Documentation

KL85xx und KL9309

Manual operating modules with K-Bus interface

Version: 2.1.0
Date: 2018-02-27

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
2.1.0	• Chapter Technical data for KL8548 updated
2.0.0	• Migration • Chapter *Function blocks* extended • Chapter *Adjustment of LED displays* added
1.5.0	• Description for disabling manual operation via the PLC program added • Description of control and status bytes for KL8524 updated • Description of control and status bytes for KL8528 updated • Description of control and status bytes for KL8548 updated
1.4.0	• Labeling for power supply (Us) of the KL85xx corrected
1.3.0	• Foreword updated • Programming description updated • Register description for KL8519 updated • Register description for KL8524 updated • Technical data of KL9020 and KL9309 updated
1.2.0	• Introduction for manual operating modules updated • Description of Status Byte corrected • Chapter *Connection* extended • Libraries updated
1.1.0	• Register descriptions updated • Technical data updated
1.0.0	First release

Firmware and hardware versions

Documentation Version	KL8500		KL8519		KL8524		KL8528		KL8548		KL9309	
	Firmw.	Hardw.	Firmw.	Hardw.	Firmw.	Hardw.	Firmw.	Hardw.	Firmw.	Hardw.	Firmw.	Hardw.
2.0.0	-	00	1C	03	1B	01	1B	03	1F	03	00	00
1.5.0	-	00	1C	03	1B	01	1B	03	1F	03	00	00
1.4.0	-	00	1C	03	1A	01	1A	03	1E	03	00	00
1.3.0	-	00	1C	03	1A	01	1A	02	1E	03	00	00
1.2.0	-	00	1C	03	1A	01	1A	02	1D	03	00	00
1.1.0	-	00	1C	00	1A	00	1A	00	1C	01	00	00
1.0.0	-	00	1C	00	1A	00	1A	00	1C	01	00	00

The firmware and hardware versions (delivery state) of the KL85xx can be found in the serial number printed on the back.
The firmware and hardware versions (delivery state) of the KL9309 can be found in the serial number printed on the side.

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.: 33 06 B1 00:

33 - week of production 33
06 - year of production 2006
B1 - firmware version B1
00 - hardware version 00

2 Product overview

2.1 Introduction

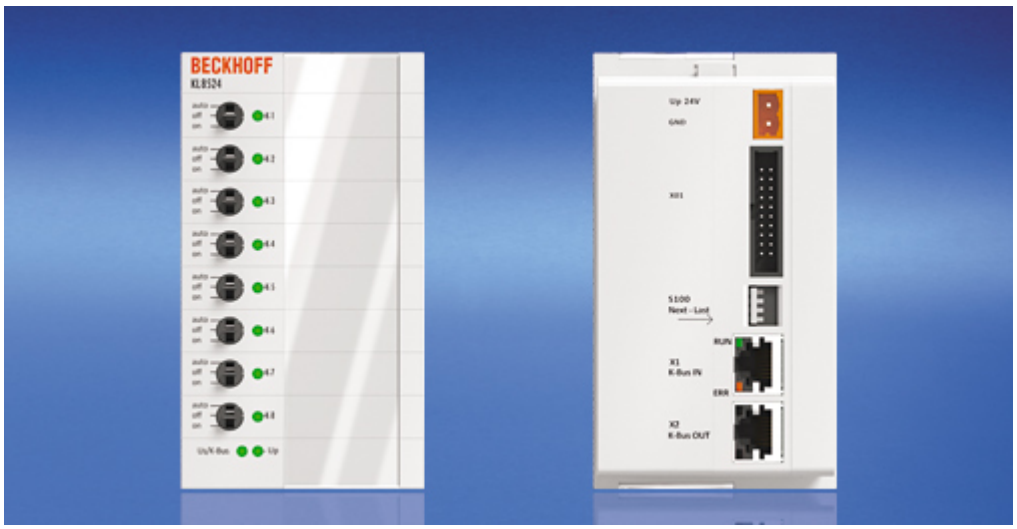


Fig. 1: Manual operating module with K-bus interface

Manual operating modules with K-bus interface

The manual operating modules have been developed for the switching, controlling and observation of digital and analog signals.
They enable the setting and reading of data and values in case of failure of a controller.

The manual operating modules provide the functionality of a local priority control/display unit according to EN ISO 16484-2:2004, Chapter 5.4.3.
The equipment manufacturer must ensure access protection against unauthorized operation by installing suitable components (see EN ISO 16484-2:2004, Chapter 5.4.3.3. - Access protection for local priority control/display units).

Manual operation can be disabled via the user program. See chapter "Control and status bytes" of the respective module.

The manual operating modules can be installed in the control cabinet door using a snap-in technique; they are wired inside the control cabinet.
Up to 31 modules can be inserted via the K-bus interface with K-bus extension. This facilitates integration of the manual operating modules in the universal bus terminal system.

Connection to the KL9309 [▶ 22]signal-independent transfer terminal takes place via ribbon cable. The signals are electrically isolated. Power and error LEDs indicate the status of the modules.

By means of installation in the control cabinet door, the modules can be operated without having to open the control cabinet door.

A total of five different modules are available:

- KL8500 [▶ 10] | Placeholder module
- KL8519 [▶ 12] | Signal module, 16-channel digital input
- KL8524 [▶ 14] | Output module, 4 x 2-channel digital output
- KL8528 [▶ 16] | Output module, 8-channel digital output
- KL8548 [▶ 18] | Analog module, 8-channel analog output

Note	**TwinCAT Version** The manual operating modules are supported from TwinCAT 2.11 R3 (build 2221) or higher. Older TwinCAT versions may also be able to support the manual operating modules, if a current bus terminal configuration file (TcTerminals) is used. The current bus terminal configuration file is available from the Beckhoff website under Download/Configuration files/Bus Terminals.

2.2 KL8500

2.2.1 KL8500 – Introduction

Fig. 2: KL8500

Placeholder module (no function)

You can install the KL8500 placeholder module in your control cabinet in order to prepare for future extensions.
You can then replace it later by another manual operating module without any further sheet metal work being necessary.

2.2.2 Technical data

Technical data	KL8500
Number of inputs/outputs	0
Bus interface	-
I/O connection	-
Diagnostic LEDs	none
Electrical isolation	-
Power consumption Us	-
Output process image	-
Input process image	-
Weight	approx. 90 g
Dimensions (W x H x D)	71 mm x 127.5 mm x 75 mm (external dimensions; see Mounting [▶ 26] for installation dimensions)
Mounting	Installation in the control cabinet door
Permissible ambient temperature range during operation	0°C ... + 55°C
Permissible ambient temperature range during storage	-25°C ... + 85°C
Permissible relative humidity	95 %, no condensation
Vibration / shock resistance	conforms to EN 60068-2-6 / EN 60068-2-27
EMC immunity / emission	conforms to EN 61000-6-2 / EN 61000-6-4
Protection class	IP20
Installation position	variable
Approval	CE

2.3 KL8519

2.3.1 KL8519 - Introduction

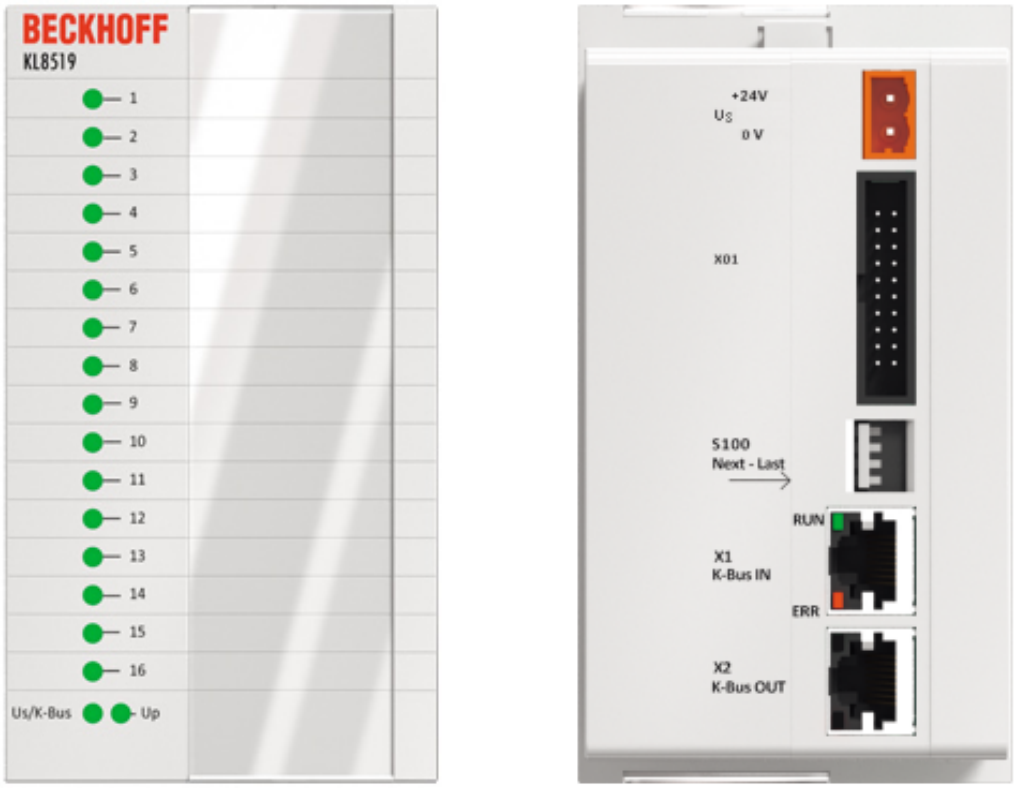


Fig. 3: KL8519

16-channel digital input signal module

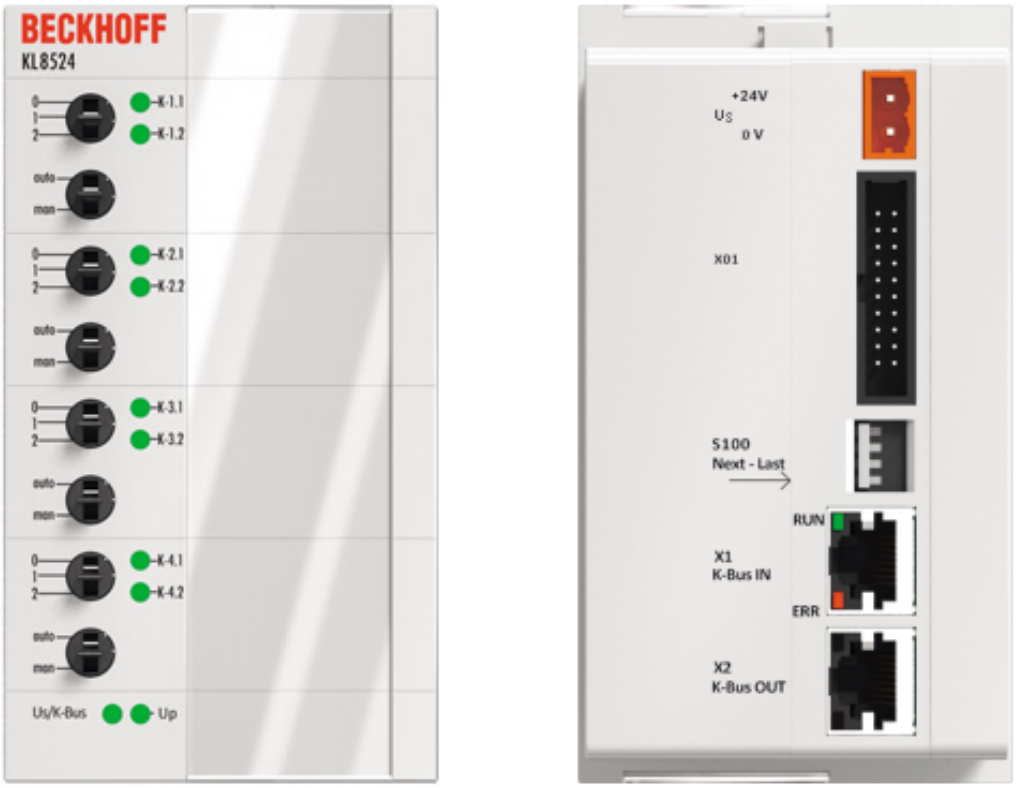
The KL8519 is a 16-channel digital input signal module. 16 digital inputs can be connected, which indicate their status via LEDs and transmit the data to the controller. The LEDs are bicolor LEDs in the colors red and green and can be parameterized individually to suit the needs of the plant. The LEDs can also be addressed by the controller.

2.3.2 Technical data

Technical data	KL8519
Number of inputs	16
Input filter	3.0 ms
Rated voltage	24 V_{DC} (-15 %/+20 %)
Bus interface	K-bus connection IN/OUT
I/O connection	Ribbon cable connection, 20-pin
Diagnostic LEDs	bicolor LEDs green/red
Electrical isolation	500 V (K-bus)
Power consumption Us	typically 50 mA
Output process image	6 bytes data, 1 control byte
Input process image	6 bytes data, 1 status byte
Weight	approx. 150 g
Dimensions (W x H x D)	71 mm x 127 mm x 69 mm (external dimensions, see Mounting for installation dimensions)
Mounting [▶ 26]	Installation in the control cabinet door
Permissible ambient temperature range during operation	0°C ... + 55°C
Permissible ambient temperature range during storage	-25°C ... + 85°C
Permissible relative humidity	95 %, no condensation
Vibration / shock resistance	conforms to EN 60068-2-6 / EN 60068-2-27
EMC immunity / emission	conforms to EN 61000-6-2 / EN 61000-6-4
Protection class	IP20
Installation position	variable
Approval	CE

2.4 KL8524

2.4.1 KL8524 - Introduction



Front view **Back view**

Fig. 4: KL8524

4 x 2-channel digital output module

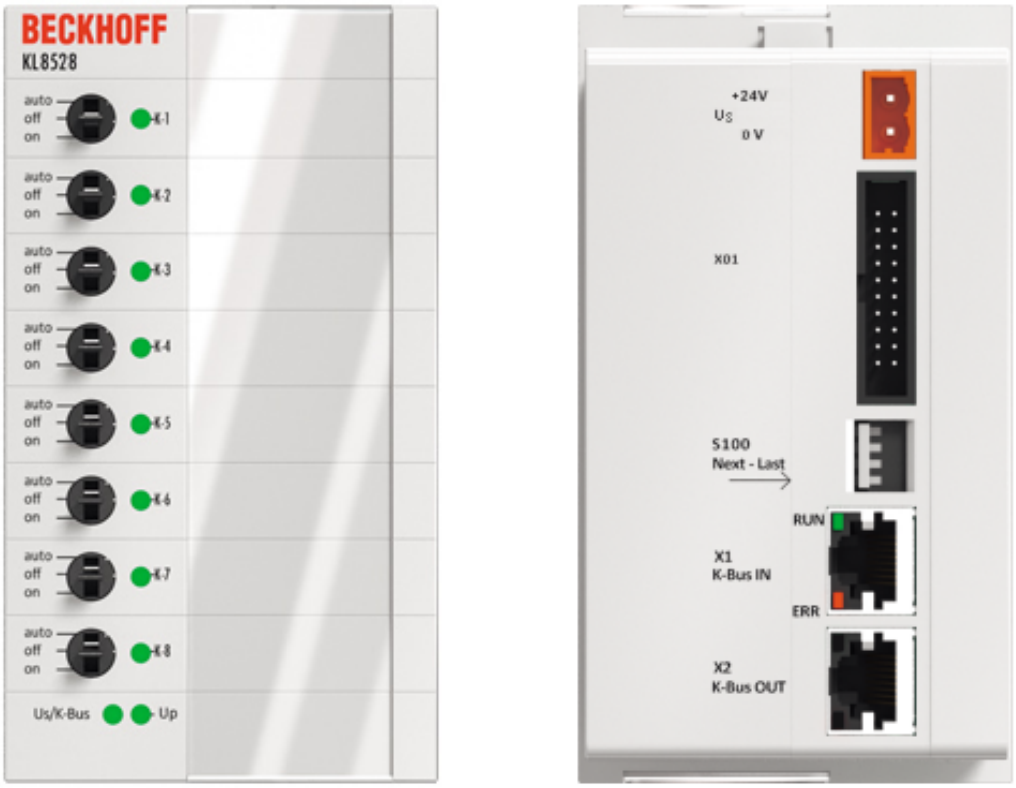
The KL8524 is a 4 x 2-channel digital output module, equipped with two switches. The first is for switching between manual and automatic operation, while the second is used to set a 2-stage output. It is possible to specify when and how the two outputs are switched. The status is indicated by a bicolor LED in green and yellow. The switching positions are readable via the PLC.

2.4.2 Technical data

Technical data	KL8524
Number of outputs	2 x 4
Output current	0.5 A (total current 2 A)
Rated voltage	24 V_{DC} (-15 %/+20 %)
Bus interface	K-bus connection IN/OUT
I/O connection	Ribbon cable connection, 20-pin
Diagnostic LEDs	bicolor LEDs green/yellow
Switch settings	4 x auto/manual; 4 x mode 0/1/2
Electrical isolation	500 V (K-bus)
Power consumption Us	typically 40 mA
Output process image	6 bytes data, 1 control byte
Input process image	6 bytes data, 1 status byte
Weight	approx. 160 g
Dimensions (W x H x D)	71 mm x 127 mm x 69 mm (external dimensions, see Mounting for installation dimensions)
Mounting [▶ 26]	Installation in the control cabinet door
Permissible ambient temperature range during operation	0°C ... + 55°C
Permissible ambient temperature range during storage	-25°C ... + 85°C
Permissible relative humidity	95 %, no condensation
Vibration / shock resistance	conforms to EN 60068-2-6 / EN 60068-2-27
EMC immunity / emission	conforms to EN 61000-6-2 / EN 61000-6-4
Protection class	IP20
Installation position	variable
Approval	CE

2.5 KL8528

2.5.1 KL8528 - Introduction



Front view **Back view**

Fig. 5: KL8528

8-channel digital output module

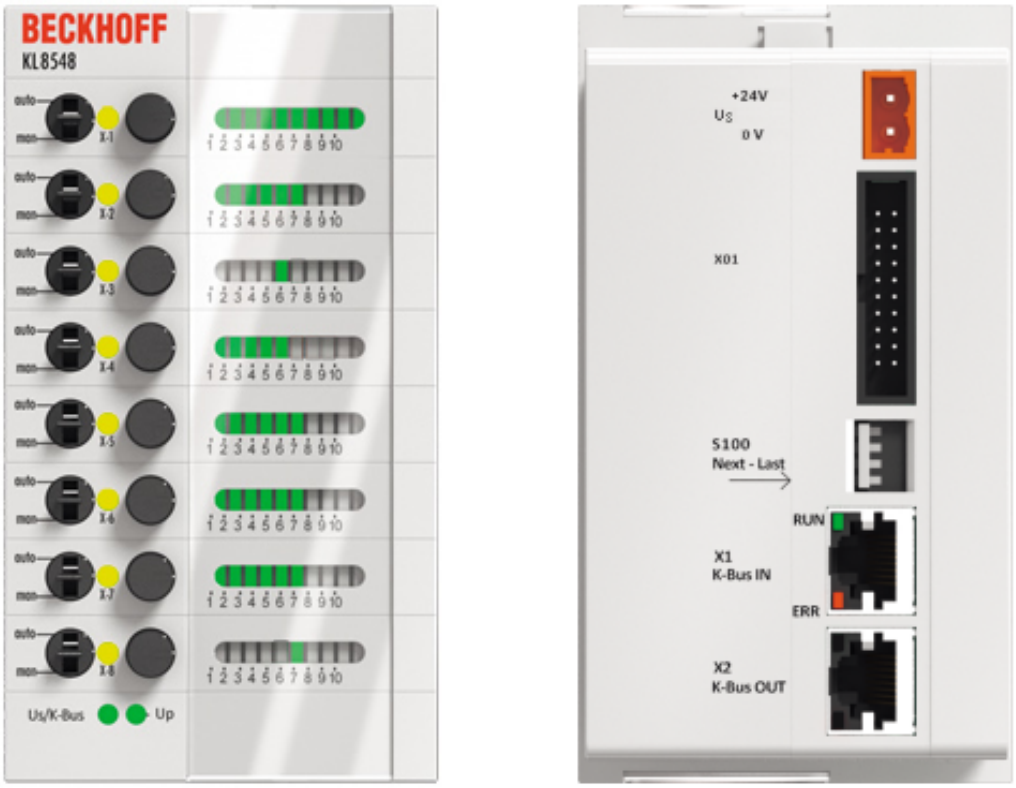
The KL8528 is an 8-channel digital output module. The outputs can be switched via a switch or specified by the controller. The status is indicated by a bicolor LED in green and yellow. The switching positions are readable via the PLC.

2.5.2 Technical data

Technical data	KL8528
Number of outputs	8
Output current	0.5 A (total current 2 A)
Rated voltage	24 V_{DC} (-15 %/+20 %)
Bus interface	K-bus connection IN/OUT
I/O connection	Ribbon cable connection, 20-pin
Diagnostic LEDs	bicolor LEDs green/yellow
Switch settings	8 x auto/off/on
Electrical isolation	500 V (K-bus)
Power consumption Us	typically 40 mA
Output process image	6 bytes data, 1 control byte
Input process image	6 bytes data, 1 status byte
Weight	approx. 160 g
Dimensions (W x H x D)	71 mm x 127 mm x 69 mm (external dimensions, see Mounting for installation dimensions)
Mounting [▶ 26]	Installation in the control cabinet door
Permissible ambient temperature range during operation	0°C ... + 55°C
Permissible ambient temperature range during storage	-25°C ... + 85°C
Permissible relative humidity	95 %, no condensation
Vibration / shock resistance	conforms to EN 60068-2-6 / EN 60068-2-27
EMC immunity / emission	conforms to EN 61000-6-2 / EN 61000-6-4
Protection class	IP20
Installation position	variable
Approval	CE

2.6 KL8548

2.6.1 KL8548 - Introduction



Front view **Back view**

Fig. 6: KL8548

8-channel analog output module 0…10 V

The KL8548 is an 8-channel analog output module for 0…10 V. The analog values must be specified individually for each channel via the controller or via a potentiometer. The actual output value is indicated by a bar graph. The position of the potentiometer is readable by the controller in each operation mode.

