0	Status byte 1	ParaIn0	ParaIn1	ParaIn2	ParaIn3
Value	xxxx $xxxx_{bin}$	0101 $xxx0_{bin}$	0x4C	0x02	0x80	0x83

In this example the IO-Link master should have detected the slaves with node numbers 2, 3, 6, 9, 23, 24, 25 and 31. It therefore responds with the parameter data 0x8380024C
(1000 0011 1000 0000 0000 0010 0100 1100_{bin})

The read sequence is completed with the byte sequence 0x00 00 00 00 00 00.

5.8 Firmware version of the Bus Couplers

Note	**Required firmware** A particular firmware version may be required for operating the KL6224 IO-Link terminal on the Bus Coupler / Bus Terminals Controller (see table [▶ 53] below)

In delivery state the KL6224 is set to a process image of 24 bytes.

Most Bus Couplers and Bus Terminal Controllers listed in the table support the 12-byte process image. Bus Couplers / Bus Terminals Controllers that are not listed are not yet ready for operation with 24-byte process images or higher.

The delivered firmware version is shown on the back of the Bus Coupler (see example below for CANopen).

Fig. 30: Specification of the firmware version on a Bus Coupler

If required, the firmware can be updated via the serial port (KS2000 cable required) or - depending on the bus system - via the fieldbus. The current firmware versions and the program for the firmware update can be found on our home page at http://www.beckhoff.de.

Support of the different process images by the firmware versions of the Bus Couplers / Bus Terminals Controllers

Fieldbus system	Bus Coupler / Bus Terminal Controller	Firmware version required on the Bus Coupler / Bus Terminal Controller for the			
		12-byte process image	24-byte process image	36-byte process image	48-byte process image
EtherCAT	BK1120	These Bus Couplers and Bus Terminal Controllers support the 12-byte process image of the KL6204.	from 08 (B8)	from 08 (B8)	from 08 (B8)
	BK1250		all	all	all
Lightbus	BK2020		from B1	in preparation	in preparation
PROFIBUS	BK3120		from B9	from BB	from BB
	BK3150		all	all	all
	BK3500		from B9	from BB	-
	BK3520		from B9	from BB	-
	BC3100		from C3	from C4	-
	BC3150		all	all	all
	BX3100		all	all	all
Interbus	BK4020		from B0	in preparation	in preparation
	BC4000		from B3	in preparation	in preparation
CANopen	BK5120		from C4	from C5	-
	BK5150		all	all	-
	BC5150		all	all	all
	BX5100		all	all	all
DeviceNet	BK5220		(from B3)*	in preparation	-
	BC5250		all	all	all
	BX5200		all	all	all
ControlNet	BK7000		from BC	from BC	-
Modbus	BK7300		from B2	from B4	-
Fip IO	BK7420		from B1	from B1	-
RS485	BK8000		from C2	in preparation	in preparation
RS232	BK8100		from C2	in preparation	in preparation
	BC8150		all	all	all
	BX8000		all	all	all
Ethernet	BK9000		from B7	from BA	from BA
	BK9100		from B1	from B1	from B1
	BC9000		from B9	from BB	from BB
	BC9100		from B1	from B1	from B1
	BC9050		all	all	all
	BX9000		all	all	all
PROFINET	BK9103	in preparation	in preparation	in preparation	in preparation
EtherNet/IP	BK9105	all	all	all	all
USB	BK9500	This Bus Coupler supports the 12-byte process image of the KL6204.	(from B1)*	in preparation	in preparation

*) only if the Bus Coupler is set to *complete mapping of the Bus Terminals* (e.g. via the KS2000 configuration software). On delivery, these Bus Couplers are set to *compact mapping of the Bus Terminals*.

6 Appendix

6.1 Support and Service

Beckhoff and their partners around the world offer comprehensive support and service, making available fast and competent assistance with all questions related to Beckhoff products and system solutions.

Beckhoff's branch offices and representatives

Please contact your Beckhoff branch office or representative for local support and service on Beckhoff products!

The addresses of Beckhoff's branch offices and representatives round the world can be found on her internet pages:
http://www.beckhoff.com

You will also find further documentation for Beckhoff components there.

Beckhoff Headquarters

Beckhoff Automation GmbH & Co. KG

Huelshorstweg 20
33415 Verl
Germany

Phone:	+49(0)5246/963-0
Fax:	+49(0)5246/963-198
e-mail:	info@beckhoff.com

Beckhoff Support

Support offers you comprehensive technical assistance, helping you not only with the application of individual Beckhoff products, but also with other, wide-ranging services:

- support
- design, programming and commissioning of complex automation systems
- and extensive training program for Beckhoff system components

Hotline:	+49(0)5246/963-157
Fax:	+49(0)5246/963-9157
e-mail:	support@beckhoff.com

Beckhoff Service

The Beckhoff Service Center supports you in all matters of after-sales service:

- on-site service
- repair service
- spare parts service
- hotline service

Hotline:	+49(0)5246/963-460
Fax:	+49(0)5246/963-479
e-mail:	service@beckhoff.com

List of illustrations