2.6.2 Technical data

Technical data	KL8548
Number of inputs	8 (potentiometer)
Number of outputs	8
Signal voltage	0...10 V
Load	> 5 kΩ (short circuit proof)
Accuracy	± 0.5 LSB linearity error, ± 0.5 LSB offset error
Output error	< ± 0.1 % (relative to full scale value)
A/D converter	12 bit
Rated voltage	24 V_{DC} (-15 %/+20 %)
Bus interface	K-bus connection IN/OUT
I/O connection	Ribbon cable connection, 20-pin
Diagnostic LEDs	yellow
Switch settings	auto/manual, potentiometer
Special features	Potentiometers and switches can be read via the PLC. Analog values are displayed in the form of bar graphs
Electrical isolation	500 V (K-bus)
Power consumption Us	typically 50 mA in ECO Mode, 95 mA in Full Scale Mode
Output process image	16 byte data, 8 control bytes
Input process image	16 byte data, 8 status bytes
Weight	approx. 215 g
Dimensions (W x H x D)	71 mm x 127 mm x 69 mm (external dimensions, see Mounting for installation dimensions)
Mounting [▶ 26]	Installation in the control cabinet door
Permissible ambient temperature range during operation	0°C ... + 55°C
Permissible ambient temperature range during storage	-25°C ... + 85°C
Permissible relative humidity	95 %, no condensation
Vibration / shock resistance	conforms to EN 60068-2-6 / EN 60068-2-27
EMC immunity / emission	conforms to EN 61000-6-2 / EN 61000-6-4
Protection class	IP20
Installation position	variable
Approval	CE

2.7 KL9020

2.7.1 KL9020: End terminal with K-bus extension

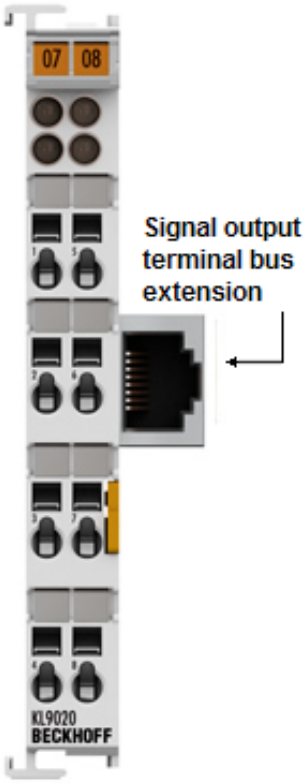


Fig. 7: KL9020

The KL9020 end terminal with K-bus extension is attached to the end of the Bus Terminal block in the same way as a standard end terminal KL9010 and terminates the block.

The KL9020 enables the connection of an Ethernet cable with a RJ-45 connector. The K-bus signals are converted to RS485. The supply is via the K-bus.

Apart from a supply voltage of 24 V and the insertion of the Ethernet cable, there is no further parameterization or configuration work necessary.
The Bus Coupler carries out all diagnosis and commissioning tasks.

Note	**Documentation for the K-bus extension** Further information may be found in the documentation for K-Bus Extension (KL9020/ KL9050), that is available at the download area of the Beckhoff homepage.

2.7.2 Technical data

Technical data	KL9020
Fieldbus	independent
Number of KL9020 per Fieldbus Coupler	1
Connection for K-bus extension	RJ-45 socket
Current consumption from the K-bus	typically 70 mA
Configuration	automatic
Dielectric strength	500 V (shielding, base plate / K-bus)
Weight	approx. 45 g
Dimensions (W x H x D)	approx. 26 mm x 100 mm x 70 mm
Mounting [▶ 27]	on 35 mm mounting rail according to EN 60715
Permissible ambient temperature range during operation	0°C ... + 55°C
Permissible ambient temperature range during storage	-25°C ... + 85°C
Permissible relative humidity	95 %, no condensation
Vibration / shock resistance	conforms to EN 60068-2-6 / EN 60068-2-27
EMC immunity / emission	conforms to EN 61000-6-2 / EN 61000-6-4
Protection class	IP20
Installation position	variable
Approvals	CE, cULus, ATEX, GL

2.8 KL9309

2.8.1 KL9309 - Introduction

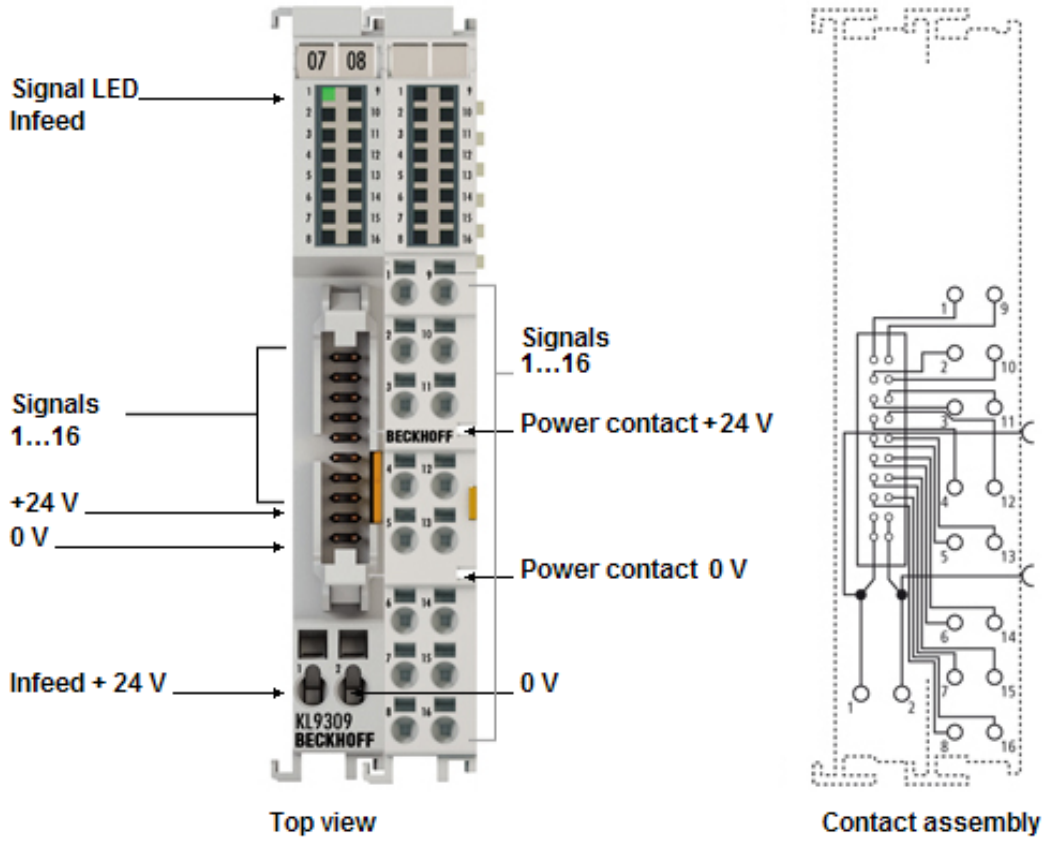


Fig. 8: KL9309

Adapter terminal for manual operating modules

The KL9309 adapter terminal has 16 terminal points, which provide the I/Os of the KL85xx manual operating modules. An additional supply voltage (24 V_{DC}) is provided via two terminal points (24 V, 0 V). The terminal can be connected to the manual operating module via a system cable using a 20-pole pin contact strip with locking.

The KL9309 has a modular design and can be incorporated seamlessly on the DIN rail. K-bus communication is passed on by the terminal, but the terminal itself is not visible to the K-bus and is therefore also not recognized by the bus coupler.

Two power contacts (24 V, 0 V) redirect the supply voltage to the downstream terminals, and four contacts (2 x 24 V, 2 x 0 V) of the 20-pin connector make it available as power supply for the manual operating modules.

2.8.2 Technical data

Technical data	KL9309
Power supply	24 V_{DC}
Power contacts	max. 24 V_{DC} / max. 10 A
Rated voltage Up	24 V_{DC}, total current max. 2 A
Bus interface	K-bus passive (no process image in the K-bus), counts as two K-bus terminals
I/O connection	Ribbon cable connection, 20-pin, 16-way HD connection
Diagnostic LEDs	green for Up
Special features	passive Bus Terminal for connection to the KL85xx manual operating modules
Electrical isolation	500 V (K-bus)
Output process image	-
Input process image	-
Weight	approx. 85 g
Dimensions (W x H x D) external dimensions	double K-bus terminal 24 mm x 100 mm x 68 mm (without plug connector)
Mounting [▶ 27]	on 35 mm mounting rail according to EN 60715
Permissible ambient temperature range during operation	0°C ... + 55°C
Permissible ambient temperature range during storage	-25°C ... + 85°C
Permissible relative humidity	95 %, no condensation
Vibration / shock resistance	conforms to EN 60068-2-6 / EN 60068-2-27
EMC immunity / emission	conforms to EN 61000-6-2 / EN 61000-6-4
Protection class	IP20
Installation position	variable
Approval	CE

2.9 Diagnostic LEDs

Front side

Fig. 9: LEDs

Meaning of the LED displays

LED	Color	State and significance
		on
Us/K-bus	green	K-bus communication OK
	red	Us present, no K-bus communication
	off	Us not present, no voltage applied
Up	green	Up connected
	off	Up not present, no voltage applied (check the voltage on the KL9309 or the connection of the 20-pin plug)

Rear side

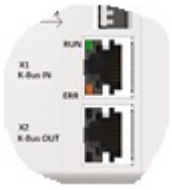


Fig. 10: LEDs

Meaning of the LED displays

LED	Color	State and significance	
		on	off
RUN	green	Lit, either weakly or strongly: K-bus communication OK	No K-Bus communication
ERR	red	No K-Bus communication	No error

3 Mounting and wiring

3.1 Dimensions

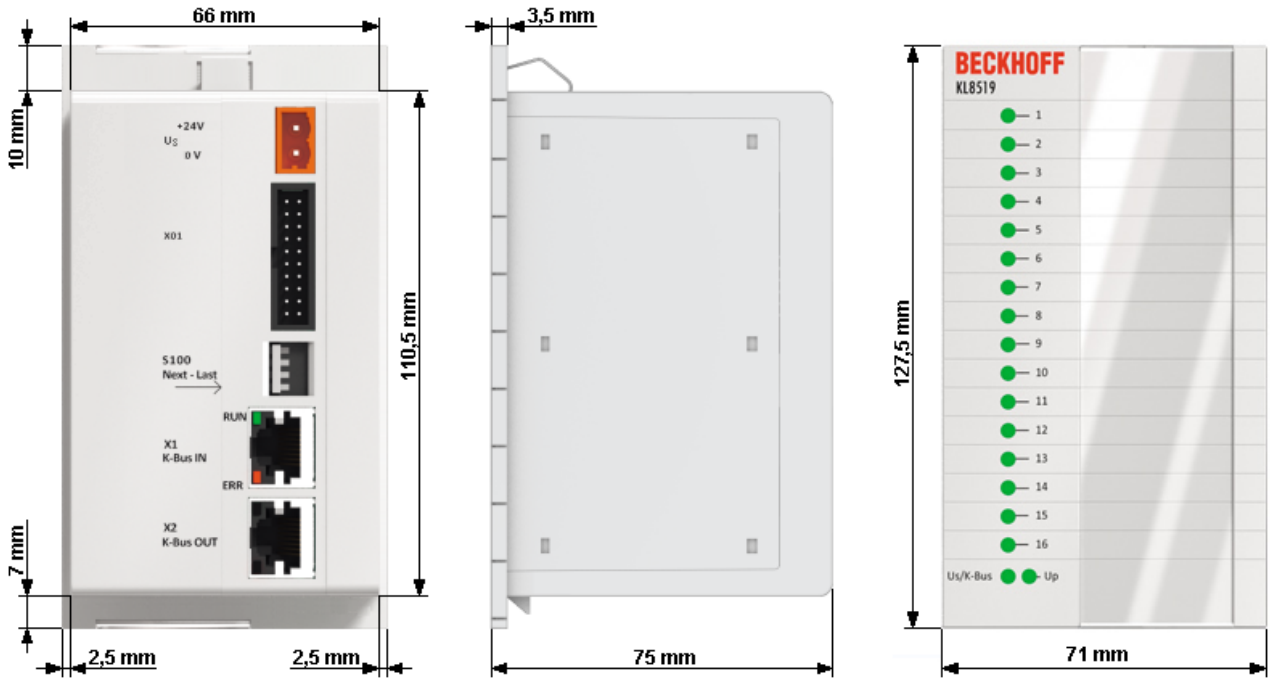


Fig. 11: Dimensions

Note	**Mounting cut-out and connector** The mounting cut-out [▶ 26] should have a size of 67 mm x 116.2 mm. Make sure that there is sufficient space behind the module for the connectors.

3.2 Mounting cut-out

The mounting cut-out should have a size of **67 mm x 116.2 mm**.

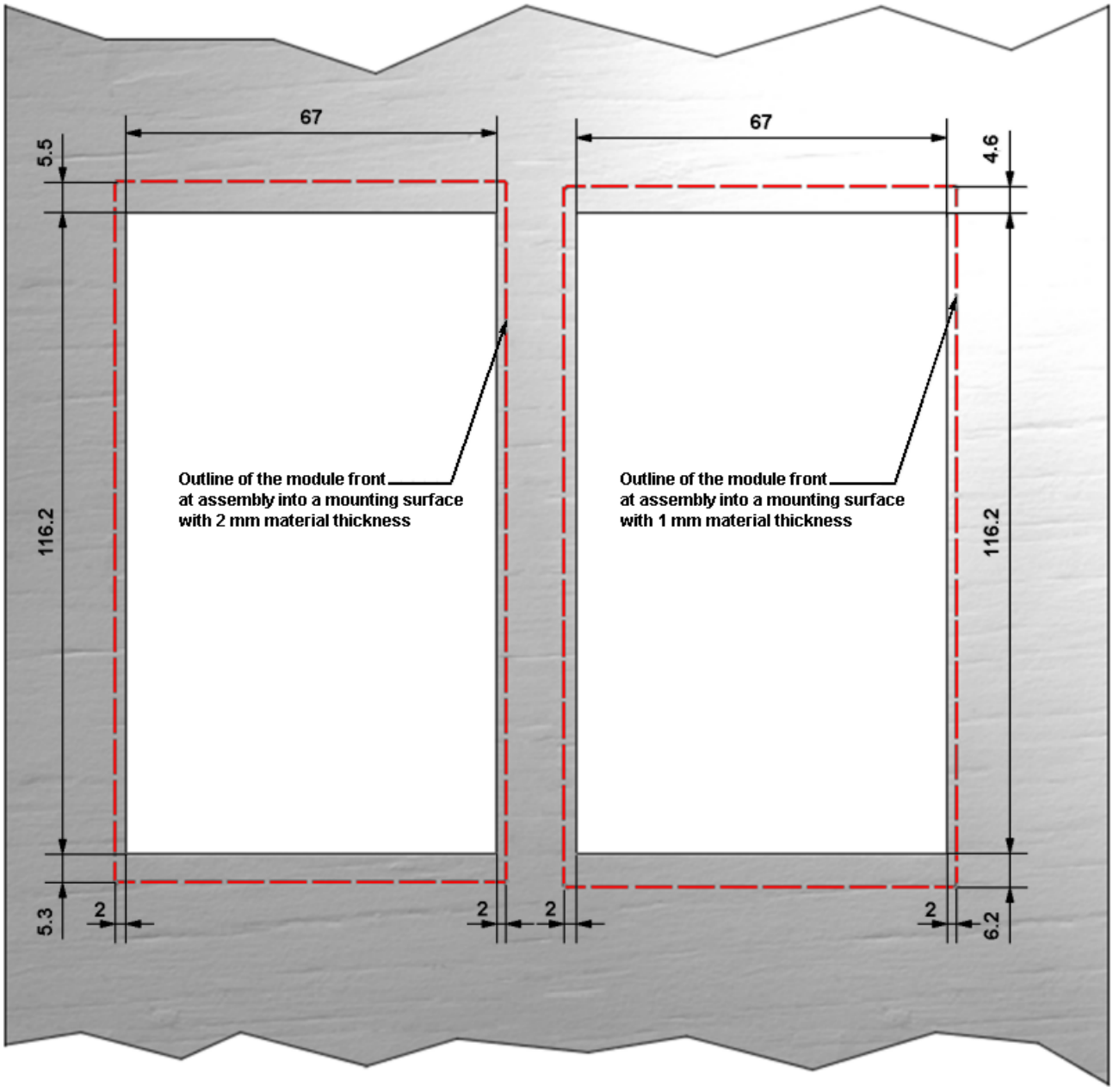


Fig. 12: Mounting cut-out

Note	**Positioning depends on the wall thickness of the mounting surface** The figure shows that different mounting heights arise depending on the wall thickness when installing in the mounting surface.

3.3 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

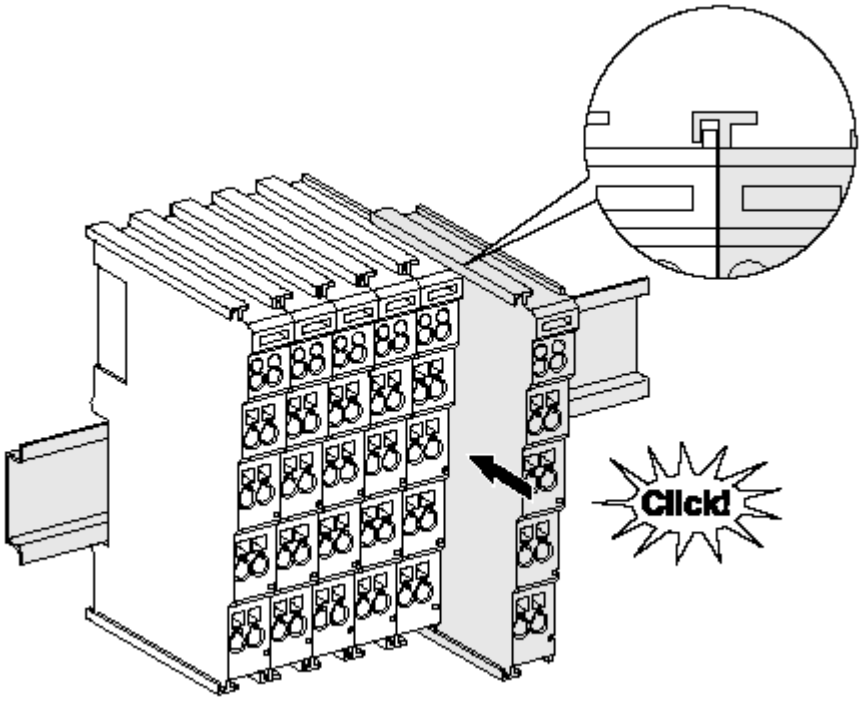


Fig. 13: Attaching on mounting rail

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

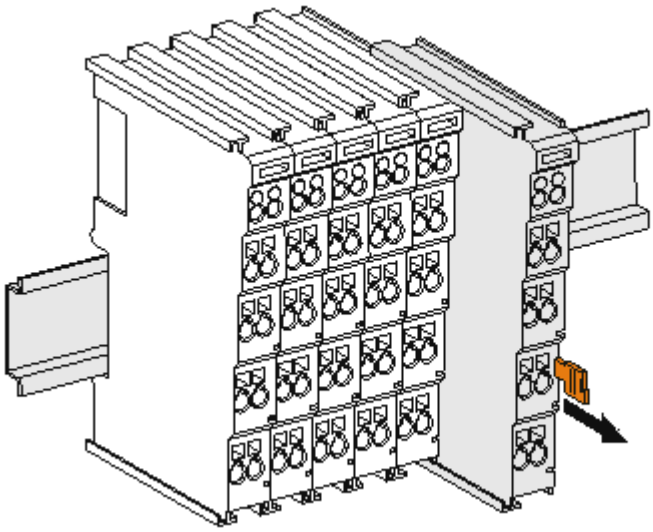


Fig. 14: Disassembling of terminal

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.

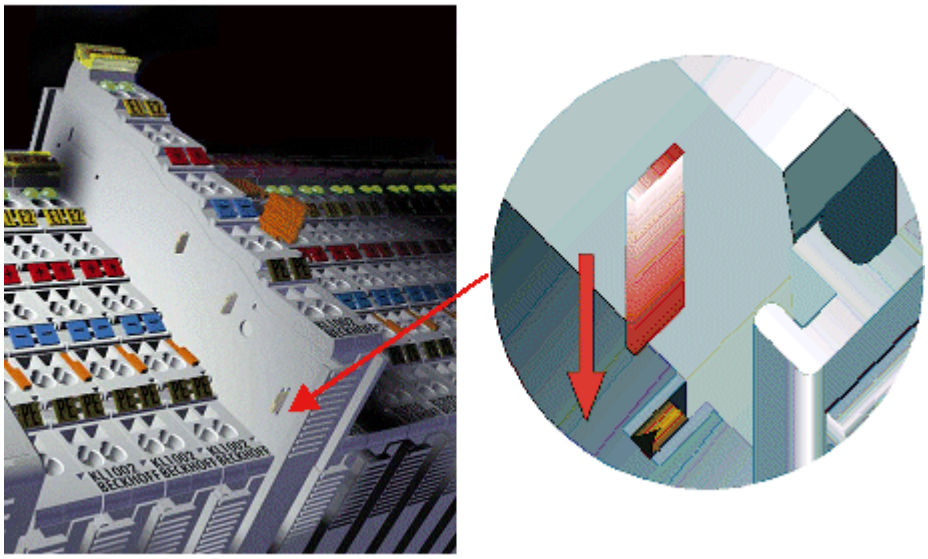


Fig. 15: *Power contact on left side*

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!

3.4 Connecting the manual operating modules

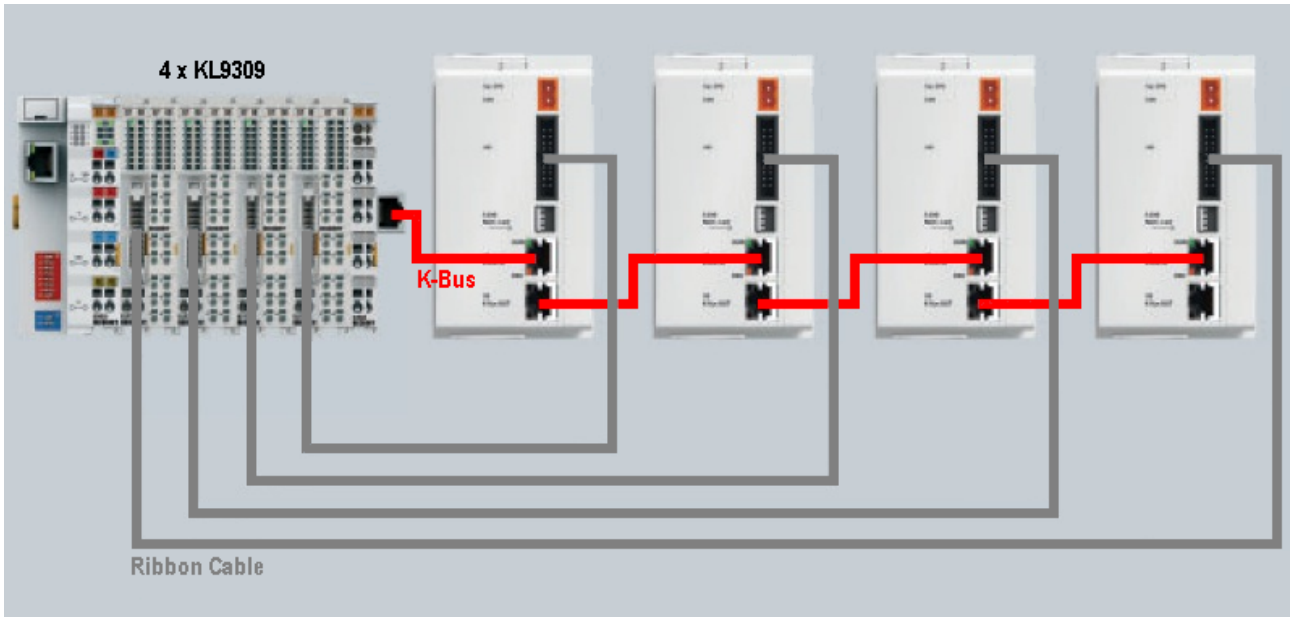


Fig. 16: Connecting the manual operating modules with K-bus extension and ribbon cable

Mounting

When mounting, observe the information in the chapter entitled Mounting rail installation [▶ 27].

1. Ensure that the system is powered down and in a safe state.
2. Install the first bus terminal block, consisting of the fieldbus coupler and the desired Bus Terminals, on a mounting rail.
   Instead of a standard end terminal (KL9010), install an end terminal with RJ45 socket (KL9020) as the last terminal at the end of the first Bus Terminal block.
3. Mount the first manual operating module.
4. Plug the RJ45 plug of an Ethernet cable into the RJ45 socket of the KL85xx until it audibly engages.
   Plug the other RJ45 plug of the Ethernet cable into the RJ45 socket of the KL85xx labelled *IN* until it audibly engages.
5. Mount the next manual operating module.
6. Plug the RJ45 plug of the Ethernet cable into the RJ45 socket of the KL85xx labelled *OUT* of the previous manual operating module until it audibly engages.
   Plug the other RJ45 plug of the Ethernet cable into the RJ45 socket of the KL85xx labelled *IN* of the added manual operating module until it audibly engages.
7. Repeat steps 5 and 6 in order to connect further expansion terminal blocks. A maximum of 31 expansion terminal blocks can be connected.
8. Set the *Function Switch* on all Coupler Terminals (KL85xx/KL9050) correctly.

Function Switch

Activate the terminating resistor at the last expansion terminal block of your K-bus extension system by switching the *Function Switch* on the last Coupler Terminal (KL85xx/KL9050) to the Last position.

WARNING	**Correct setting of the *Function Switch*** Correct setting of the *Function Switches* of all Coupler Terminals (KL9050) within a K-bus extension system must be ensured: The *Function Switch* of all Coupler Terminals (KL85xx/KL9050) to which a continuing Ethernet cable is connected must be set to position *Next*! (see image below) The *Function Switch* may be set to the Last position only at the last Coupler Terminal (KL85xx/KL9050) of the K-bus extension system! (see image below) All expansion terminal blocks connected after a Coupler Terminal (KL85xx/KL9050), whose Function Switch is set to position Last, are not included correctly in the process image: - The inputs of these terminals are not visible in the process image! - The outputs of these terminals are not controlled by the process image! Correct setting of the *Function Switches* must also be ensured if Coupler Terminals (KL85xx/KL9050) are replaced!

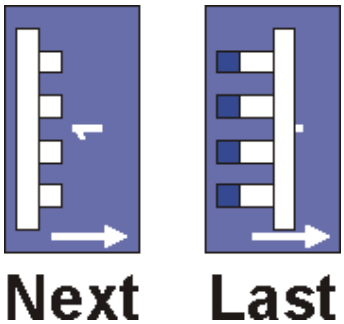


Fig. 17: Function Switch

Disassembly

When removing, observe the information in the chapter entitled Mounting rail installation [▶ 27].

1. Ensure that the system is powered down and in a safe state.
2. Press the plastic lock of the RJ45 plug and pull it from the socket.

3. Carefully pull the orange-colored strap approximately 1 cm out of the terminal to be disassembled, until it protrudes loosely. The lock with the mounting rail is now released for this terminal, and the terminal can be pulled from the mounting rail without excessive force.
4. Grasp the released terminal with thumb and index finger simultaneous at the upper and lower grooved housing surfaces and pull the terminal away from the mounting rail.

3.5 Power supply

WARNING	**Risk of injury through electric shock and damage to the device!** Bring the Bus Terminal system into a safe, voltage-free state before starting mounting, disassembly or wiring of the Bus Terminals!
Attention	**Link the ground connections of the power supply units of the Bus Coupler and all manual operating modules with a low-resistance connection** For interference-proof operation of the K-bus extension, the ground of the K-bus power supply of the higher-level fieldbus coupler must be connected with low-resistance to the grounds of the power supplies of all manual operating modules (see figure)!

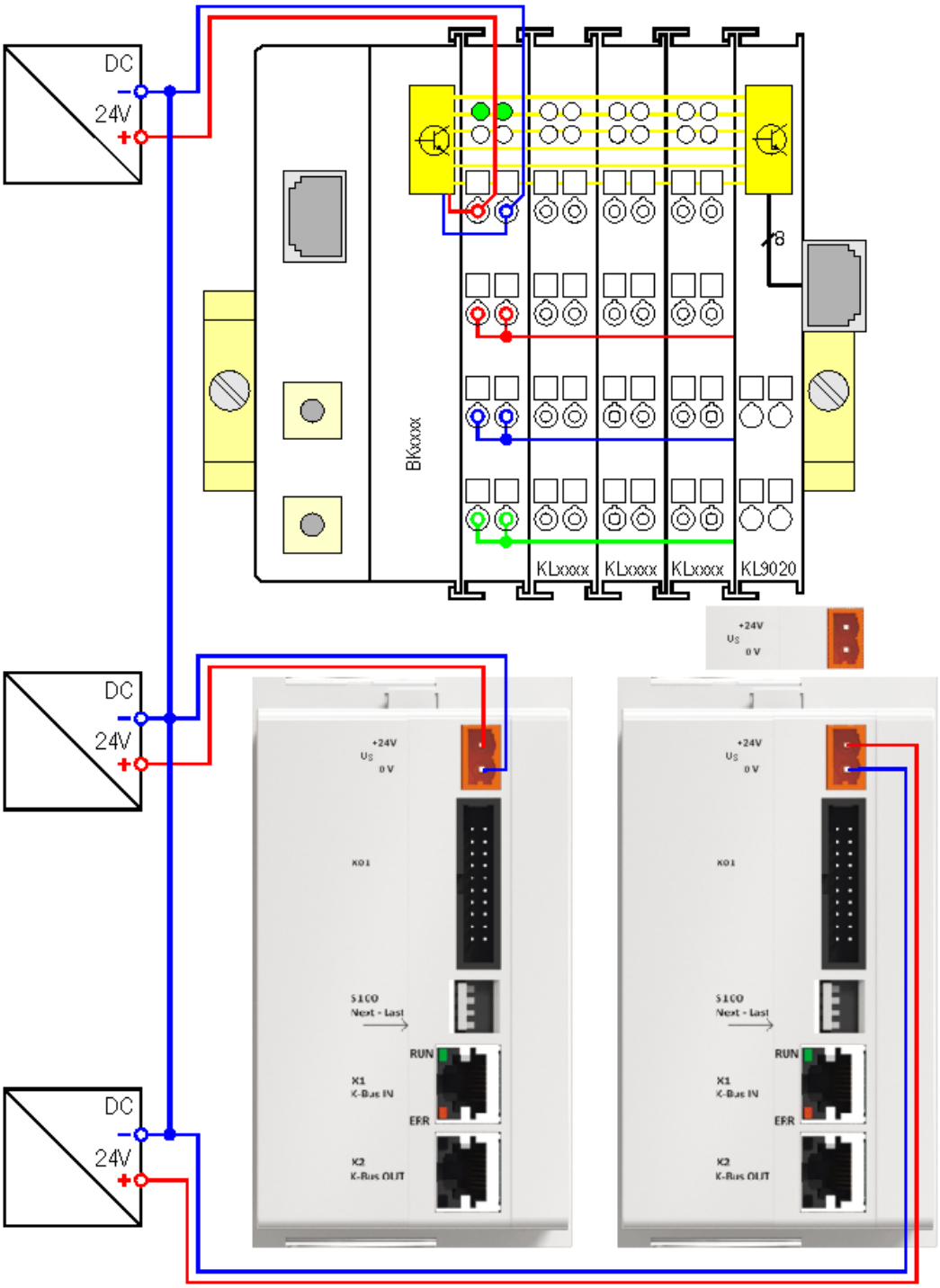


Fig. 18: Power supply with low-resistance ground connection

3.6 Connection diagrams of the 20-pin connectors

KL8519

Connection diagram for the 20-pin connector on the rear side of the KL8519.

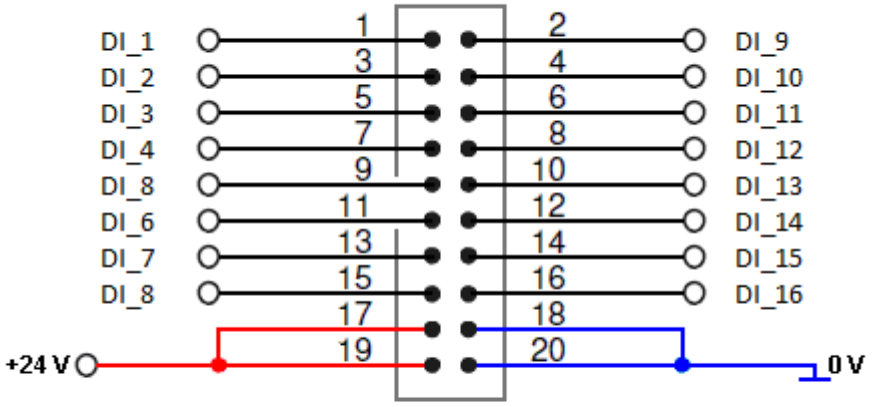


Fig. 19: KL8519 - Connecting the 20-pin connector

KL8524

Connection diagram for the 20-pin connector on the rear side of the KL8524.

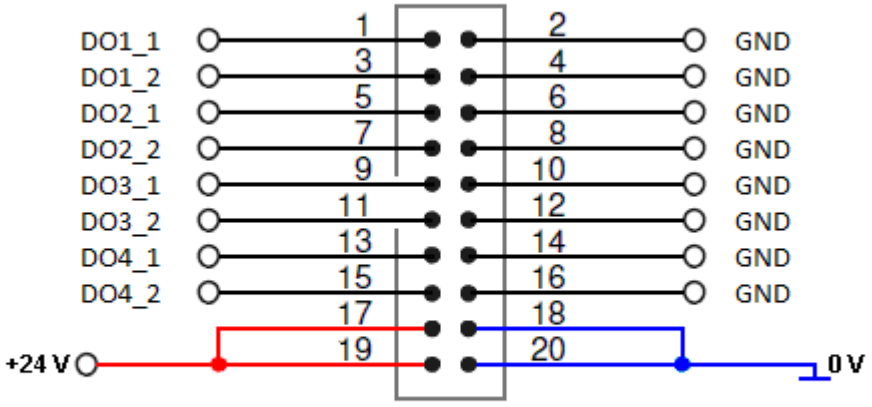


Fig. 20: KL8524 - Connecting the 20-pin connector

KL8528

Connection diagram for the 20-pin connector on the rear side of the KL8528.

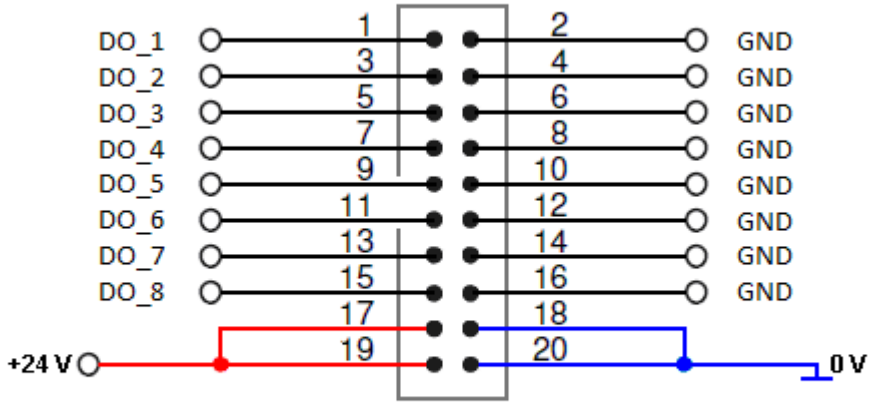


Fig. 21: KL8528 - Connecting the 20-pin connector

KL8548

Connection diagram for the 20-pin connector on the rear side of the KL8548.

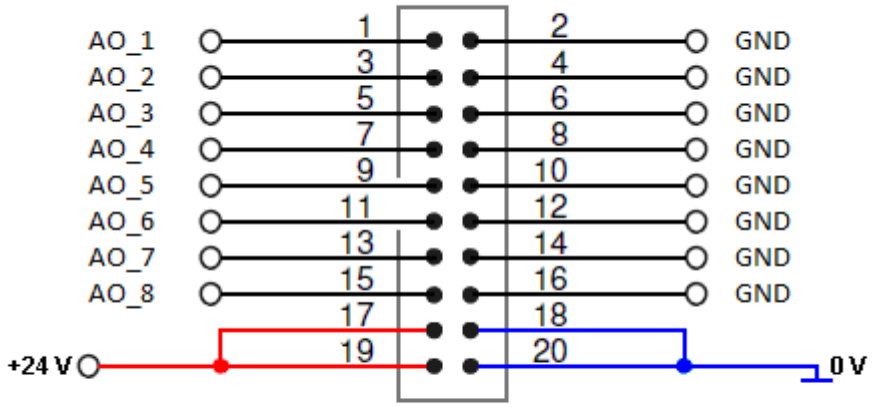


Fig. 22: KL8548 - Connecting the 20-pin connector

KL9309

Connection diagram of the 20-pin connector at the front of the KL9309.

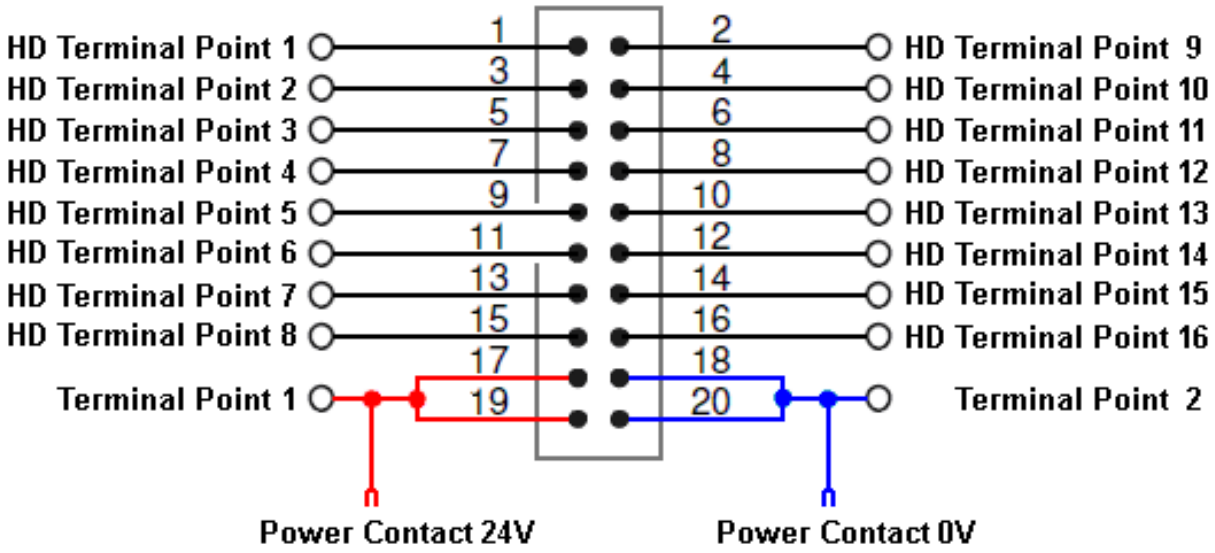


Fig. 23: KL9309 - Connecting the 20-pin connector

3.7 Labelling

This documentation contains a Word file with templates for the labelling of the control cabinet modules:

https://infosys.beckhoff.com/content/1033/kl85xx/Resources/zip/3913381387.zip

Make sure when printing that your printer does not scale the page size, so that the label strips retain their original size.

3.8 Ordering information

K-bus extension

Order designation	Description
KL9020	End Terminal with RJ45 socket [▶ 20] for K-bus extension
ZK1090-0101-1005	K-bus extension cable with two plugs attached, double shielded, grey, 50 cm
ZK1090-0101-1010	K-bus extension cable with two plugs attached, double shielded, grey, 100 cm
ZK1090-0101-1020	K-bus extension cable with two plugs attached, double shielded, grey, 200 cm
ZK1090-0101-1030	K-bus extension cable with two plugs attached, double shielded, grey, 300 cm
ZK1090-0101-1050	K-bus extension cable with two plugs attached, double shielded, grey, 500 cm

Sensor/actuator cable from KL85xx to KL9303

Order designation	Description
ZK8500-8282-7030	PVC cable, 20 x 0.14 mm², plug connector at both ends for KL85xx, KL9309 and terminals with ribbon cable connection, length 3 m, shielded
ZK8500-8282-7040	PVC cable, 20 x 0.14 mm², plug connector at both ends for KL85xx, KL9309 and terminals with ribbon cable connection, length 4 m, shielded
ZK8500-8282-7050	PVC cable, 20 x 0.14 mm², plug connector at both ends for KL85xx, KL9309 and terminals with ribbon cable connection, length 5 m, shielded

4 Programming in TwinCAT

4.1 Required libraries

4.1.1 TwinCAT 2

Note	**Installation** From TwinCAT 2.11 build 2251 (R3 and x64 engineering) the "TcKL85xx.lib/.lb6/.lbx" libraries are installed by default.

Further libraries required

For PC systems (x86) and Embedded PCs (CXxxxx):

- Standard.lib

For Bus Terminal Controllers from the BCxx00 series:

- Standard.lb6

For Bus Terminal controllers from the BCxx50, BCxx20 and BC9191 series:

- Standard.lbx

For Bus Terminal Controllers from the BXxx00 series:

- Standard.lbx

Note	**Memory usage** Some of the PLC program memory is already used up by integrating the library. Depending on the application program, therefore, the remaining memory may not be sufficient.

Download for CX9000 (https://infosys.beckhoff.com/content/1033/kl85xx/Resources/zip/3913383563.zip)

Required libraries:

- Standard.lib
- TcKL85xx.lib

Download for BX9000 (serial) (https://infosys.beckhoff.com/content/1033/kl85xx/Resources/zip/3913385739.zip)

Required libraries:

- Standard.lbx
- TcKL85xx.lbx

Download for BC9000 (serial) (https://infosys.beckhoff.com/content/1033/kl85xx/Resources/zip/3913387915.zip)

Required libraries:

- Standard.lb6
- TcKL85xx.lb6

Download for BC9050 (serial) (https://infosys.beckhoff.com/content/1033/kl85xx/Resources/zip/3913390091.zip)

Required libraries:

- Standard.lbx
- TcKL85xx.lbx

4.1.2 TwinCAT 3

Note	**Installation** From TwinCAT 3.1.4020.14 the "Tc2_KL85xx" libraries are installed by default.

4.2 Function blocks

Overview

Function block	Description
FB_KL8519 [▶ 39]	Function block for the configuration of the KL8519, for reading out the digital signals and for setting the LEDs.
FB_KL8524 [▶ 41]	Function block for the configuration of the KL8524, for reading out the digital signals and for setting the outputs/LEDs.
FB_KL8524Ex [▶ 43]	Function block for the configuration of the KL8524, for reading out the digital signals and for setting the outputs/LEDs.
FB_KL8528 [▶ 45]	Function block for configuration of the KL8528, for reading out the digital signals and for setting the outputs/LEDs.
FB_KL8528Ex [▶ 47]	Function block for configuration of the KL8528, for reading out the digital signals and for setting the outputs/LEDs.
FB_KL8548 [▶ 49]	Function block for configuration of the KL8548, for reading out the digital signals and for setting the outputs/LEDs.
FB_KL8548Ex [▶ 50]	Function block for configuration of the KL8548, for reading out the digital signals and for setting the outputs/LEDs.
FB_KL85xx16BitToWord [▶ 52]	Conversion from 16 bits to 1 Word
FB_KL85xx8BitToByte [▶ 53]	Conversion from 8 bits to 1 byte
FB_KL85xxByteTo8Bit [▶ 54]	Conversion from 1 byte to 8 bits.
FB_KL85xxWordTo16Bit [▶ 54]	Conversion from 1 word to 16 bits.

4.2.1 KL8519

Function blocks	Description
FB_KL8519 [▶ 39]	Function block for the configuration of the KL8519, for reading out the digital signals and for setting the LEDs.

4.2.1.1 FB_KL8519

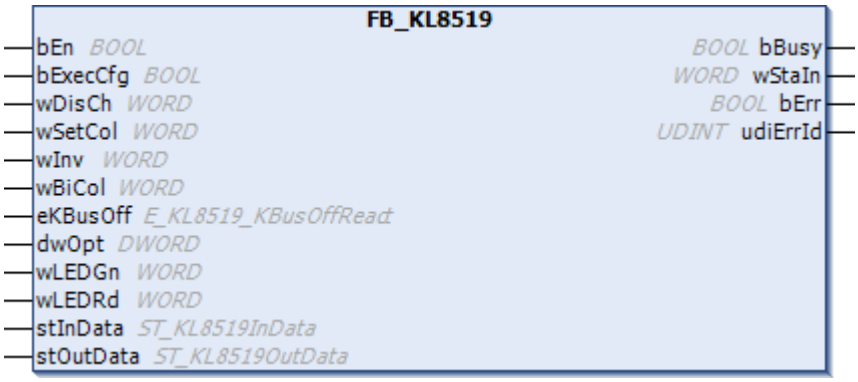


Fig. 24: Function block FB_KL8519

Application

This function block is used to configure the KL8519 [▶ 12], to read out the digital signals and to set the LEDs.

The function block must be called once per cycle.

The parameters are transferred to the terminal with a positive edge on *bExecCfg*.

The variable *wDisCh* switches the standard function [▶ 59] of the LED off. The LEDs can now be switched independently of the digital input signal via the PLC with the variables *wLEDGn* or *wLEDRd*. Each LED can be changed individually here.

The variable *wSetCol* defines how the LED should light up when the input is occupied. If it is "0" the LED will be green, if "1" the LED will be red. If the input is not occupied the LED remains off.

The variable *wInv* inverts the behavior of the LED. If the input is "FALSE" the LED turns on, if "TRUE" the LED turns off.

The variable *wBiCol* can be used to switch between green and red. If the variable is "1" the LED is red if the input is "FALSE" and green if it is TRUE. This behavior can be inverted with the variable *wInv*.

The variable *eKBusOff* defines how the LEDs should behave in the event of a K-bus error (or PLC stopped), if these are controlled by the PLC. If the variable is "0" (eKL8519_WatchdogOff) the LEDs remain off, if it is "1" (eKL8519_Watchdog500ms) the last state of the LED is toggled every 500 ms and with "2" (Watchdog1000ms) it is toggled every second.

VAR_INPUT

```
bEn         : BOOL;
bExecCfg    : BOOL;
wDisCh      : WORD;
wSetCol     : WORD;
wInv        : WORD;
wBiCol      : WORD;
eKBusOff    : E_KL8519_KBusOffReact [▶ 56];
dwOpt       : DWORD;
wLEDGn      : WORD;
wLEDRd      : WORD;
```

bEn: The function block is enabled by setting *bEn* to TRUE in the PLC program. With *bEn* = FALSE the execution of the function block will be stopped and all outputs reset.

bExecCfg: The configurations *wDisCh*, *wSetCol*, *wInv*, *wBiCol* and *eKBusOff* are written to the terminals when a positive edge is detected. *bBusy* goes TRUE. Note that the inputs are not updated during this time.

wDisCh: Configuration: Disablement of the standard function [▶ 59]of the LEDs. The LEDs can be set by the PLC. Bit 0 = LED 1, ... , bit 15 = LED 16. 65535_{dec} ($FFFF_{hex}$) deactivates the standard function [▶ 59] of all LEDs. These can then be set from the PLC via *wLEDGn* or *wLEDRd*.

wSetCol: Configuration: Using this variable you can define the color for the status LEDs of the individual channels (0=green or 1=red). With 65535_{dec} ($FFFF_{hex}$) all LEDs turn red if the input is occupied.

wInv: Configuration: Using this variable you can invert the display of the status LEDs of the individual channels. The process data (input signals) transmitted to the PLC are not affected by this.

wBiCol: Configuration: Using this variable you can switch the display of the status LEDs of the individual channels between two colors. Input unoccupied - LED is red, input occupied - LED is green, or vice versa if *wInv* is active.

eKBusOff: Configuration: In the event of a K-bus error all LEDs that are set via the PLC turn off, unless this contains a value other than 00_{bin}. In addition the standard function [▶ 59] of the LED must have been disabled with *wDisCh*.

dwOpt: For future options.

wLEDGn: Bit 0 = TRUE,..., bit 15 =TRUE sets the green LEDs, provided that the standard function [▶ 59] of the LED has been disabled with *wDisCh*. All green LEDs are set with 65535_{dec} ($FFFF_{hex}$). The FB_KL85xx16BitToWord [▶ 52] function block is available for preparing the signals for the graphic programming languages.

wLEDRd: Bit 0 = TRUE,..., bit 15 =TRUE sets the red LEDs, provided that the standard function [▶ 59] of the LED has been disabled with *wDisCh*. All red LEDs are set with 65535_{dec} ($FFFF_{hex}$). The FB_KL85xx16BitToWord [▶ 52] function block is available for preparing the signals for the graphic programming languages.

VAR_OUTPUT

```
bBusy      : BOOL;
wStaIn     : WORD;
bErr       : BOOL;
udiErrID   : UDINT;
```

bBusy: *bBusy* is TRUE as long as the configuration is being written (start with *bExecCfg*). Note that the inputs are not updated during this time.

wStaIn: Status of the digital input signals. Bit 0 = input 1,…, bit 15 = input 16. The FB_KL85xxWordTo16Bit [▶ 54] function block is available for preparing the signals for the graphic programming languages.

bErr: The *bErr*or output goes TRUE as soon as an error occurs. This error is described via the *udiErrID* variable.

udiErrID: In the event of an error the output issues an error code (see error codes [▶ 59]). *bError* goes TRUE at the same time.

VAR_IN_OUT

```
stInData       : ST_KL8519InData;
stOutData      : ST_KL8519OutData;
```

stInData: Process image of the inputs (see ST_KL8519InData [▶ 56]).

stOutData: Process image of the outputs (see ST_KL8519OutData [▶ 56]).

4.2.2 KL8524

Function blocks	Description
FB_KL8524 [▶ 41]	Function block for the configuration of the KL8524, for reading out the digital signals and for setting the outputs/LEDs.

4.2.2.1 FB_KL8524

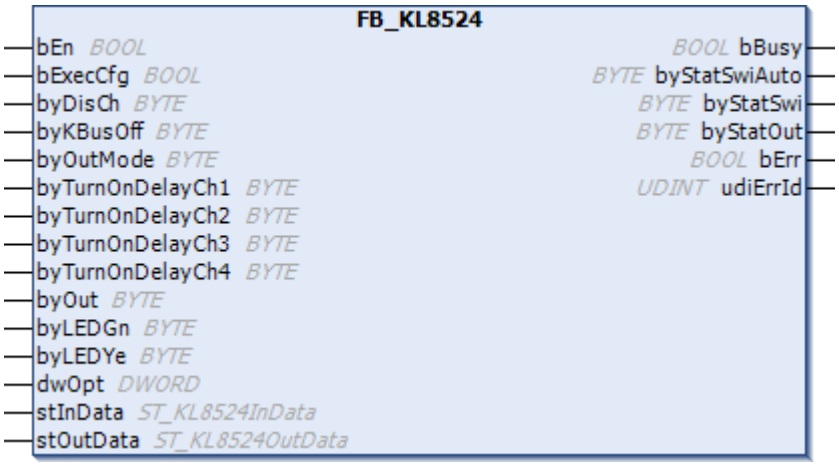


Fig. 25: Function block FB_KL8524

Application

This function block is used to configure the KL8524 [▶ 14], to read out the digital signals and to set the outputs/LEDs.

The function block must be called once per cycle.

The parameters are transferred to the terminal with a positive edge on *bExecCfg*.

The variable *byDisCh* switches the standard function of the LEDs off. The LEDs can now be switched independently of the digital input signal via the PLC with the variables *byLEDGn* or *byLEDYe*. Each LED can be changed individually here.

The respective output can be set to "TRUE" with *byKBusOff* in the event of a K-bus error (or PLC stopped) and switch position "auto". Bit 0..7 for the outputs 1..8. If the flag in *byKBusOff* is not set the output goes "FALSE" in the event of a K-bus error (or PLC stopped).

VAR_INPUT

```
bEn               : BOOL;
bExecCfg          : BOOL;
byDisCh           : BYTE := 0;
byKBusOff         : BYTE := 0;
byOutMode         : BYTE;
byTurnOnDelayCh1  : BYTE;
byTurnOnDelayCh2  : BYTE;
byTurnOnDelayCh3  : BYTE;
byTurnOnDelayCh4  : BYTE;
byOut             : BYTE;
byLEDGn           : BYTE;
byLEDYe           : BYTE;
dwOpt             : DWORD := 0;
```

bEn: The function block is enabled by setting *bEn* to TRUE in the PLC program. With *bEn* = FALSE the execution of the function block will be stopped and all outputs reset.

bExecCfg: The configurations *byDisCh*, *byKBusOff*, *byOutMode*, *byTurnOnDelayCh1*, *byTurnOnDelayCh2*, *byTurnOnDelayCh3* and *byTurnOnDelayCh4* are written to the terminals. *bBusy* goes TRUE. Note that the inputs/outputs are not updated during this time.

byDisCh: Configuration: Disablement of the standard function [▶ 59]of the LEDs. The LEDs can be set by the PLC. Bit 0 = LED 1, ... , bit 7 = LED 8. 255_{dec} (FF_{hex}) deactivates the standard function [▶ 59] of all LEDs. These can then be set from the PLC via *byLEDGn* or *byLEDYe*.

byKBusOff: Configuration: Reaction of the outputs in the event of a K-bus error. The respective output is set with bit 0 = output 1,…, bit 7 =output 8 in the event of a K-bus error (or PLC stopped) provided that the switch is set to "auto". Note: 2 outputs are assigned to a channel and only one output may be activated per channel. If the value 255_{dec} (FF_{hex}) is transferred, for example, no output is set when the PLC stops.

byOutMode: Configuration: The outputs assigned to a channel can be set in 2 variants. Bit 0 = FALSE = channel 1 output mode 1, bit 0 = TRUE = channel 1 output mode 2,…, bit 3 = FALSE = channel 4 output mode 1, bit 3 = TRUE = channel 4 output mode 2.
Output mode 1: Output 1 is switched if the three-step switch is in position 1; output 2 is switched if the three-step switch is in position 2
Output mode 2: Output 1 is switched if the three-step switch is in position 1 or position 2; output 2 is switched if the three-step switch is in position 2, following a delay which can be set with *byTurnOnDelayChx*.

byTurnOnDelayCh1: Configuration: Switching time for channel 1 output mode 2. 1 bit = 10 ms -> with default 255_{dec}, FF_{hex} the second output is switched on after 2550 ms if the three-step switch is in position 2.

byTurnOnDelayCh2: Configuration: Switching time for channel 2 output mode 2. 1 bit = 10 ms -> with default 255_{dec}, FF_{hex} the second output is switched on after 2550 ms if the three-step switch is in position 2.

byTurnOnDelayCh3: Configuration: Switching time for channel 3 output mode 2. 1 bit = 10 ms -> with default 255_{dec}, FF_{hex} the second output is switched on after 2550 ms if the three-step switch is in position 2.

byTurnOnDelayCh4: Configuration: Switching time for channel 4 output mode 2. 1 bit = 10 ms -> with default 255_{dec}, FF_{hex} the second output is switched on after 2550 ms if the three-step switch is in position 2.

byOut: Bit 0 = TRUE, ... , Bit 7 = TRUE sets the respective output, provided the switch is set to "auto". All outputs are set with 255_{dec} (FF_{hex}). The FB_KL85xx8BitToByte [▶ 53] function block is available for preparing the signals for the graphic programming languages.

byLEDGn: Bit 0 = TRUE,..., bit 7 = TRUE sets the green LEDs, provided that the standard function [▶ 59] of the LEDs has been disabled with *byDisCh*. All green LEDs are set with 255_{dec} (FF_{hex}). The FB_KL85xx8BitToByte [▶ 53] function block is available for preparing the signals for the graphic programming languages.

byLEDYe: Bit 0 = TRUE,..., bit 7 = TRUE sets the yellow LEDs, provided that the standard function [▶ 59] of the LEDs has been disabled with *byDisCh*. All yellow LEDs are set with 255_{dec} (FF_{hex}). The FB_KL85xx8BitToByte [▶ 53] function block is available for preparing the signals for the graphic programming languages.

dwOpt: For future options.

VAR_OUTPUT

```
bBusy          : BOOL;
byStatSwiAuto  : BYTE;
byStatSwi      : BYTE;
byStatOut      : BYTE;
bErr           : BOOL;
udiErrID       : UDINT;
```

bBusy: *bBusy* is TRUE as long as the configuration is being written (start with *bExecCfg*). Note that the inputs are not updated during this time.

byStatSwiAuto: Status of the "auto" switch. Bit 0 = channel 1, ... , bit 3 = channel 4. The FB_KL85xxByteTo8Bit [▶ 54] function block is available for preparing the signals for the graphic programming languages.

byStatSwi: Status of the three-step switch. Bit 0 = channel 1 position 1, bit 1 = channel 1 position 2 ... , bit 7 = channel 4 position 2. The FB_KL85xxByteTo8Bit [▶ 54] function block is available for preparing the signals for the graphic programming languages.

byStatOut: Status of the digital output signals. Bit 0 = channel 1 output 1, bit 1 = channel 1 output 2, ... , bit 7 = channel 4 output 2. The FB_KL85xxByteTo8Bit [▶ 54] function block is available for preparing the signals for the graphic programming languages.

bErr: The *bErr*or output goes TRUE as soon as an error occurs. This error is described via the *udiErrID* variable.

udiErrID: In the event of an error the output issues an error code (see error codes [▶ 59]). *bError* goes TRUE at the same time.

VAR_IN_OUT

```
stInData          : ST_KL8524InData;
stOutData         : ST_KL8524OutData;
```

stInData: Process image of the inputs (see ST_KL8524InData [▶ 57]).

stOutData: Process image of the outputs (see ST_KL8524OutData [▶ 57]).

4.2.2.2 FB_KL8524Ex

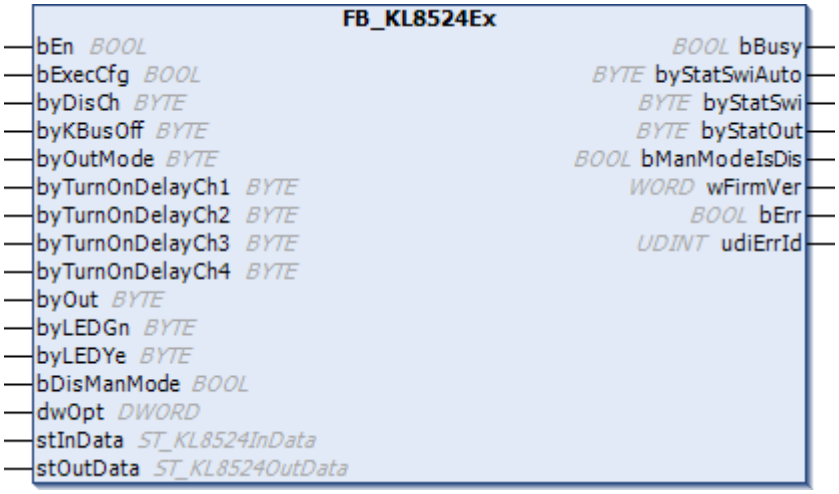


Fig. 26: Function block FB_KL8524Ex

Application

This function block is used to configure the KL8524, to read out the digital signals and to set the outputs/ LEDs.

The function block must be called once per cycle.

The parameters are transferred to the terminal with a positive edge on bExecCfg.

The variable byDisCh switches the standard function of the LEDs off. The LEDs can now be switched independently of the digital input signal via the PLC with the variables byLEDGn or byLEDYe. Each LED can be changed individually here.

The respective output can be set to "TRUE" with byKBusOff in the event of a K-bus error (or PLC stopped) and switch position "auto". Bit 0..7 for the outputs 1..8. If the flag in byKBusOff is not set the output goes "FALSE" in the event of a K-bus error (or PLC stopped).

VAR_INPUT

```
bEn               : BOOL;
bExecCfg          : BOOL;
byDisCh           : BYTE := 0;
byKBusOff         : BYTE := 0;
byOutMode         : BYTE;
byTurnOnDelayCh1  : BYTE;
byTurnOnDelayCh2  : BYTE;
byTurnOnDelayCh3  : BYTE;
```

```
byTurnOnDelayCh4    : BYTE;
byOut               : BYTE;
byLEDGn             : BYTE;
byLEDYe             : BYTE;
bDisManMode         : BOOL;
dwOpt               : DWORD := 0;
```

bEn: The function block is enabled by setting bEn to TRUE in the PLC program. With bEn = FALSE the execution of the function block will be stopped and all outputs reset.

bExecCfg: The configurations byDisCh, byKBusOff, byOutMode, byTurnOnDelayCh1, byTurnOnDelayCh2, byTurnOnDelayCh3 and byTurnOnDelayCh4 are written to the terminals. *bBusy* becomes TRUE. Note that the inputs/outputs are not updated during this time.

byDisCh: Configuration: Disablement of the standard function of the LEDs. The LEDs can be set by the PLC. Bit 0 = LED 1, ... , bit 7 = LED 8. 255_{dec} (FF_{hex}) deactivates the standard function of all LEDs. These can then be set from the PLC via *byLEDGn* or *byLEDYe*.

byKBusOff: Configuration: Reaction of the outputs in the event of a K-bus error. The respective output is set with bit 0 = output 1, ... , bit 7 =output 8 in the event of a K-bus error (or PLC stopped) provided that the switch is set to "auto". Note: 2 outputs are assigned to a channel and only one output may be activated per channel. If the value 255_{dec} (FF_{hex}) is transferred, for example, no output is set when the PLC stops.

byOutMode: Configuration: The outputs assigned to a channel can be set in 2 variants. Bit 0 = FALSE = channel 1 output mode 1, bit 0 = TRUE = channel 1 output mode 2, ... , bit 3 = FALSE = channel 4 output mode 1, bit 3 = TRUE = channel 4 output mode 2.
Output mode 1: Output 1 is switched if the three-step switch is in position 1; output 2 is switched if the three-step switch is in position 2.
Output mode 2: Output 1 is switched if the three-step switch is in position 1 or position 2; output 2 is switched if the three-step switch is in position 2, following a delay which can be set with *byTurnOnDelayChx.*

byTurnOnDelayCh1: Configuration: Switching time for channel 1 output mode 2. 1 bit = 10 ms -> with default 255_{dec}, FFhex the second output is switched on after 2550 ms if the three-step switch is in position 2.

byTurnOnDelayCh2: Configuration: Switching time for channel 2 output mode 2. 1 bit = 10 ms -> with default 255_{dec}, FF_{hex} the second output is switched on after 2550 ms if the three-step switch is in position 2.

byTurnOnDelayCh3: Configuration: Switching time for channel 3 output mode 2. 1 bit = 10 ms -> with default 255_{dec}, FF_{hex} the second output is switched on after 2550 ms if the three-step switch is in position 2.

byTurnOnDelayCh4: Configuration: Switching time for channel 4 output mode 2. 1 bit = 10 ms -> with default 255_{dec}, FF_{hex} the second output is switched on after 2550 ms if the three-step switch is in position 2.

byOut: Bit 0 = TRUE, ... , Bit 7 = TRUE sets the respective output, provided the switch is set to "auto". All outputs are set with 255_{dec} (FF_{hex}). The FB_KL85xx8BitToByte function block is available for preparing the signals for the graphic programming languages.

byLEDGn: Bit 0 = TRUE,..., bit 7 = TRUE sets the green LEDs, provided that the standard function of the LEDs has been disabled with byDisCh. All green LEDs are set with 255_{dec} (FF_{hex}). The FB_KL85xx8BitToByte function block is available for preparing the signals for the graphic programming languages.

byLEDYe: Bit 0 = TRUE,..., bit 7 = TRUE sets the yellow LEDs, provided that the standard function of the LEDs has been disabled with byDisCh. All yellow LEDs are set with 255_{dec} (FF_{hex}). The FB_KL85xx8BitToByte function block is available for preparing the signals for the graphic programming languages.

bDisManMode: Disabling manual operating mode.

dwOpt: For future options.

VAR_OUPUT

```
bBusy               : BOOL;
byStatSwiAuto       : BYTE;
byStatSwi           : BYTE;
byStatOut           : BYTE;
bManModelsDis       : BOOL;
wFirmVer            : WORD;
bErr                : BOOL;
udiErrID            : UDINT;
```

bBusy: bBusy is TRUE as long as the configuration is being written (start with *bExecCfg*). Note that the inputs are not updated during this time.

byStatSwiAuto: Status of the "auto" switch. Bit 0 = channel 1, ... , bit 3 = channel 4. The FB_KL85xxByteTo8Bit function block is available for preparing the signals for the graphic programming languages.

byStatSwi: Status of the three-step switch. Bit 0 = channel 1 position 1, bit 1 = channel 1 position 2 ... , bit 7 = channel 4 position 2. The FB_KL85xxByteTo8Bit function block is available for preparing the signals for the graphic programming languages.

byStatOut: Status of the digital output signals. Bit 0 = channel 1 output 1, bit 1 = channel 1 output 2, ... , bit 7 = channel 4 output 2. The FB_KL85xxByteTo8Bit function block is available for preparing the signals for the graphic programming languages.

bManModeIsDis: Manual operating mode disabled.

wFirmVer: Issues the firmware version.

bErr: The *bErr*or output goes TRUE as soon as an error occurs. This error is described via the *udiErrID* variable.

udiErrID: In the event of an error the output issues an error code (see error codes). *bError* goes TRUE at the same time.

VAR_IN_OUT

```
stInData        : ST_KL8524InData;
stOutData       : ST_KL8524OutData;
```

stInData: Process image of the inputs (see ST_KL8524InData).

stOutData: Process image of the outputs (see ST_KL8524OutData).

4.2.3 KL8528

Function blocks	Description
FB_KL8528 [▶ 45]	Function block for configuration of the KL8528, for reading out the digital signals and for setting the outputs/LEDs.

4.2.3.1 FB_KL8528

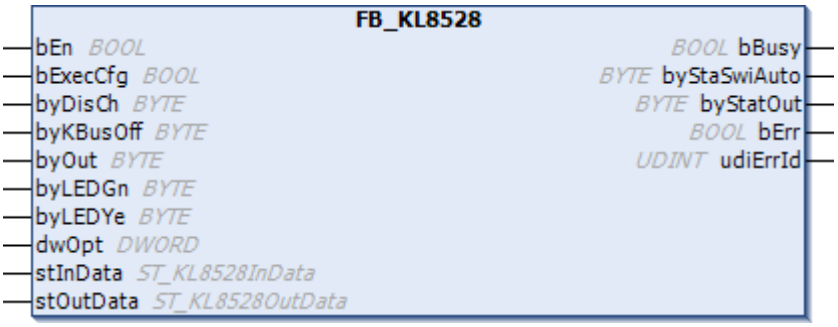


Fig. 27: Function block FB_KL8528

Application

This function block is used to configure the KL8528 [▶ 16], to read out the digital signals and to set the outputs/LEDs.

The function block must be called once per cycle.

The parameters are transferred to the terminal with a positive edge on *bExecCfg*.

The variable *byDisCh* switches the standard function [▶ 60] of the LEDs off. The LEDs can now be switched independently of the digital input signal via the PLC with the variables *byLEDGn* or *byLEDYe*. Each LED can be changed individually here.

The respective output can be set to "TRUE" with *byKBusOff* in the event of a K-bus error (or PLC stopped) and switch position "auto". Bit 0..7 for the outputs 1..8. If the flag in *byKBusOff* is not set the output goes "FALSE" in the event of a K-bus error (or PLC stopped).

VAR_INPUT

```
bEn           : BOOL;
bExecCfg      : BOOL;
byDisCh       : BYTE;
byKBusOff     : BYTE;
byOut         : BYTE;
byLEDGn       : BYTE;
byLEDYe       : BYTE;
dwOpt         : DWORD;
```

bEn: The function block is enabled by setting *bEn* to TRUE in the PLC program. With *bEn* = FALSE the execution of the function block will be stopped and all outputs reset.

bExecCfg: The configurations *byDisCh* and *byKBusOff* are written to the terminals on a positive edge. *bBusy* goes TRUE. Note that the inputs/outputs are not updated during this time.

byDisCh: Configuration: Disablement of the standard function [▶ 60] of the LEDs. The LEDs can be set by the PLC. Bit 0 = LED 1, ... , bit 7 = LED 8. 255_{dec} (FF_{hex}) deactivates the standard function [▶ 60] of all LEDs. These can then be set from the PLC via *byLEDGn* or *byLEDYe*.

byKBusOff: Configuration: Reaction of the outputs in the event of a K-bus error. The respective output is set with bit 0 = output 1, ... , bit 7 =output 8 in the event of a K-bus error (or PLC stopped) provided that the switch is set to "auto".

byOut: Bit 0 = TRUE, ... , Bit 7 = TRUE sets the respective output, provided the switch is set to "auto". All outputs are set with 255_{dec} (FF_{hex}). The FB_KL85xx8BitToByte [▶ 53] function block is available for preparing the signals for the graphic programming languages.

byLEDGn: Bit 0 = TRUE,..., bit 7 = TRUE sets the green LEDs, provided that the standard function [▶ 60] of the LEDs has been disabled with *byDisCh*. All green LEDs are set with 255_{dec} (FF_{hex}). The FB_KL85xx8BitToByte [▶ 53] function block is available for preparing the signals for the graphic programming languages.

byLEDYe: Bit 0 = TRUE,..., bit 7 = TRUE sets the yellow LEDs, provided that the standard function [▶ 60] of the LEDs has been disabled with *byDisCh*. All yellow LEDs are set with 255_{dec} (FF_{hex}). The FB_KL85xx8BitToByte [▶ 53] function block is available for preparing the signals for the graphic programming languages.

dwOpt: For future options.

VAR_OUTPUT

```
bBusy         : BOOL;
byStaSwiAuto  : BYTE;
byStatOut     : BYTE;
bErr          : BOOL;
udiErrID      : UDINT;
```

bBusy: *bBusy* is TRUE as long as the configuration is being written (start with *bExecCfg*). Note that the inputs are not updated during this time.

byStaSwiAuto: Status of the "auto" switch. Bit 0 = switch 1, ... , bit 7 = switch 8. The FB_KL85xxByteTo8Bit [▶ 54] function block is available for preparing the signals for the graphic programming languages.

byStatOut: Status of the digital output signals. Bit 0 = output 1, ... , bit 7 = output 8. The FB_KL85xxByteTo8Bit [▶ 54] function block is available for preparing the signals for the graphic programming languages.

bErr: The *bErr*or output goes TRUE as soon as an error occurs. This error is described via the *udiErrID* variable.

udiErrID: In the event of an error the output issues an error code (see error codes [▶ 59]). *bError* goes TRUE at the same time.

VAR_IN_OUT

```
stInData      : ST_KL8528InData;
stOutData     : ST_KL8528OutData;
```

ST_KL8528OutData [▶ 58] ST_KL8528InData [▶ 57]

stInData: Process image of the inputs (see ST_KL8528InData [▶ 57]).

stOutData: Process image of the outputs (see ST_KL8528OutData [▶ 58]).

4.2.3.2 FB_KL8528Ex

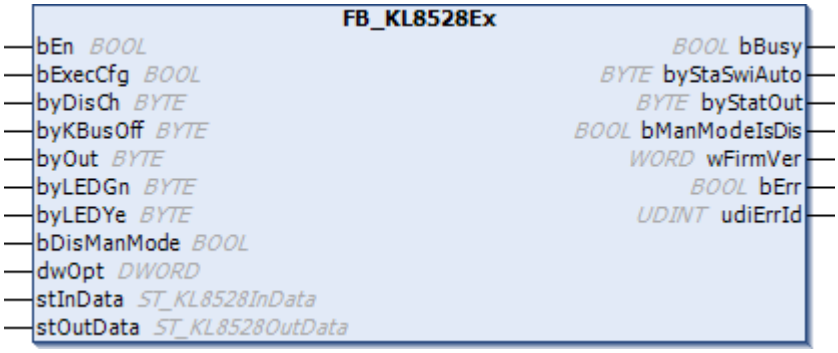


Fig. 28: Function block FB_KL8528Ex

Application

This function block is used to configure the KL8528, to read out the digital signals and to set the outputs/ LEDs.

The function block must be called once per cycle.

The parameters are transferred to the terminal with a positive edge on *bExecCfg*.

The variable *byDisCh* switches the standard function of the LEDs off. The LEDs can now be switched independently of the digital input signal via the PLC with the variables *byLEDGn* or *byLEDYe*. Each LED can be changed individually here.

The respective output can be set to "TRUE" with *byKBusOff* in the event of a K-bus error (or PLC stopped) and switch position "auto". Bit 0..7 for the outputs 1..8. If the flag in *byKBusOff* is not set the output goes "FALSE" in the event of a K-bus error (or PLC stopped).

VAR_INPUT

```
bEn           : BOOL;
bExecCfg      : BOOL;
byDisCh       : BYTE;
byKBusOff     : BYTE;
byOut         : BYTE;
byLEDGn       : BYTE;
byLEDYe       : BYTE;
bDisManMode   : BOOL;
dwOpt         : DWORD;
```

bEn: The function block is enabled by setting *bEn* to TRUE in the PLC program. With *bEn* = FALSE the execution of the function block will be stopped and all outputs reset.

bExecCfg: The configurations *byDisCh* and *byKBusOff* are written to the terminals on a positive edge. *bBusy* goes TRUE. Note that the inputs/outputs are not updated during this time.

byDisCh: Configuration: Disablement of the standard function of the LEDs. The LEDs can be set by the PLC. Bit 0 = LED 1, ... , bit 7 = LED 8. 255_{dec} (FF_{hex}) deactivates the standard function of all LEDs. These can then be set from the PLC via *byLEDGn* or *byLEDYe*.

byKBusOff: Configuration: Reaction of the outputs in the event of a K-bus error. The respective output is set with bit 0 = output 1, ... , bit 7 =output 8 in the event of a K-bus error (or PLC stopped) provided that the switch is set to "auto".

byOut: Bit 0 = TRUE, ... , Bit 7 = TRUE sets the respective output, provided the switch is set to "auto". All outputs are set with 255_{dec} (FF_{hex}). The FB_KL85xx8BitToByte function block is available for preparing the signals for the graphic programming languages.

byLEDGn: Bit 0 = TRUE,..., bit 7 = TRUE sets the green LEDs, provided that the standard function of the LEDs has been disabled with *byDisCh*. All green LEDs are set with 255_{dec} (FF_{hex}). The FB_KL85xx8BitToByte function block is available for preparing the signals for the graphic programming languages.

byLEDYe: Bit 0 = TRUE,..., bit 7 = TRUE sets the yellow LEDs, provided that the standard function of the LEDs has been disabled with *byDisCh*. All yellow LEDs are set with 255_{dec} (FF_{hex}). The FB_KL85xx8BitToByte function block is available for preparing the signals for the graphic programming languages.

bDisManMode: Disabling manual operating mode.

dwOpt: For future options.

VAR_OUPUT

```
bBusy           : BOOL;
byStaSwiAuto    : BYTE;
byStatOut       : BYTE;
bManModelsDis   : BOOL;
wFirmVer        : WORD;
bErr            : BOOL;
udiErrID        : UDINT;
```

bBusy: *bBusy* is TRUE as long as the configuration is being written (start with *bExecCfg*). Note that the inputs are not updated during this time.

byStaSwiAuto: Status of the "auto" switch. Bit 0 = switch 1, ... , bit 7 = switch 8. The FB_KL85xxByteTo8Bit function block is available for preparing the signals for the graphic programming languages.

byStatOut: Status of the digital output signals. Bit 0 = output 1, ... , bit 7 = output 8. The FB_KL85xxByteTo8Bit function block is available for preparing the signals for the graphic programming languages.

bManModelsDis: Manual operating mode disabled.

wFirmVer: Issues the firmware version.

bErr: The *bErr*or output goes TRUE as soon as an error occurs. This error is described via the *udiErrID* variable.

udiErrID: In the event of an error the output issues an error code (see error codes). *bError* goes TRUE at the same time.

VAR_IN_OUT

```
stInData      : ST_KL8528InData;
stOutData     : ST_KL8528OutData;
```

stInData: Process image of the inputs (see ST_KL8528InData).

stOutData: Process image of the outputs (see ST_KL8528OutData).

4.2.4 KL8548

Function blocks	Description
FB_KL8548 [▶ 49]	Function block for configuration of the KL8548, for reading out the digital signals and for setting the outputs/LEDs.

4.2.4.1 FB_KL8548

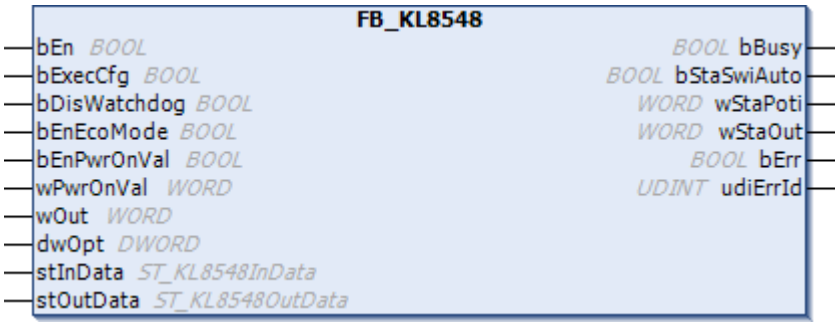


Fig. 29: Function block FB_KL8548

Application

This function block is used to configure the KL8548 [▶ 18], to read out the digital signals and to set the outputs/LEDs.

The function block must be called once per cycle.

The parameters are transferred to the terminal with a positive edge on *bExecCfg*.

The standard function [▶ 60] of the LEDs cannot be deactivated.

VAR_INPUT

```
bEn            : BOOL;
bExecCfg       : BOOL;
bDisWatchdog   : BOOL;
bEnEcoMode     : BOOL;
bEnPwrOnVal    : BOOL;
wPwrOnVal      : WORD;
wOut           : WORD;
dwOpt          : DWORD;
```

bEn: The function block is enabled by setting *bEn* to TRUE in the PLC program. With *bEn* = FALSE the execution of the function block will be stopped and all outputs reset.

bExecCfg: The configurations *bDisWatchdog*, *bEnEcoMode*, *bEnPwrOnVal* and *wPwrOnVal* are written to the terminals when a positive edge is encountered. *bBusy* goes TRUE. Note that the inputs are not updated during this time.

bDisWatchdog: Configuration: If False the last value of *wStaOut* remains unchanged in the event of a K-bus error (the value is set to 0 if the PLC stops). If TRUE and *bEnPwrOnVal* = FALSE, the output goes to 0 in the event of a K-bus error (or PLC stopped). If TRUE and *bEnPwrOnVal* = TRUE, the output adopts the value of *wPwrOnVal* in the event of a K-bus error (or PLC stopped).

bEnEcoMode: Configuration: Full Scale Mode [▶ 60] or ECO Mode [▶ 60]. TRUE: ECO Mode. FALSE: Full Scale Mode.

bEnPwrOnVal: Configuration: TRUE: In the event of a K-bus error (or PLC stopped), the analog output is set to the value of *wPowerOnValue* if *bDisWatchdog* = TRUE.

wPwrOnVal: Configuration: Value that the output is to adopt in the event of a K-bus error (or PLC stopped).

wOut: Value of the analog output signal, provided that the switch is set to "auto". 0 V = 0000_{hex} = 0_{dec} , 10 V = $7FFF_{hex}$ = 32767_{dec}. If the switch is set to "man" the value of the potentiometer is output.

dwOpt: For future options.

VAR_OUTPUT

```
bBusy          : BOOL;
bStaSwiAuto    : BOOL;
wStaPoti       : WORD;
wStaOut        : WORD;
bErr           : BOOL;
udiErrID       : UDINT;
```

bBusy: *bBusy* is TRUE as long as the configuration is being written (start with *bExecCfg*). Note that the inputs are not updated during this time.

bStaSwiAuto: Status of the switch. TRUE = "auto" FALSE = "man".

wStaPoti: Status of the potentiometer. CCW stop = 0000_{hex} = 0_{dec} = 0 V, CW stop = $7FFF_{hex}$ = 32767_{dec} = 10 V.

wStaOut: Status of the analog output signal. CCW stop = 0000_{hex} = 0_{dec} = 0 V, CW stop = $7FFF_{hex}$ = 32767_{dec} = 10 V.

bErr: The *bErr*or output goes TRUE as soon as an error occurs. This error is described via the *udiErrID* variable.

udiErrID: In the event of an error the output issues an error code (see error codes [▶ 59]). *bError* goes TRUE at the same time.

VAR_IN_OUT

```
stInData       : ST_KL8548InData;
stOutData      : ST_KL8548OutData;
```

stInData: Process image of the inputs (see ST_KL8548InData [▶ 58]).

stOutData: Process image of the outputs (see ST_KL8548OutData [▶ 58]).

4.2.4.2 FB_KL8548Ex

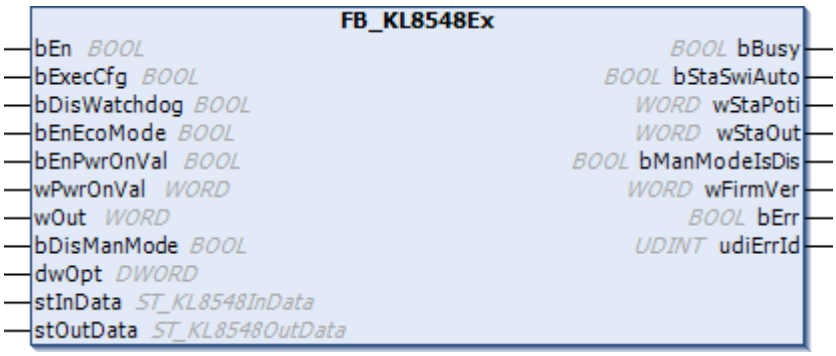


Fig. 30: Function block FB_KL8548Ex

Application

This function block is used to configure the KL8548, to read out the digital signals and to set the outputs/ LEDs.

The function block must be called once per cycle.

The parameters are transferred to the terminal with a positive edge on *bExecCfg*.

The standard function of the LEDs cannot be deactivated.

VAR_INPUT

```
bEn             : BOOL;
bExecCfg        : BOOL;
bDisWatchdog    : BOOL;
bEnEcoMode      : BOOL;
bEnPwrOnVal     : BOOL;
wPwrOnVal       : WORD;
wOut            : WORD;
bDisManMode     : BOOL;
dwOpt           : DWORD;
```

bEn: The function block is enabled by setting *bEn* to TRUE in the PLC program. With *bEn* = FALSE the execution of the function block will be stopped and all outputs reset.

bExecCfg: The configurations *bDisWatchdog*, *bEnEcoMode*, *bEnPwrOnVal* and *wPwrOnVal* are written to the terminals when a positive edge is encountered. *bBusy* goes TRUE. Note that the inputs are not updated during this time.

bDisWatchdog: Configuration: If False the last value of *wStaOut* remains unchanged in the event of a K-bus error (the value is set to 0 if the PLC stops). If True and *bEnPwrOnVal* =False, the output goes to 0 in the event of a K-bus error (or PLC stopped). If True and *bEnPwrOnVal* =True, the output adopts the value of *wPwrOnVal* in the event of a K-bus error (or PLC stopped).

bEnEcoMode: Configuration: Full Scale Mode or ECO Mode. TRUE: ECO Mode. FALSE: Full Scale Mode.

bEnPwrOnVal: Configuration: TRUE: In the event of a K-bus error (or PLC stopped), the analog output is set to the value of *wPowerOnValue* if *bDisWatchdog* = True.

wPwrOnVal: Configuration: Value that the output is to adopt in the event of a K-bus error (or PLC stopped).

wOut: Value of the analog output signal, provided that the switch is set to "auto". 0 V = 0000_{hex} = 0_{dec} , 10 V = $7FFF_{hex}$ = 32767_{dec}. If the switch is set to "man" the value of the potentiometer is output.

bDisManMode: Disabling manual operating mode.

dwOpt: For future options.

VAR_OUPUT

```
bBusy           : BOOL;
bStaSwiAuto     : BOOL;
wStaPoti        : WORD;
wStaOut         : WORD;
bManModelsDis   : BOOL;
wFirmVer        : WORD;
bErr            : BOOL;
udiErrID        : UDINT;
```

bBusy: *bBusy* is TRUE as long as the configuration is being written (start with *bExecCfg*). Note that the inputs are not updated during this time.

bStaSwiAuto: Status of the switch. True ="auto" False = "man".

wStaPoti: Status of the potentiometer. CCW stop = 0000_{hex} = 0_{dec} = 0 V, CW stop = $7FFF_{hex}$ = 32767_{dec} = 10 V

wStaOut: Status of the analog output signal. CCW stop = 0000_{hex} = 0_{dec} = 0 V, CW stop = $7FFF_{hex}$ = 32767_{dec} = 10 V

bManModelsDis: Manual operating mode disabled.

wFirmVer: Issues the firmware version.

bErr: The *bErr*or output goes TRUE as soon as an error occurs. This error is described via the *udiErrID* variable.

udiErrID: In the event of an error the output issues an error code (see error codes). *bError* goes TRUE at the same time.

VAR_IN_OUT

```
stInData      : ST_KL8548InData;
stOutData     : ST_KL8548OutData;
```

stInData: Process image of the inputs (see ST_KL8548InData).

stOutData: Process image of the outputs (see ST_KL8548OutData).

4.2.5 KL85xx - Helper

Function blocks	Description
FB_KL85xx16BitToWord [▶ 52]	Conversion from 16 bits to 1 Word
FB_KL85xx8BitToByte [▶ 53]	Conversion from 8 bits to 1 byte
FB_KL85xxByteTo8Bit [▶ 54]	Conversion from 1 byte to 8 bits.
FB_KL85xxWordTo16Bit [▶ 54]	Conversion from 1 word to 16 bits.

4.2.5.1 FB_KL85xx16BitToWord

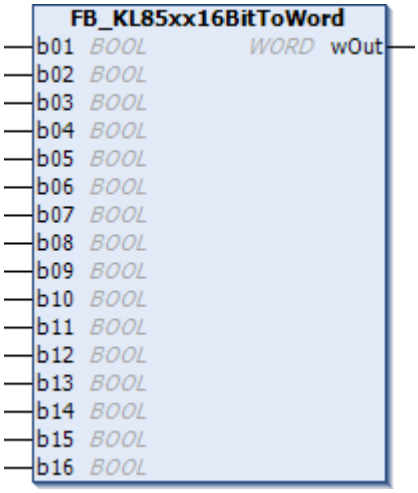


Fig. 31: Function block FB_KL85xx16BitToWord

Application

This function block converts 16 bits to 1 word.

VAR_INPUT

```
b01       : BOOL;
b02       : BOOL;
b03       : BOOL;
b04       : BOOL;
b05       : BOOL;
b06       : BOOL;
b07       : BOOL;
b08       : BOOL;
b09       : BOOL;
b10       : BOOL;
b11       : BOOL;
b12       : BOOL;
b13       : BOOL;
b14       : BOOL;
b15       : BOOL;
b16       : BOOL;
```

b01: Bit 0
b02: Bit 1
b03: Bit 2
b04: Bit 3

b05: Bit 4
b06: Bit 5
b07: Bit 6
b08: Bit 7
b09: Bit 8
b10: Bit 9
b11: Bit 10
b12: Bit 11
b13: Bit 12
b14: Bit 13
b15: Bit 14
b16: Bit 15

VAR_OUTPUT

```
wOut      : WORD;
```

wOut: Word composed of the 16 bits.

4.2.5.2 FB_KL85xx8BitToByte

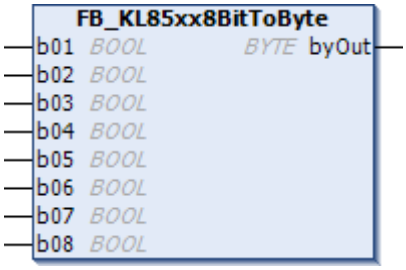


Fig. 32: Function block FB_KL85xx8BitToByte

Application

This function block converts 8 bits to 1 byte.

VAR_INPUT

```
b01     : BOOL;
b02     : BOOL;
b03     : BOOL;
b04     : BOOL;
b05     : BOOL;
b06     : BOOL;
b07     : BOOL;
b08     : BOOL;
```

b01: Bit 0
b02: Bit 1
b03: Bit 2
b04: Bit 3
b05: Bit 4
b06: Bit 5
b07: Bit 6
b08: Bit 7

VAR_OUTPUT

```
byOut      : BYTE;
```

byOut: Byte composed of the 8 bits.

4.2.5.3 FB_KL85xxByteTo8Bit

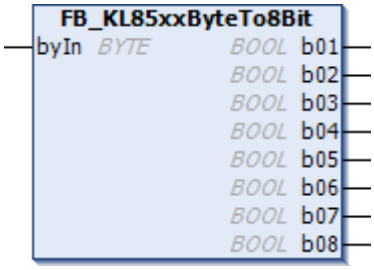


Fig. 33: Function block FB_KL85xxByteTo8Bit

Application

This function block converts 1 byte to 8 bits.

VAR_INPUT

```
byIn      : BYTE;
```

byIn: Byte to be broken down into 8 bits.

VAR_OUTPUT

```
b01       : BOOL;
b02       : BOOL;
b03       : BOOL;
b04       : BOOL;
b05       : BOOL;
b06       : BOOL;
b07       : BOOL;
b08       : BOOL;
```

b01: Bit 0
b02: Bit 1
b03: Bit 2
b04: Bit 3
b05: Bit 4
b06: Bit 5
b07: Bit 6
b08: Bit 7

4.2.5.4 FB_KL85xxWordTo16Bit

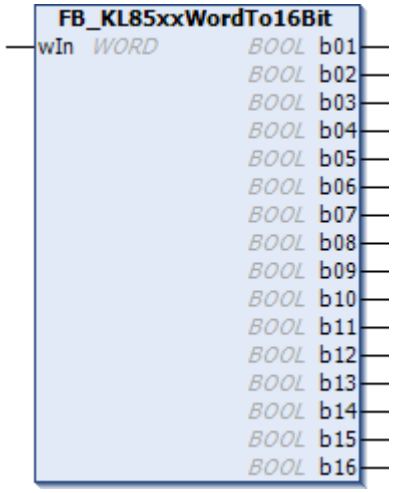


Fig. 34: Function block FB_KL85xxWordTo16Bit

Application

This function block converts 1 word to 16 bits.

VAR_INPUT

```
wIn      : WORD;
```

wIn: Word to be broken down into 16 bits.

VAR_OUTPUT

```
b01      : BOOL;
b02      : BOOL;
b03      : BOOL;
b04      : BOOL;
b05      : BOOL;
b06      : BOOL;
b07      : BOOL;
b08      : BOOL;
b09      : BOOL;
b10      : BOOL;
b11      : BOOL;
b12      : BOOL;
b13      : BOOL;
b14      : BOOL;
b15      : BOOL;
b16      : BOOL;
```

b01: Bit 0
b02: Bit 1
b03: Bit 2
b04: Bit 3
b05: Bit 4
b06: Bit 5
b07: Bit 6
b08: Bit 7
b09: Bit 8
b10: Bit 9
b11: Bit 10
b12: Bit 11
b13: Bit 12
b14: Bit 13
b15: Bit 14
b16: Bit 15

4.3 Data types

Hardware types

Data type	Description
ST_KL8519InData [▶ 56]	Process image of the inputs for the KL8519
ST_KL8519OutData [▶ 56]	Process image of the outputs for the KL8519
ST_KL8524InData [▶ 57]	Process image of the inputs for the KL8524
ST_KL8524OutData [▶ 57]	Process image of the outputs for the KL8524
ST_KL8528InData [▶ 57]	Process image of the inputs for the KL8528
ST_KL8528OutData [▶ 58]	Process image of the outputs for the KL8528
ST_KL8548InData [▶ 58]	Process image of the inputs for the KL8548
ST_KL8548OutData [▶ 58]	Process image of the outputs for the KL8548

Data type	Description
E_KL8519_KBusOffReact [▶ 56]	Reaction of the LEDs of the KL8519 in the event of a K-bus error

4.3.1 E_KL8519_KBusOffReact

Reaction of the LEDs of the KL8519 in the event of a K-bus error. These LEDs must have been activated via "wDisCh".

```
TYPE E_KL8519_KBusOffReact :
(
  eKL8519_WatchdogOff [▶ 56]      := 0,
  eKL8519_Watchdog500ms [▶ 56]    := 1,
  eKL8519_Watchdog1000ms [▶ 56]   := 2
)
END_TYPE
```

eKL8519_WatchdogOff: In the case of a K-bus error, all LEDs set via the PLC turn off.

eKL8519_Watchdog500ms: The last state of the LED toggles every 500 ms. If the last state was OFF the LED remains OFF.

eKL8519_Watchdog1000ms: The last state of the LED toggles every 1000 ms. If the last state was OFF the LED remains OFF.

4.3.2 ST_KL8519InData

Process image of the inputs for the KL8519

Linked to the terminals in the System Manager.

```
TYPE ST_KL8519InData :
STRUCT
  byState     : BYTE;
  byDataIN_1  : BYTE;
  byDataIN_2  : BYTE;
END_STRUCT
END_TYPE
```

byState: Status byte

byDataIN_1: Data byte 1: Input 1 to 8

byDataIN_2: Data byte 2: Input 9 to 16

4.3.3 ST_KL8519OutData

Process image of the outputs for the KL8519

Linked to the terminals in the System Manager.

```
TYPE ST_KL8519OutData :
STRUCT
  byCtrl           : BYTE;
  byLEDoutGreen_1  : BYTE;
  byLEDoutGreen_2  : BYTE;
  byLEDoutRed_1    : BYTE;
  byLEDoutRed_2    : BYTE;
END_STRUCT
END_TYPE
```

byCtrl: Control byte

byLEDoutGreen_1: Data byte 1: Green LED 1 to 8 (must be enabled via *wDisCh*)

byLEDoutGreen_2: Data byte 2: Green LED 9 to 16 (must be enabled via *wDisCh*)

byLEDoutRed_1: Data byte 3: red LED 1 to 8 (must be enabled via *wDisCh*)

byLEDoutRed_2: Data byte 4: red LED 9 to 16 (must be enabled via *wDisCh*)

4.3.4 ST_KL8524InData

Process image of the inputs for the KL8524

Linked to the terminals in the System Manager.

```
TYPE ST_KL8524InData :
STRUCT
  byState       : BYTE;
  byAutoManual  : BYTE;
  bySwitchMode  : BYTE;
  byOnOff       : BYTE;
END_STRUCT
END_TYPE
```

byState: Status byte

byAutoManual: Data byte 1: Position of the auto/man switch (bit 0 = channel 1, bit 3 = channel 4)

bySwitchMode: Data byte 2: Position of the three-step switch (bit 0 = channel 1 position 1, bit 1 = channel 1 position 2 .. , bit 6 = channel 1 position 1, bit 7 = channel 4 position 2)

byOnOff: Data byte 3: State of the physical outputs (bit 0 = channel 1 output 1, bit 1 = channel 1 output 2, ..., bit 6 = channel 1 output 1, bit 7 = channel 4 output 2)

4.3.5 ST_KL8524OutData

Process image of the outputs for the KL8524

Linked to the terminals in the System Manager.

```
TYPE ST_KL8524OutData :
STRUCT
  byCtrl           : BYTE;
  bySwitchModeOut  : BYTE;
  byData           : BYTE;
  byLEDGreen       : BYTE;
  byLEDYellow      : BYTE;
END_STRUCT
END_TYPE
```

byCtrl: Control byte

bySwitchModeOut: Data byte 1: sets the outputs of the three-step switch via the PLC if the automatic switch is set to "auto"

byData: Data byte 2: reserved for register communication

byLEDGreen: Data byte 3: sets the output LEDs to green (must be enabled via *byDisCh*)

byLEDYellow: Data byte 4: sets the output LEDs to yellow (must be enabled via *byDisCh*)

4.3.6 ST_KL8528InData

Process image of the inputs for the KL8528

Linked to the terminals in the System Manager.

```
TYPE ST_KL8528InData :
STRUCT
  byState       : BYTE;
  byAutoManual  : BYTE;
  byOnOff       : BYTE;
END_STRUCT
END_TYPE
```

byState: Status byte

byAutoManual: Data byte 1: Switch position "auto" for output 1 to 8 (the output is set via the PLC)

byOnOff: Data byte 2: Switch position "on" for output 1 to 8 (the output is set independently of the PLC)

4.3.7 ST_KL8528OutData

Process image of the outputs for the KL8528

Linked to the terminals in the System Manager.

```
TYPE ST_KL8528OutData :
STRUCT
  byCtrl          : BYTE;
  byOut           : BYTE;
  byData          : BYTE;
  byLEDOutGreen   : BYTE;
  byLEDOutYellow  : BYTE;
END_STRUCT
END_TYPE
```

byCtrl: Control byte

byOut: Data byte 1: sets output 1 to 8 via the PLC if the respective switch is set to "auto".

byData: Data byte 2: reserved for register communication

byLEDOutGreen: Data byte 3: sets LED 1 to 8 to green (must be enabled via *byDisCh*)

byLEDOutYellow: Data byte 4: sets LED 1 to 8 to yellow (must be enabled via *byDisCh*)

4.3.8 ST_KL8548InData

Process image of the inputs for the KL8548

Linked to the terminals in the System Manager.

```
TYPE ST_KL8548InData :
STRUCT
  byState  : BYTE;
  byDummy  : BYTE;
  wDataIn  : WORD;
END_STRUCT
END_TYPE
```

byState: Status byte: Bit 0 is TRUE if the automatic switch is set to "man" (Status byte=1).

byDummy: Dummy byte.

wDataIn: Data word 1: position of the potentiometer (0x0000=0V, 0x3FFF=+5V, 0x7FFF=+10V)

4.3.9 ST_KL8548OutData

Process image of the outputs for the KL8548

Linked to the terminals in the System Manager.

```
TYPE ST_KL8548OutData :
STRUCT
  byCtrl    : BYTE;
  byDummy   : BYTE;
  wDataOut  : WORD;
END_STRUCT
END_TYPE
```

byCtrl: Control byte

byDummy: Dummy byte.

wDataOut: Data word 1: sets the output value via the PLC if the respective switch is set to "auto". (0x0000=0V, 0x3FFF=+5V, 0x7FFF=+10V)

4.4 Error codes

Value (hex)	Value (dec)	Description
0x0000	0	No error
0x8001	32769	Watchdog Error. The terminal does not respond. This message usually means that there is no connection to the terminal. • Is the terminal linked to the variables in the TwinCAT System Manager? • Terminal plugged in incorrectly? • Clean all, rebuild all and load again into the TwinCAT System Manager?
0x8002	32770	Incorrect terminal connected. The terminal type can be checked in online mode (*uiTerm* - internal variable of the function block).
0x8003	32771	The input variable *wOut* of the function block FB_KL8548() [▶ 49] is too big. Maximum 32767 is allowed. Processing continues internally (in the function block) with 32767 (corresponds to 10 V).
0x8004	32772	The function block FB_KL8524() [▶ 41] tries to set the outputs 1 and 2 simultaneously on channel 1. That is not allowed. No output is set.
0x8005	32773	The function block FB_KL8524() [▶ 41] tries to set the outputs 1 and 2 simultaneously on channel 2. That is not allowed. No output is set.
0x8006	32774	The function block FB_KL8524() [▶ 41] tries to set the outputs 1 and 2 simultaneously on channel 3. That is not allowed. No output is set.
0x8007	32775	The function block FB_KL8524() [▶ 41] tries to set the outputs 1 and 2 simultaneously on channel 4. That is not allowed. No output is set.

4.5 Adjustment of the LED displays

4.5.1 Standard function LED KL8519

In delivery state the LEDs light up green when the respective input is occupied and are off when the input is not occupied.

The variable *wSetCol* can be used to specify that the respective LED lights up red.

The channel can be inverted with the variable *wInv*. This means that the LEDs light up if the input is not occupied and are off if the input is occupied.

Using the variable *wBiCol* the LEDs can be activated in two colors. This means that the LEDs light up green if the input is occupied and red if the input is not occupied. This behavior is reversed if the channels are inverted with *wInv*.

This functionality can be disabled with the variable *wDisCh*. The LEDs can then be set from the PLC.

4.5.2 Standard function LED KL8524

The LEDs light up steadily yellow if the automatic switch is set to **auto** and the respective output is FALSE.

The LEDs light up steadily green if the automatic switch is set to **auto** and the respective output is TRUE (set by the PLC).

The LEDs flash yellow if the automatic switch is set to "man" and the respective output is FALSE.

The LEDs flash green if the automatic switch is set to "man" and the respective output is TRUE (set by the three-step switch).

This functionality can be disabled with the variable *byDisCh*. The LEDs can then be set from the PLC.

4.5.3 Standard function LED KL8528

The LEDs light up steadily yellow if the switch is set to **auto** and the output is FALSE.

The LEDs light up steadily green if the switch is set to **auto** and the output is TRUE (set by the PLC).

The LEDs flash yellow if the automatic switch is set to **off**. The respective output is then FALSE (even if it is set by the PLC).

The LEDs flash green if the automatic switch is set to **man**. The respective output is then TRUE.

This functionality can be disabled with the variable *byDisCh*. The LEDs can then be set from the PLC.

4.5.4 Standard function LED KL8548

The LEDs are off if the switch is set to **auto**. The bar graph indicates the PLC value.

The LEDs light up steadily if the switch is set to **man**. The bar graph indicates the potentiometer value.

This functionality cannot be disabled.

4.5.5 Bar graph display mode - KL8548

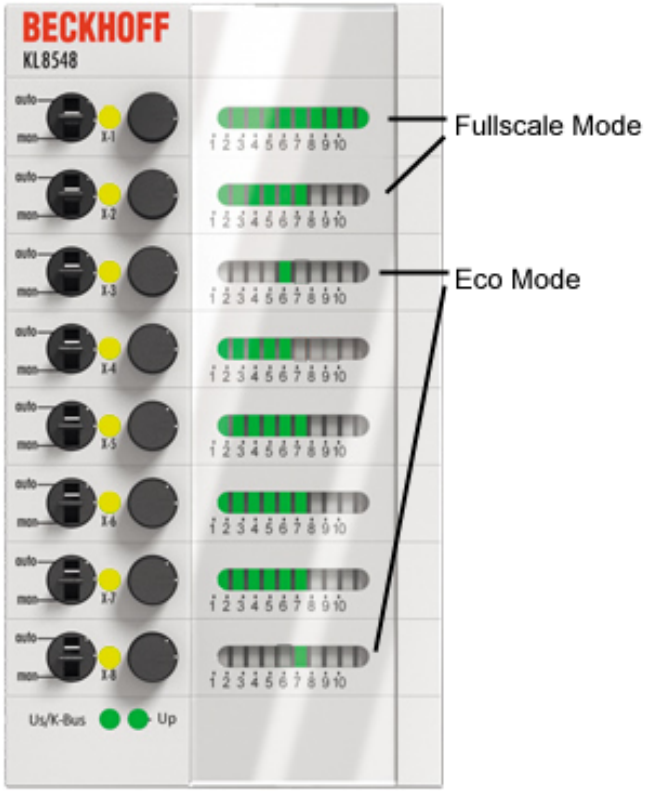


Fig. 35: KL8548 - Bar graph display mode

The KL8548 offers two bar graph display modes for the analog values.

In full scale mode the current analog value is shown as a full bar.

In ECO mode the analog value is shown with one LED only.

The display mode can be selected separately for each analog value by writing to register R32.4 [▶ 86] or with the function blocks FB_KL8548 [▶ 49] or FB_KL8548Ex [▶ 50].

5 Data structures of the modules

5.1 KL8519

5.1.1 KL8519 – Process image

The KL8519 is represented in the process image with 6 bytes each of input and output data. These are organized as follows:

Byte offset	Format	Input data	Output data
0	Byte	Status byte (SB [▶ 62])	Control byte (CB [▶ 62])
1	Byte	Data byte 0 (input 1 to 8)	Data byte 0 (green LED 1 to 8)
2	Byte	Data byte 1 (input 9 to 16)	Data byte 1 (green LED 9 to 16)
3	Byte	reserved	Data byte 2 (red LED 1 to 8)
4	Byte	reserved	Data byte 3 (red LED 9 to 16)
5	Byte	reserved	reserved

5.1.2 KL8519 - control and status bytes

Channel 1

Process data mode

Control byte 1 in process data mode

Control byte 1 (CB1) is located in the output image [▶ 62], and is transmitted from the controller to the terminal.

Bit	CB1.7	CB1.6	CB1.5	CB1.4	CB1.3	CB1.2	CB1.1	CB1.0
Name	RegAccess	-	-	-	-	-	-	-

Key

Bit	Name	Description	
CB1.7	RegAccess	0_{bin}	Register communication off (process data mode)
CB1.6 - CB1.0	-	0_{bin}	reserved

Status byte 1 in process data mode

The status byte 1 (SB1) is located in the input image [▶ 62] and is transmitted from terminal to the controller.

Bit	SB1.7	SB1.6	SB1.5	SB1.4	SB1.3	SB1.2	SB1.1	SB1.0
Name	RegAccess	-	-	-	-	-	-	-

Key

Bit	Name	Description	
SB1.7	RegAccess	0_{bin}	Acknowledgment for process data mode
SB1.6 - SB1.0	-	0_{bin}	reserved

Register communication

Control byte 1 in register communication

Control byte 1 (CB1) is located in the output image [▶ 62], and is transmitted from the controller to the terminal.

Bit	CB1.7	CB1.6	CB1.5	CB1.4	CB1.3	CB1.2	CB1.1	CB1.0
Name	RegAccess	R/W	Reg. no.					

Key

Bit	Name	Description	
CB1.7	RegAccess	1_{bin}	Register communication switched on
CB1.6	R/W	0_{bin}	Read access
		1_{bin}	Write access
CB1.5 to CB1.0	Reg. no.	Register number: Enter the number of the register that you - want to read with input data word DataIN1 [▶ 62] or - want to write with output data word DataOUT1 [▶ 62].	

Status byte 1 in register communication

The status byte 1 (SB1) is located in the input image [▶ 62] and is transmitted from terminal to the controller.

Bit	SB1.7	SB1.6	SB1.5	SB1.4	SB1.3	SB1.2	SB1.1	SB1.0
Name	RegAccess	R/W	Reg. no.					

Key

Bit	Name	Description	
SB1.7	RegAccess	1_{bin}	Acknowledgment for register access
SB1.6	R	0_{bin}	Read access
SB1.5 to SB1.0	Reg. no.	Number of the register that was read or written.	

Channels 2 to 8

The control and status bytes of channels 2 to 8 are structured like the control and status byte of channel 1.

5.1.3 KL8519 – Register overview

The registers are used for the parameterization of the Bus Terminals and are available for each channel. They can be read or written by means of the register communication [▶ 63].

Register no.	Comment	Default value		R/W	Memory
R0	reserved	-	-	-	-
...	...	...	...	...	...
R6	reserved	-	-	-	-
R7 [▶ 64]	Command register	0x0000	0_{dec}	R/W	RAM
R8 [▶ 65]	Terminal type	0x2147	8519_{dec}	R	ROM
R9 [▶ 65]	Firmware version (ASCI)	e.g. 0x3141	e.g. 12609_{dec}	R	ROM
R10	Multiplex shift register			R	ROM
R11	Signal channels			R	ROM
R12 [▶ 65]	minimum data length of a channel			R	ROM
R13	Data structure			R	ROM
R14	reserved	-	-	-	-
R15	Alignment register			R/W	RAM
R16 [▶ 65]	Hardware version number	e.g. 0x0000	e.g. 0_{dec}	R/W	SEEPROM
R17	reserved	-	-	-	-
...	...	...	...	...	...
R30	reserved	-	-	-	-
R31 [▶ 65]	Code word register	0x0000	0_{dec}	R/W	RAM
R32 [▶ 65]	Disable Channel LED	0x0000	0_{dec}	R/W	SEEPROM
R33 [▶ 65]	LED Color	0x0000	0_{dec}	R/W	SEEPROM
R34 [▶ 66]	Inv LED	0x0000	0_{dec}	R/W	SEEPROM
R35 [▶ 66]	LED BiColor	0x0000	0_{dec}	R/W	SEEPROM
R36 [▶ 67]	K-bus off reaction	0x0000	0_{dec}	R/W	SEEPROM
R37	reserved	-	-	-	-
...	reserved	-	-	-	-
R63	reserved	-	-	-	-

5.1.4 KL8519 – Register description

The registers are used for the parameterization of the Bus Terminals and are available for each channel. They can be read or written by means of the register communication.

R7: Command register

To execute a standard command, the user code word 0x1235 must be entered in register R31 [▶ 65].

Command 0x7000: Restore Factory Settings

Entering 0x7000 in register R7 restores the delivery state for the following registers.

- Register R32 [▶ 65]: 0x0000
- Register R33 [▶ 65]: 0x0000
- Register R34 [▶ 66]: 0x0000
- Register R35 [▶ 66]: 0x0000
- Register R36 [▶ 67]: 0x0000
  to

- Register R63: 0x0000

R8: Terminal description

The name of the terminal is contained in register R8.
KL8519: 0 x 2147 (8519_{dec})

R9: Firmware version

Register R9 contains the ASCII coding of the terminal's firmware version, e.g. **0x3141 = '1A'**. The **'0x31'** corresponds here to the ASCII character **'1'**, while the **'0x41'** represents the ASCII character **'A'**.
This value cannot be changed.

R12: Minimum data length of a channel

R16: Hardware version number

Register R16 contains the hardware version of the terminal.

R31: Code word register

- If you write values into the user registers without first entering the user code word (0x1235) into the code word register, the terminal will not accept the supplied data.
- If you write values into the user registers and have previously entered the user code word (0x1235) in the code word register, these values are stored in the RAM registers and in the SEEPROM registers and are therefore retained when the terminal is restarted.

The code word is reset when the terminal is restarted.

R32: Disable Channel LED

Using this register you can deactivate the status LEDs of the individual channels.

Bit	R32.15	R32.14	R32.13	R32.12	R32.11	R32.10	R32.9	R32.8
Name	disLED_Ch16	disLED_Ch15	disLED_Ch14	disLED_Ch13	disLED_Ch12	disLED_Ch11	disLED_Ch10	disLED_Ch9

Bit	R32.7	R32.6	R32.5	R32.4	R32.3	R32.2	R32.1	R32.0
Name	disLED_Ch8	disLED_Ch7	disLED_Ch6	disLED_Ch5	disLED_Ch4	disLED_Ch3	disLED_Ch2	disLED_Ch1

Key

Bit	Name	Description		default
R32.15	disLED_Ch16	0_{bin}	The LED indicates the state of input 16. Further properties can be set with R33, R34 and R35.	0_{bin}
		1_{bin}	The LED does not indicate the state of input 16. However, the state of input 16 continues to be transmitted to the PLC. In addition the PLC can now switch the LED selectively to green or red.	
R32.14	disLED_Ch15	0_{bin}	The LED indicates the state of input 15. Further properties can be set with R33, R34 and R35.	0_{bin}
		1_{bin}	The LED does not indicate the state of input 15. However, the state of input 15 continues to be transmitted to the PLC. In addition the PLC can now switch the LED selectively to green or red.	
...	...	...		...
R32.0	disLED_Ch1	0_{bin}	The LED indicates the state of input 1. Further properties can be set with R33, R34 and R35.	0_{bin}
		1_{bin}	The LED does not indicate the state of input 1. However, the state of input 1 continues to be transmitted to the PLC. In addition the PLC can now switch the LED selectively to green or red.	

R33: LED Set Color

Using this register you can define the color for the status LEDs of the individual channels (green or red).

Bit	R33.15	R33.14	R33.13	R33.12	R33.11	R33.10	R33.9	R33.8
Name	colLED_Ch16	colLED_Ch15	colLED_Ch14	colLED_Ch13	colLED_Ch12	colLED_Ch11	colLED_Ch10	colLED_Ch9

Bit	R33.7	R33.6	R33.5	R33.4	R33.3	R33.2	R33.1	R33.0
Name	colLED_Ch8	colLED_Ch7	colLED_Ch6	colLED_Ch5	colLED_Ch4	colLED_Ch3	colLED_Ch2	colLED_Ch1

Key

Bit	Name	Description		default
R33.15	colLED_Ch16	0_{bin}	The LED of input 16 lights up green.	0_{bin}
		1_{bin}	The LED of input 16 lights up red.	
R33.14	colLED_Ch15	0_{bin}	The LED of input 15 lights up green.	0_{bin}
		1_{bin}	The LED of input 15 lights up red.	
...	...	...		...
R33.0	colLED_Ch1	0_{bin}	The LED of input 1 lights up green.	0_{bin}
		1_{bin}	The LED of input 1 lights up red.	

R34: Inv LED

Using this register you can invert the display of the status LEDs of the individual channels. The process data transmitted to the PLC are not affected by this.

Bit	R34.15	R34.14	R34.13	R34.12	R34.11	R34.10	R34.9	R34.8
Name	invLED_Ch16	invLED_Ch15	invLED_Ch14	invLED_Ch13	invLED_Ch12	invLED_Ch11	invLED_Ch10	invLED_Ch9

Bit	R34.7	R34.6	R34.5	R34.4	R34.3	R34.2	R34.1	R34.0
Name	invLED_Ch8	invLED_Ch7	invLED_Ch6	invLED_Ch5	invLED_Ch4	invLED_Ch3	invLED_Ch2	invLED_Ch1

Key

Bit	Name	Description		default
R34.15	invLED_Ch16	0_{bin}	The LED of input 16 lights up if the input is active (1_{bin}).	0_{bin}
		1_{bin}	The LED of input 16 lights up if the input is inactive (0_{bin}).	
R34.14	invLED_Ch15	0_{bin}	The LED of input 15 lights up if the input is active (1_{bin}).	0_{bin}
		1_{bin}	The LED of input 15 lights up if the input is inactive (0_{bin}).	
...	...	...		...
R34.0	invLED_Ch1	0_{bin}	The LED of input 1 lights up if the input is active (1_{bin}).	0_{bin}
		1_{bin}	The LED of input 1 lights up if the input is inactive (0_{bin}).	

R35: LED BiColor

Using this register you can switch the display of the status LEDs of the individual channels between two colors.

Bit	R35.15	R35.14	R35.13	R35.12	R35.11	R35.10	R35.9	R35.8
Name	invLED_Ch16	invLED_Ch15	invLED_Ch14	invLED_Ch13	invLED_Ch12	invLED_Ch11	invLED_Ch10	invLED_Ch9

Bit	R35.7	R35.6	R35.5	R35.4	R35.3	R35.2	R35.1	R35.0
Name	invLED_Ch8	invLED_Ch7	invLED_Ch6	invLED_Ch5	invLED_Ch4	invLED_Ch3	invLED_Ch2	invLED_Ch1

Key

Bit	Name	Description		default
R35.15	invLED_Ch16	0_{bin}	The LED of input 16 lights up green if the input is inactive (0_{bin}).	0_{bin}
		1_{bin}	The LED of input 16 lights up red if the input is active (1_{bin}).	
R35.14	invLED_Ch15	0_{bin}	The LED of input 15 lights up green if the input is inactive (0_{bin}).	0_{bin}
		1_{bin}	The LED of input 15 lights up red if the input is active (1_{bin}).	
...	...	...		...
R35.0	invLED_Ch1	0_{bin}	The LED of input 1 lights up green if the input is inactive (0_{bin}).	0_{bin}
		1_{bin}	The LED of input 1 lights up red if the input is active (1_{bin}).	

These displays can also be inverted using register 34.

R36: Reaction to K-bus errors

In the event of a K-bus error, all LEDs set via the PLC (R32) are off. Unless R36 contains a value other than 00_{bin}.

Bit	R36.15	R36.14	R36.13	R36.12	R36.11	R36.10	R36.9	R36.8
Name	-	-	-	-	-	-	-	-

Bit	R36.7	R36.6	R36.5	R36.4	R36.3	R36.2	R36.1	R36.0
Name	-	-	-	-	-	-	K-bus-OFF reaction	

Key

Bit	Name	Description		default
R36.15	-	reserved		0_{bin}
...	...	...		...
R36.2	-	reserved		0_{bin}
R36.1, R36.0	K-bus-OFF reaction	00_{bin}	In the event of a K-bus error all LEDs that have been set via the PLC (R32) remain on.	00_{bin}
		01_{bin}	The last state of the LED toggles every 500 ms. If the last state was OFF the LED remains OFF.	
		10_{bin}	The last state of the LED toggles every 1000 ms. If the last state was OFF the LED remains OFF.	

5.2 KL8524

5.2.1 KL8524 – Process image

The KL8524 is represented in the process image with 6 bytes each of input and output data. These are organized as follows:

Byte offset	Format	Input data	Output data
0	Byte	Status byte (SB [▶ 70])	Control byte (CB [▶ 70])
1	Byte	Data byte 1: position of the auto/man switch	Data byte 1: overwrites the position of the three-step switch in automatic mode
2	Byte	Data byte 2: position of the three-step switch	Data byte 2: reserved for register communication
3	Byte	Data byte 3: state of the physical outputs	Data byte 3: sets the output LEDs to green
4	Byte	Data byte 4: reserved	Data byte 4: sets the output LEDs to yellow
5	Byte	Data byte 5: reserved	Data byte 5: reserved

Control and status byte

See chapter Control and status byte [▶ 70]

Process data

Input data

Data byte 1

Data byte 1 indicates the positions of the switches auto/man.

Bit	Name	Description	
0	enAutoC1	0_{bin}	auto/man switch of channel 1 is set to man: the three-step switch controls this channel.
		1_{bin}	auto/man switch of channel 1 is set to auto: the PLC controls this channel.
1	enAutoC2	0_{bin}	auto/man switch of channel 2 is set to man: the three-step switch controls this channel.
		1_{bin}	auto/man switch of channel 2 is set to auto: the PLC controls this channel.
2	enAutoC3	0_{bin}	auto/man switch of channel 3 is set to man: the three-step switch controls this channel.
		1_{bin}	auto/man switch of channel 3 is set to auto: the PLC controls this channel.
3	enAutoC4	0_{bin}	auto/man switch of channel 4 is set to man: the three-step switch controls this channel.
		1_{bin}	auto/man switch of channel 4 is set to auto: the PLC controls this channel.
4	-	reserved	
...	...	...	
7	-	reserved	

Data byte 2

Data byte 2 indicates the positions of the three-step switch.

Bit	Channel	Description	
1 and 0	Channel 1	Bits	Three-step switch
		00_{bin}	Position 0
		01_{bin}	Position 1
		10_{bin}	Position 2
		11_{bin}	- (not possible)
3 and 2	Channel 2	See channel 1	
5 and 4	Channel 3	See channel 1	
7 and 6	Channel 4	See channel 1	

Data byte 3

Data byte 3 indicates the states of the physical outputs of the KL8524.

Bit	Name	Description	
0	stateC1.1	0_{bin}	Output C1.1 is not set
		1_{bin}	Output C1.1 is set
1	stateC1.2	0_{bin}	Output C1.2 is not set
		1_{bin}	Output C1.2 is set
...	...	...	
6	stateC4.2	0_{bin}	Output C4.1 is not set
		1_{bin}	Output C4.1 is set
7	stateC4.2	0_{bin}	Output C4.2 is not set
		1_{bin}	Output C4.2 is set

Output data

Data byte 1

Data byte 1 overwrites the position of the three-step switch in automatic mode.

Bit	Channel	Description	
1 and 0	Channel 1	Bits	Three-step switch
		00_{bin}	Position 0
		01_{bin}	Position 1
		10_{bin}	Position 2
		11_{bin}	Position 0*
3 and 2	Channel 2	See channel 1	
5 and 4	Channel 3	See channel 1	
7 and 6	Channel 4	See channel 1	

*) since switch position 3 does not exist, the channel behaves as if no bit had been set when setting both bits, corresponding to switch position 0.

Data byte 2

Has no function (reserved) in process data mode, but is used for register communication.

Data byte 3

Data byte 3 sets the LEDs of the channels to green.

Bit	Name	Description	
0	LED C1.1 green	1_{bin}	LED K1.1 lights up green
1	LED C1.2 green	1_{bin}	LED K1.2 lights up green
...	...	...	
6	LED C4.1 green	1_{bin}	LED K4.1 lights up green
7	LED C4.2 green	1_{bin}	LED K4.2 lights up green

Data byte 4

Data byte 4 sets the LEDs of the channels to yellow.

Bit	Name	Description	
0	LED C1.1 yellow	1_{bin}	LED K1.1 lights up yellow
1	LED C1.2 yellow	1_{bin}	LED K1.2 lights up yellow
...	...	...	
6	LED C4.1 yellow	1_{bin}	LED K4.1 lights up yellow
7	LED C4.2 yellow	1_{bin}	LED K4.2 lights up yellow

5.2.2 KL8524 - control and status bytes

Channel 1

Process data mode

Control byte 1 in process data mode

Control byte 1 (CB1) is located in the output image [▶ 67], and is transmitted from the controller to the terminal.

Bit	CB1.7	CB1.6	CB1.5	CB1.4	CB1.3	CB1.2	CB1.1	CB1.0
Name	RegAccess	-	-	-	-	-	-	-

Key

Bit	Name	Description	
CB1.7	RegAccess	0_{bin}	Register communication off (process data mode)
CB1.6 - CB1.0	-	0_{bin}	reserved

Status byte 1 in process data mode

The status byte 1 (SB1) is located in the input image [▶ 67] and is transmitted from terminal to the controller.

Bit	SB1.7	SB1.6	SB1.5	SB1.4	SB1.3	SB1.2	SB1.1	SB1.0
Name	RegAccess	-	-	-	-	-	-	-

Key

Bit	Name	Description	
SB1.7	RegAccess	0_{bin}	Acknowledgment for process data mode
SB1.6 - SB1.0	-	0_{bin}	reserved

Register communication

Control byte 1 in register communication

Control byte 1 (CB1) is located in the output image [▶ 67], and is transmitted from the controller to the terminal.

Bit	CB1.7	CB1.6	CB1.5	CB1.4	CB1.3	CB1.2	CB1.1	CB1.0
Name	RegAccess	R/W	Reg. no.					

Key

Bit	Name	Description	
CB1.7	RegAccess	1_{bin}	Register communication switched on
CB1.6	R/W	0_{bin}	Read access
		1_{bin}	Write access
CB1.5 to CB1.0	Reg. no.	Register number: Enter the number of the register that you - want to read with input data word DataIN1 [▶ 67] or - want to write with output data word DataOUT1 [▶ 67].	

Status byte 1 in register communication

The status byte 1 (SB1) is located in the input image [▶ 67] and is transmitted from terminal to the controller.

Bit	SB1.7	SB1.6	SB1.5	SB1.4	SB1.3	SB1.2	SB1.1	SB1.0
Name	RegAccess	R/W	Reg. no.					

Key

Bit	Name	Description	
SB1.7	RegAccess	1_{bin}	Acknowledgment for register access
SB1.6	R	0_{bin}	Read access
SB1.5 to SB1.0	Reg. no.	Number of the register that was read or written.	

Channels 2 to 8

The control and status bytes of channels 2 to 8 are structured like the control and status byte of channel 1.

5.2.3 KL8524 – Register overview

The registers are used for the parameterization of the Bus Terminals and are available for each channel. They can be read or written by means of the register communication [▶ 70].

Register no.	Comment	Default value		R/W	Memory
R0	reserved	-	-	-	-
...	...	...	...	...	...
R6	reserved	-	-	-	-
R7 [▶ 73]	Command register	0x0000	0_{dec}	R/W	RAM
R8 [▶ 73]	Terminal type	0x214C	8524_{dec}	R	ROM
R9 [▶ 73]	Firmware version (ASCI)	e.g. 0x3141	e.g. 12609_{dec}	R	ROM
R10	Multiplex shift register			R	ROM
R11	Signal channels			R	ROM
R12 [▶ 73]	minimum data length of a channel			R	ROM
R13	Data structure			R	ROM
R14	reserved	-	-	-	-
R15	Alignment register			R/W	RAM
R16 [▶ 73]	Hardware version number	e.g. 0x0000	e.g. 0_{dec}	R/W	SEEPROM
R17	reserved	-	-	-	-
...	...	...	...	...	...
R30	reserved	-	-	-	-
R31 [▶ 73]	Code word register	0x0000	0_{dec}	R/W	RAM
R32 [▶ 73]	Enable PLC Set LED	0x0000	0_{dec}	R/W	SEEPROM
R33 [▶ 74]	K-bus off reaction	0x0000	0_{dec}	R/W	SEEPROM
R34	reserved	-	-	-	-
...	reserved	-	-	-	-
R36	reserved	-	-	-	-
R37 [▶ 74]	Channel 1: output mode / switch-on delay	0x0000	0_{dec}	R/W	SEEPROM
R38 [▶ 76]	Channel 2: output mode / switch-on delay	0x0000	0_{dec}	R/W	SEEPROM
R39 [▶ 76]	Channel 3: output mode / switch-on delay	0x0000	0_{dec}	R/W	SEEPROM
R40 [▶ 76]	Channel 4: output mode / switch-on delay	0x0000	0_{dec}	R/W	SEEPROM
R41	reserved	-	-	-	-
...	reserved	-	-	-	-
R63	reserved	-	-	-	-

5.2.4 KL8524 – Register description

The registers are used for the parameterization of the Bus Terminals and are available for each channel. They can be read or written by means of the register communication.

R7: Command register

To execute a standard command, the user code word 0x1235 must be entered in register R31 [▶ 73].

Command 0x7000: Restore Factory Settings

Entering 0x7000 in register R7 restores the delivery state for the following registers.

- Register R32 [▶ 73]: 0x0000
- Register R33 [▶ 74]: 0x0000
- Register R34: 0x0000
- Register R35: 0x0000
- Register R36: 0x0000
- Register R37 [▶ 74]: 0x0000
- Register R38 [▶ 76]: 0x0000
- Register R39 [▶ 76]: 0x0000
- Register R40 [▶ 76]: 0x0000
  to
- Register R63: 0x0000

R8: Terminal description

The name of the terminal is contained in register R8.
KL8524: 0x214C (8524_{dec})

R9: Firmware version

Register R9 contains the ASCII coding of the terminal's firmware version, e.g. **0x3141 = '1A'.** The **'0x31'** corresponds here to the ASCII character **'1'**, while the **'0x41'** represents the ASCII character **'A'**.
This value cannot be changed.

R12: Minimum data length of a channel

R16: Hardware version number

Register R16 contains the hardware version of the terminal.

R31: Code word register

- If you write values into the user registers without first entering the user code word (0x1235) into the code word register, the terminal will not accept the supplied data.
- If you write values into the user registers and have previously entered the user code word (0x1235) in the code word register, these values are stored in the RAM registers and in the SEEPROM registers and are therefore retained when the terminal is restarted.

The code word is reset when the terminal is restarted.

R32: Enable PLC Set LED

Using this register you can allow the PLC to set the status LEDs of the individual channels.

Bit	R32.15	R32.14	R32.13	R32.12	R32.11	R32.10	R32.9	R32.8
Name	-	-	-	-	-	-	-	-

Bit	R32.7	R32.6	R32.5	R32.4	R32.3	R32.2	R32.1	R32.0
Name	-	-	-	-	enSetLED_Ch4	enSetLED_Ch3	enSetLED_Ch2	enSetLED_Ch1

Key

Bit	Name	Description		default
R32.15	-	reserved		0_{bin}
...	...	...		...
R32.4	-	reserved		0_{bin}
R32.3	enSetLED_Ch4	0_{bin}	LEDs of channel 4 cannot be set by the PLC	0_{bin}
		1_{bin}	LEDs of channel 4 can be set by the PLC	
...	...	...		...
R32.0	enSetLED_Ch1	0_{bin}	LEDs of channel 1 cannot be set by the PLC	0_{bin}
		1_{bin}	LEDs of channel 1 can be set by the PLC	

R33: Reaction to K-bus errors

In the event of a K-bus error all outputs that have been set via the PLC go out. Unless other instructions are set in R33.

Bit	R33.15	R33.14	R33.13	R33.12	R33.11	R33.10	R33.9	R33.8
Name	-	-	-	-	-	-	-	-

Bit	R33.7	R33.6	R33.5	R33.4	R33.3	R33.2	R33.1	R33.0
Name	KBOR_Ch8	KBOR_Ch7	KBOR_Ch6	KBOR_Ch5	KBOR_Ch4	KBOR_Ch3	KBOR_Ch2	KBOR_Ch1

Key

Bit	Name	Description		default
R33.15	-	reserved		0_{bin}
...	...	...		...
R33.8	-	reserved		0_{bin}
R33.7	K-bus-OFF-Reaction_Ch4.2	0_{bin}	In the event of a K-bus error output 4.2 is reset (0_{bin}).	0_{bin}
		1_{bin}	In event of a K-bus error output 4.2 is set (1_{bin}).	
R33.6	K-bus-OFF-Reaction_Ch4.1	0_{bin}	In the event of a K-bus error output 4.1 is reset (0_{bin}).	0_{bin}
		1_{bin}	In event of a K-bus error output 4.1 is set (1_{bin}).	
...	...	...		...
R33.1	K-bus-OFF-Reaction_Ch1.2	0_{bin}	In the event of a K-bus error output 1.2 is reset (0_{bin}).	0_{bin}
		1_{bin}	In event of a K-bus error output 1.2 is set (1_{bin}).	
R33.0	K-bus-OFF-Reaction_Ch1.1	0_{bin}	In the event of a K-bus error output 1.1 is reset (0_{bin}).	0_{bin}
		1_{bin}	In event of a K-bus error output 1.1 is set (1_{bin}).	

R37: Output mode / switch-on delay for channel 1

In register 37 you can define the output mode and the switch-on delay for channel 1.

Bit	R37.15	R37.14	R37.13	R37.12	R37.11	R37.10	R37.9	R37.8
Name	-	-	-	-	-	-	-	OutputMode

Bit	R37.7	R37.6	R37.5	R37.4	R37.3	R37.2	R37.1	R37.0
Name	TurnOnDelay							

Key

Bit	Name	Description		default
R37.15	-	reserved		0_{bin}
...	...	...		...
R37.9	-	reserved		0_{bin}
R37.8	OutputMode[1] (arMode)	0_{bin}	Channel 1 is operated in output mode 1	0_{bin}
		1_{bin}	Channel 1 is operated in output mode 2	
R37.7-R37.0	TurnOnDelay[2]	0_{dec}	No switch-on delay	0_{dec}
		1_{dec}	10 ms switch-on delay	
		2_{dec}	20 ms switch-on delay	
		...	...	
		255_{dec}	2550 ms switch-on delay	

[1]) OutputMode (arMode):

Output mode 1:

Switch settings	Output C 1.2	Output C 1.1
0	0	0
1	0	1
2	1	0

Output mode 2:

Switch settings	Output C 1.2	Output C 1.1
0	0	0
1	0	1
2	1	1

[2]) Switchover delay: A delay time can be specified, after which the switch position becomes effective.

Delay time in output mode 1

Output mode 1 can be used if the two outputs may not be switched on at the same time (e.g. fan motor with star-delta switching).

The delay time starts in output mode 1 when changing from

- switch position 1 to switch position 2
- switch position 2 to switch position 1
- Fast switching from 0 to 2 or from 2 to 0 always takes place via switch position 1 and thus also starts the delay time.

Examples:

- Output 1 switches on immediately when the three-stage switch is switched from 0 to 1.
- The delay time starts when the three-stage switch is switched from 1 to 2. No output is switched during the delay time. Output 2 is switched on when the delay time expires.
- The delay time starts when the three-stage switch is switched back from 2 to 1. No output is switched during the delay time. Output 1 is switched on when the delay time expires.
- Switching from 1 or 2 to 0 takes place immediately. Both outputs are switched off immediately.
- If the switch is switched back to the previous switch position during the delay time, the previous output immediately becomes active again.

Delay time in output mode 2

Output mode 2 can be used if there has to be a minimum time before output 2 is switched on (e.g. consumers with two stages that would have an excessively high starting current if switched on directly in stage 2).

The delay time starts output mode 2 when changing from

- switch position 1 to switch position 2
- Fast switching from 0 to 2 always takes place via switch position 1 and therefore also starts the delay time.

Examples:

- Output 1 switches on immediately when the three-stage switch is switched from 0 to 1.
- The delay time starts when the three-stage switch is switched from 1 to 2. Output 1 remains switched on as long as the delay time is running. Output 2 is additionally switched on when the delay time expires.
- The switching back of the three-stage switch from 2 to 1 takes place immediately. Output 2 switches off, output 1 remains switched on.
- Switching from 1 or 2 to 0 takes place immediately. Both outputs are switched off immediately.
- The delay time starts again if the switch is switched from 1 to 2 when the delay time is running.

R38: Output mode / switch-on delay for channel 2

See register 37.

R39: Output mode / switch-on delay for channel 3

See register 37.

R40: Output mode / switch-on delay for channel 4

See register 37.

5.3 KL8528

5.3.1 KL8528 – Process image

The KL8528 is represented in the process image with 6 bytes each of input and output data. These are organized as follows:

Byte offset	Format	Input data	Output data
0	Byte	Status byte (SB [▶ 78]), for register communication only	Control byte (CB [▶ 78]), for register communication only
1	Byte	Data byte 1 (auto/manual for output 1 to 8)	Data byte 1 (sets output 1 to 8)*
2	Byte	Data byte 2 (switch position for output 1 to 8)	Data byte 2 (reserved)
3	Byte	reserved	Data byte 3 (sets LED 1 to 8 green)
4	Byte	reserved	Data byte 4 (sets LED 1 to 8 yellow)
5	Byte	reserved	reserved

*) If the respective switch is set to auto.

Control and status byte

See chapter Control and status byte [▶ 78]

Process data

Input data

Data byte 1

Data byte 1 indicates the positions of the switches (auto/man).

Bit	Name	Description	
0	enAutoCh1	1_{bin}	Switch of channel 1 is set to auto: the PLC controls this channel.
1	enAutoCh2	1_{bin}	Switch of channel 2 is set to auto: the PLC controls this channel.
...	...	...	
7	enAutoCh8	1_{bin}	Switch of channel 8 is set to auto: the PLC controls this channel.

Data byte 2

Data byte 2 indicates the positions of the switches (off/on).

Bit	Name	Description	
0	Switch1	0_{bin}	Switch of channel 1 is set to off
		1_{bin}	Switch of channel 1 is set to on
1	Switch1	0_{bin}	Switch of channel 2 is set to off
		1_{bin}	Switch of channel 2 is set to on
...	...	...	
7	Switch1	0_{bin}	Switch of channel 8 is set to off
		1_{bin}	Switch of channel 8 is set to on

Output data

Data byte 1

Data byte 1 sets the outputs in automatic mode.

Bit	Name	Description	
0	setCh1	1_{bin}	Output 1 is set.
1	setCh2	1_{bin}	Output 2 is set.
...	...	...	
7	setCh8	1_{bin}	Output 8 is set.

Data byte 2

Has no function (reserved) in process data mode, but is used for register communication.

Data byte 3

Data byte 3 sets the LEDs of the channels to green.

Bit	Name	Description	
0	LED 1 green	1_{bin}	LED 1 lights up green
1	LED 2 green	1_{bin}	LED 2 lights up green
...	...	...	
7	LED 8 green	1_{bin}	LED 8 lights up green

Data byte 4

Data byte 4 sets the LEDs of the channels to yellow.

Bit	Name	Description	
0	LED 1 green	1_{bin}	LED 1 lights up yellow
1	LED 2 green	1_{bin}	LED 2 lights up yellow
...	...	...	
7	LED 8 green	1_{bin}	LED 8 lights up yellow

5.3.2 KL8528 - control and status bytes

Channel 1

Process data mode

Control byte 1 in process data mode

Control byte 1 (CB1) is located in the output image [▶ 76], and is transmitted from the controller to the terminal.

Bit	CB1.7	CB1.6	CB1.5	CB1.4	CB1.3	CB1.2	CB1.1	CB1.0
Name	RegAccess	-	-	-	-	-	-	-

Key

Bit	Name	Description	
CB1.7	RegAccess	0_{bin}	Register communication off (process data mode)
CB1.6 - CB1.0	-	0_{bin}	reserved

Status byte 1 in process data mode

The status byte 1 (SB1) is located in the input image [▶ 76] and is transmitted from terminal to the controller.

Bit	SB1.7	SB1.6	SB1.5	SB1.4	SB1.3	SB1.2	SB1.1	SB1.0
Name	RegAccess	-	-	-	-	-	-	-

Key

Bit	Name	Description	
SB1.7	RegAccess	0_{bin}	Acknowledgment for process data mode
SB1.6 - SB1.0	-	0_{bin}	reserved

Register communication

Control byte 1 in register communication

Control byte 1 (CB1) is located in the output image [▶ 76], and is transmitted from the controller to the terminal.

Bit	CB1.7	CB1.6	CB1.5	CB1.4	CB1.3	CB1.2	CB1.1	CB1.0
Name	RegAccess	R/W	Reg. no.					

Key

Bit	Name	Description	
CB1.7	RegAccess	1_{bin}	Register communication switched on
CB1.6	R/W	0_{bin}	Read access
		1_{bin}	Write access
CB1.5 to CB1.0	Reg. no.	Register number: Enter the number of the register that you - want to read with input data word DataIN1 [▶ 76] or - want to write with output data word DataOUT1 [▶ 76].	

Status byte 1 in register communication

The status byte 1 (SB1) is located in the input image [▶ 76] and is transmitted from terminal to the controller.

Bit	SB1.7	SB1.6	SB1.5	SB1.4	SB1.3	SB1.2	SB1.1	SB1.0
Name	RegAccess	R/W	Reg. no.					

Key

Bit	Name	Description	
SB1.7	RegAccess	1_{bin}	Acknowledgment for register access
SB1.6	R	0_{bin}	Read access
SB1.5 to SB1.0	Reg. no.	Number of the register that was read or written.	

Channels 2 to 8

The control and status bytes of channels 2 to 8 are structured like the control and status byte of channel 1.

5.3.3 KL8528 – Register overview

The registers are used for the parameterization of the Bus Terminals and are available for each channel. They can be read or written by means of the register communication [▶ 78].

Register no.	Comment	Default value		R/W	Memory
R0	reserved	-	-	-	-
...	...	...	...	...	...
R6	reserved	-	-	-	-
R7 [▶ 80]	Command register	0x0000	0_{dec}	R/W	RAM
R8 [▶ 80]	Terminal type	0x2150	8528_{dec}	R	ROM
R9 [▶ 81]	Firmware version (ASCI)	e.g. 0x2150	e.g. 12609_{dec}	R	ROM
R10	Multiplex shift register			R	ROM
R11	Signal channels			R	ROM
R12 [▶ 81]	minimum data length of a channel			R	ROM
R13	Data structure			R	ROM
R14	reserved	-	-	-	-
R15	Alignment register			R/W	RAM
R16 [▶ 81]	Hardware version number	e.g. 0x0000	e.g. 0_{dec}	R/W	SEEPROM
R17	reserved	-	-	-	-
...	...	...	...	...	...
R30	reserved	-	-	-	-
R31 [▶ 81]	Code word register	0x0000	0_{dec}	R/W	RAM
R32 [▶ 81]	Disable Channel LED	0x0000	0_{dec}	R/W	SEEPROM
R33 [▶ 81]	K-bus off reaction	0x0000	0_{dec}	R/W	SEEPROM
R34	reserved	-	-	-	-
...	reserved	-	-	-	-
R63	reserved	-	-	-	-

5.3.4 KL8528 – Register description

The registers are used for the parameterization of the Bus Terminals and are available for each channel. They can be read or written by means of the register communication.

R7: Command register

To execute a standard command, the user code word 0x1235 must be entered in register R31 [▶ 81].

Command 0x7000: Restore Factory Settings

Entering 0x7000 in register R7 restores the delivery state for the following registers.

- Register R32 [▶ 81]: 0x0000
- Register R33 [▶ 81]: 0x0000
  to
- Register R63: 0x0000

R8: Terminal description

The name of the terminal is contained in register R8.
KL8528: 0x2150 (8528_{dec})

R9: Firmware version

Register R9 contains the ASCII coding of the terminal's firmware version, e.g. **0x3141 = '1A'.** The **'0x31'** corresponds here to the ASCII character **'1'**, while the **'0x41'** represents the ASCII character **'A'**.
This value cannot be changed.

R12: Minimum data length of a channel

R16: Hardware version number

Register R16 contains the hardware version of the terminal.

R31: Code word register

- If you write values into the user registers without first entering the user code word (0x1235) into the code word register, the terminal will not accept the supplied data.
- If you write values into the user registers and have previously entered the user code word (0x1235) in the code word register, these values are stored in the RAM registers and in the SEEPROM registers and are therefore retained when the terminal is restarted.

The code word is reset when the terminal is restarted.

R32: Disable Channel LED

Using this register you can deactivate the status LEDs of the individual channels.

Bit	R32.15	R32.14	R32.13	R32.12	R32.11	R32.10	R32.9	R32.8
Name	-	-	-	-	-	-	-	-

Bit	R32.7	R32.6	R32.5	R32.4	R32.3	R32.2	R32.1	R32.0
Name	disLED_Ch8	disLED_Ch7	disLED_Ch6	disLED_Ch5	disLED_Ch4	disLED_Ch3	disLED_Ch2	disLED_Ch1

Key

Bit	Name	Description		default
R32.15	-	reserved		0_{bin}
...	...	...		...
R32.8	-	reserved		0_{bin}
R32.7	disLED_Ch8	0_{bin}	The LED indicates the state of output 8. In the event of a K-bus error the LED flashes.	0_{bin}
		1_{bin}	The LED does not indicate the state of output 8. The PLC can now switch the LED selectively to green or red. In the event of a K-bus error the LED turns off.	
R32.6	disLED_Ch7	0_{bin}	The LED indicates the state of output 7. In the event of a K-bus error the LED flashes.	0_{bin}
		1_{bin}	The LED does not indicate the state of output 7. The PLC can now switch the LED selectively to green or red. In the event of a K-bus error the LED turns off.	
...	...	...		...
R32.0	disLED_Ch1	0_{bin}	The LED indicates the state of output 1. In the event of a K-bus error the LED flashes.	0_{bin}
		1_{bin}	The LED does not indicate the state of output 1. The PLC can now switch the LED selectively to green or red. In the event of a K-bus error the LED turns off.	

R33: Reaction to K-bus errors

In the event of a K-bus error all outputs that have been set via the PLC go out. Unless other instructions are set in R33.

Bit	R33.15	R33.14	R33.13	R33.12	R33.11	R33.10	R33.9	R33.8
Name	-	-	-	-	-	-	-	-

Bit	R33.7	R33.6	R33.5	R33.4	R33.3	R33.2	R33.1	R33.0
Name	KBOR_Ch8	KBOR_Ch7	KBOR_Ch6	KBOR_Ch5	KBOR_Ch4	KBOR_Ch3	KBOR_Ch2	KBOR_Ch1

Key

Bit	Name	Description		default
R33.15	-	reserved		0_{bin}
...	...	...		...
R33.8	-	reserved		0_{bin}
R33.7	K-bus-OFF-Reaction_Ch8	0_{bin}	In the event of a K-bus error output 8 is reset (0_{bin}).	0_{bin}
		1_{bin}	In event of a K-bus error output 8 is set (1_{bin}).	
R33.6	K-Bus-OFF-Reaction_Ch7	0_{bin}	In the event of a K-bus error output 7 is reset (0_{bin}).	0_{bin}
		1_{bin}	In event of a K-bus error output 7 is set (1_{bin}).	
...	...	...		...
R33.0	K-bus-OFF-Reaction_Ch1	0_{bin}	In the event of a K-bus error output 1 is reset (0_{bin}).	0_{bin}
		1_{bin}	In event of a K-bus error output 1 is set (1_{bin}).	

5.4 KL8548

5.4.1 KL8548 – Process image

The KL8548 is represented in the process image with 24 bytes each of input and output data. These are organized as follows:

Byte offset (without word alignment)	Byte offset (with word alignment*)	Format	Input data	Output data
0	0	Byte	SB1 [▶ 83]	CB1 [▶ 83]
1	2	Word	DataIN1	DataOUT1
3	4	Byte	SB2 [▶ 83]	CB2 [▶ 83]
4	6	Word	DataIN2	DataOUT2
6	8	Byte	SB3 [▶ 83]	CB3 [▶ 83]
7	10	Word	DataIN3	DataOUT3
9	12	Byte	SB4 [▶ 83]	CB5 [▶ 83]
10	14	Word	DataIN4	DataOUT4
12	16	Byte	SB1 [▶ 83]	CB1 [▶ 83]
13	18	Word	DataIN5	DataOUT5
15	20	Byte	SB2 [▶ 83]	CB2 [▶ 83]
16	22	Word	DataIN6	DataOUT6
18	24	Byte	SB3 [▶ 83]	CB3 [▶ 83]
19	26	Word	DataIN7	DataOUT7
21	28	Byte	SB4 [▶ 83]	CB5 [▶ 83]
22	30	Word	DataIN8	DataOUT8

*) Word alignment: The Bus Coupler places values on even byte addresses

Key

SB1 to SB8: status bytes of channels 1 to 8
CB1 to CB8: control bytes of channels 1 to 8
DataIN1 to DataIN8: input words of channels 1 to 8
DataOUT1 to DataOUT8: output words of channels 1 to 8

Control and status byte

See chapter Control and status byte [▶ 83]

Process data

Input data (DataIN1 to DataIN8)

Position of the potentiometers with user scaling switched off (R32.0 [▶ 81] = 0)

Potentiometer position	Input value in the PLC	
	hexadecimal	decimal
CCW stop	0x0000	0
Center	0x3FFF	16383
CW stop	0x7FFF	32767

Output data (DataOUT1 to DataOUT8)

Specification of the output values by the controller.

Output value		Output voltage
hexadecimal	decimal	
0x0000	0	0 V
0x3FFF	16383	+5 V
0x7FFF	32767	+10 V

5.4.2 KL8548 - control and status bytes

Channel 1

Process data mode

Control byte 1 in process data mode

Control byte 1 (CB1) is located in the output image [▶ 82], and is transmitted from the controller to the terminal.

Bit	CB1.7	CB1.6	CB1.5	CB1.4	CB1.3	CB1.2	CB1.1	CB1.0
Name	RegAccess	-	-	-	-	-	-	-

Key

Bit	Name	Description	
CB1.7	RegAccess	0_{bin}	Register communication off (process data mode)
CB1.6 - CB1.0	-	0_{bin}	reserved

Status byte 1 in process data mode

The status byte 1 (SB1) is located in the input image [▶ 82] and is transmitted from terminal to the controller.

Bit	SB1.7	SB1.6	SB1.5	SB1.4	SB1.3	SB1.2	SB1.1	SB1.0
Name	RegAccess	-	-	-	-	-	-	auto/man

Key

Bit	Name	Description	
SB1.7	RegAccess	0_{bin}	Acknowledgment for process data mode
SB1.6 - SB1.1	-	0_{bin}	reserved
SB1.0	auto/man	0_{bin}	The auto/man switch of channel 1 is set to *auto*. The PLC controls the output.
		1_{bin}	The auto/man switch of channel 1 is set to *manual*. The potentiometer controls the output.

Register communication

Control byte 1 in register communication

Control byte 1 (CB1) is located in the output image [▶ 82], and is transmitted from the controller to the terminal.

Bit	CB1.7	CB1.6	CB1.5	CB1.4	CB1.3	CB1.2	CB1.1	CB1.0
Name	RegAccess	R/W	Reg. no.					

Key

Bit	Name	Description	
CB1.7	RegAccess	1_{bin}	Register communication switched on
CB1.6	R/W	0_{bin}	Read access
		1_{bin}	Write access
CB1.5 to CB1.0	Reg. no.	Register number: Enter the number of the register that you - want to read with input data word DataIN1 [▶ 82] or - want to write with output data word DataOUT1 [▶ 82].	

Status byte 1 in register communication

The status byte 1 (SB1) is located in the input image [▶ 82] and is transmitted from terminal to the controller.

Bit	SB1.7	SB1.6	SB1.5	SB1.4	SB1.3	SB1.2	SB1.1	SB1.0
Name	RegAccess	R/W	Reg. no.					

Key

Bit	Name	Description	
SB1.7	RegAccess	1_{bin}	Acknowledgment for register access
SB1.6	R	0_{bin}	Read access
SB1.5 to SB1.0	Reg. no.	Number of the register that was read or written.	

Channels 2 to 8

The control and status bytes of channels 2 to 8 are structured like the control and status byte of channel 1.

5.4.3 KL8548 – Register overview

Register Overview

The registers are used for the parameterization of the Bus Terminals and are available for each channel. They can be read or written by means of the register communication [▶ 84].

Register no.	Comment	Default value		R/W	Memory
R0	reserved	-	-	-	-
...	...	...	...	...	...
R6	reserved	-	-	-	-
R7 [▶ 85]	Command register	0x0000	0_{dec}	R/W	RAM
R8 [▶ 85]	Terminal type	0x2164	8548_{dec}	R	ROM
R9 [▶ 85]	Firmware version (ASCI)	e.g. 0x3141	e.g. 12609_{dec}	R	ROM
R10	Multiplex shift register			R	ROM
R11	Signal channels			R	ROM
R12 [▶ 86]	minimum data length of a channel			R	ROM
R13	Data structure			R	ROM
R14	reserved	-	-	-	-
R15	Alignment register			R/W	RAM
R16 [▶ 86]	Hardware version number	e.g. 0x0000	e.g. 0_{dec}	R/W	SEEPROM
R17	reserved	-	-	-	-
...	...	...	...	...	...
R30	reserved	-	-	-	-
R31 [▶ 86]	Code word register	0x0000	0_{dec}	R/W	RAM
R32 [▶ 86]	Feature register	0x0004	4_{dec}	R/W	SEEPROM
R33 [▶ 86]	User scaling: offset	0x0000	0_{dec}	R/W	SEEPROM
R34	User scaling: gain	0x0100	256_{dec}	R/W	SEEPROM
R35	User switch-on value	0x0000	0_{dec}	R/W	SEEPROM
R36	reserved	-	-	-	-
...	reserved	-	-	-	-
R63	reserved	-	-	-	-

5.4.4 KL8548 – Register description

The registers are used for the parameterization of the Bus Terminals and are available for each channel. They can be read or written by means of the register communication.

R7: Command register

Not used.

R8: Terminal description

The name of the terminal is contained in register R8.
KL8548: 0x2164 ($KL8548_{dec}$)

R9: Firmware version

Register R9 contains the ASCII coding of the terminal's firmware version, e.g. **0x3141 = '1A'.** The **'0x31'** corresponds here to the ASCII character **'1'**, while the **'0x41'** represents the ASCII character **'A'**.
This value cannot be changed.

R12: Minimum data length of a channel

R16: Hardware version number

Register R16 contains the hardware version of the terminal.

R31: Code word register

- If you write values into the user registers without first entering the user code word (0x1235) into the code word register, the terminal will not accept the supplied data.
- If you write values into the user registers and have previously entered the user code word (0x1235) in the code word register, these values are stored in the RAM registers and in the SEEPROM registers and are therefore retained when the terminal is restarted.

The code word is reset when the terminal is restarted.

R32: Feature register

With this register you can switch various features for this channel on or off.

Bit	R32.15	R32.14	R32.13	R32.12	R32.11	R32.10	R32.9	R32.8
Name	-	-	-	-	-	-	-	enUserValue

Bit	R32.7	R32.6	R32.5	R32.4	R32.3	R32.2	R32.1	R32.0
Name	-	-	-	enEcoMode	-	enWatchdog	-	-

Key

Bit	Name	Description		default
R32.15	-	reserved		0_{bin}
...	...	...		...
R32.9	-	reserved		0_{bin}
R32.8	enUserValue	0_{bin}	User switch-on value activated	0_{bin}
		1_{bin}	User switch-on value deactivated	
R32.7	-	reserved		0_{bin}
...	...	...		...
R32.5	-	reserved		0_{bin}
R32.4	enEcoMode	0_{bin}	Bar graph activated	0_{bin}
		1_{bin}	Eco Mode activated	
R32.3	...	reserved		...
R32.2	enWatchdog	0_{bin}	Watchdog deactivated	1_{bin}
		1_{bin}	Watchdog activated	
R32.1	-	reserved		0_{bin}
R32.0	-	reserved		0_{bin}

R33: User switch-on value

If the user switch-on value has been enabled with bit R32.8 [▶ 86] of the feature register, the terminal sets its output to the user switch-on value in place of the manufacturer switch-on value on the occurrence of a system reset or a watchdog timer overflow (terminal has not received any process data for 100 ms).

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

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 (response from 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!

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