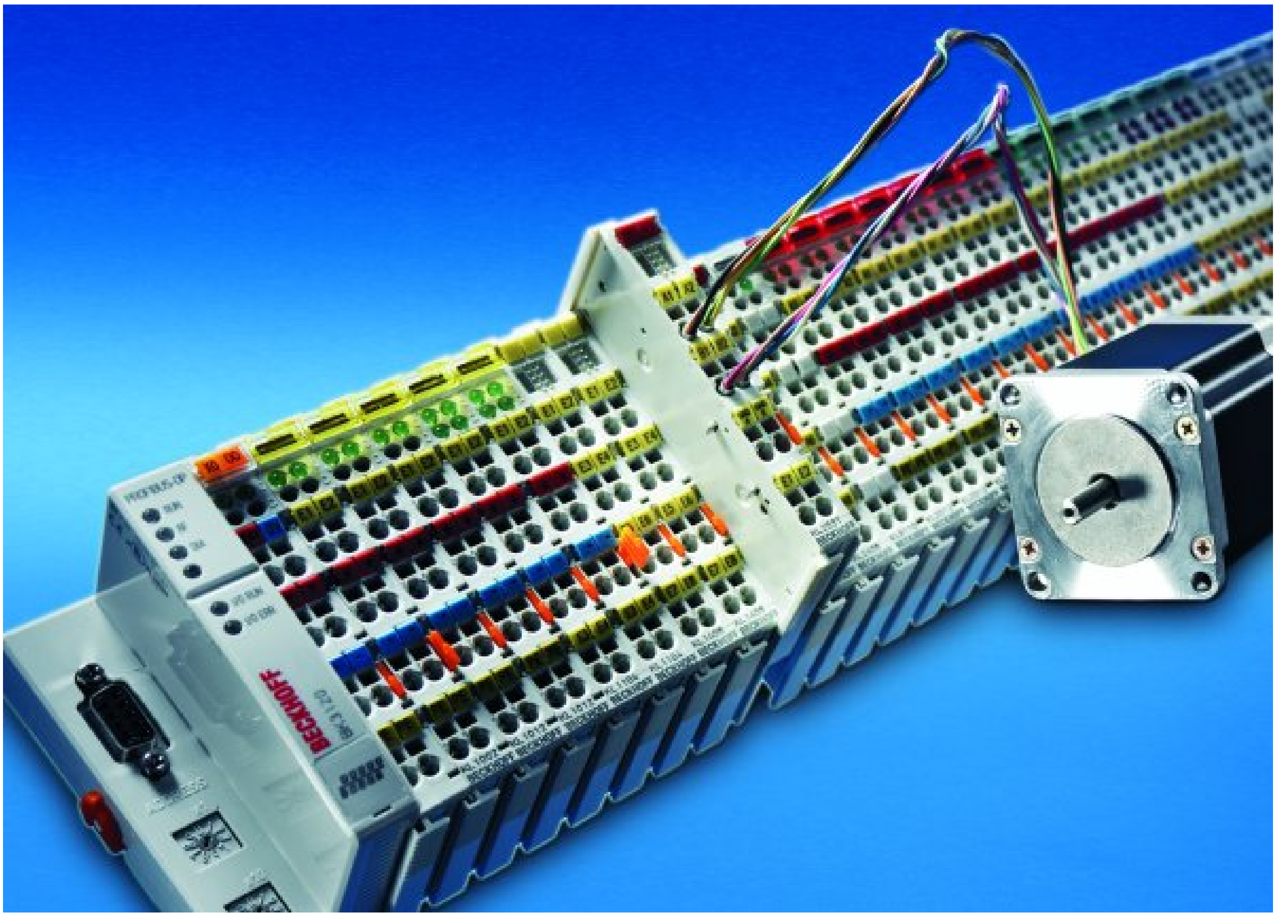


Documentation

KL2531/KS2531, KL2541/KS2541

One channel stepper motor terminals

Version: 2.1.0
Date: 2017-12-07

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.
ℹ **Note**	**Tip or pointer** This symbol indicates information that contributes to better understanding.

1.3 Documentation issue status

Version	Comment
2.1.0	• Introduction for KL2541 updated • Chapter *Switching behavior* for KL2541 added • Technical data updates • KL2541-0006 added
2.0.0	• Migration • Technical data updated • Chapter "Installation instructions for enhanced mechanical load capacity" added • Update Chapter *Connection* • Chapter *Prescribed installation position* replaced with chapter *Installation position with or without fan* • Revision status updated • Chapter *UL notice - Compact Motion* added • Update structure
1.8.0	• Technical data updated • Information on basic function principles expanded • Register description expanded
1.7.0	• Mounting instructions extended • Technical data updated • Description of temperature monitoring expanded • Colours of conductor in the connection examples adapted to the colours of motor cable ZK4000-6200-2xxx for Beckhoff stepper motors (AS10x0)
1.6.3	• Description of the operating modes expanded • Register description expanded • Description of the KS2000 configuration software expanded
1.6.2	• Description of the process image updated
1.6.1	• Description of the process image expanded • Basic function principles updated • Description of the operating modes expanded
1.6.0	• Information on basic function principles expanded • Description of the operating modes expanded • Introductions and connection instructions updated • Technical data updated • Description of control and status byte updated • Register description expanded • Mounting description expanded
1.5.0	• Description of the basic function principles expanded • Description of the operating modes expanded • Register description expanded • Description of the KS2000 configuration software expanded • Technical data updated
1.4.1	Minor routine corrections (typing errors, spelling, etc.)
1.4	(only published in German) • Description of the basic function principles expanded • Description of the operating modes expanded • Connection examples updated • Register description expanded • Description of control and status byte extended • Description of control and status word expanded • Description of the KS2000 configuration software expanded • Various examples for terminal handling added
1.3	• Connection examples for KL2531 revised • Connection examples for KL2541 added

Version	Comment
1.2	• Description of the power contacts the KL2541 expanded • Description of the command register expanded
1.1	• Formula for calculating the register values simplified
1.0	• Description of the basic function principles expanded • Description of the operating modes expanded • Example for motion command mode added • Description of the KS2000 configuration software expanded
0.2	• KL2541 added • Register description updated and register pages added • Further operating modes added: • Description of the KS2000 configuration software expanded
0.1	• First provisional documentation for KL2531 • Describes only *velocity* mode

Firmware and hardware versions

Documentation Version	KL2531-0000, KS2531-0000		KL2541-0000, KS2541-0000		KL2541-0006	
	Firmware	Hardware	Firmware	Hardware	Firmware	Hardware
2.1.0	4A	06	6B	17	6B	17
2.0.0	4A	06	6B	17	-	-
1.8.0	4A	04	4K	13		
1.7.0	3E	02	4A	10		
1.6.3	3E	02	3E	08		
1.6.2	3D	02	3D	07		
1.6.1	3D	02	3D	07		
1.6.0	3D	02	3D	07		
1.5.0	3A	02	3A	06		
1.4.1	2C	01	2D	02		
1.4	2C	01	2D	02		
1.3	2B	01	2C	02		
1.2	2B	01	2C	02		
1.1	2B	01	2C	02		
1.0	2B	01	2C	02		
0.2	2B	01	2C	01		
0.1	1B	01	-	-		

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

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.: 12 06 3A 02:

12 - week of production 12
06 - year of production 2006
3A - firmware version 3A
02 - hardware version 02

2 Product overview

2.1 KL2531 - Introduction

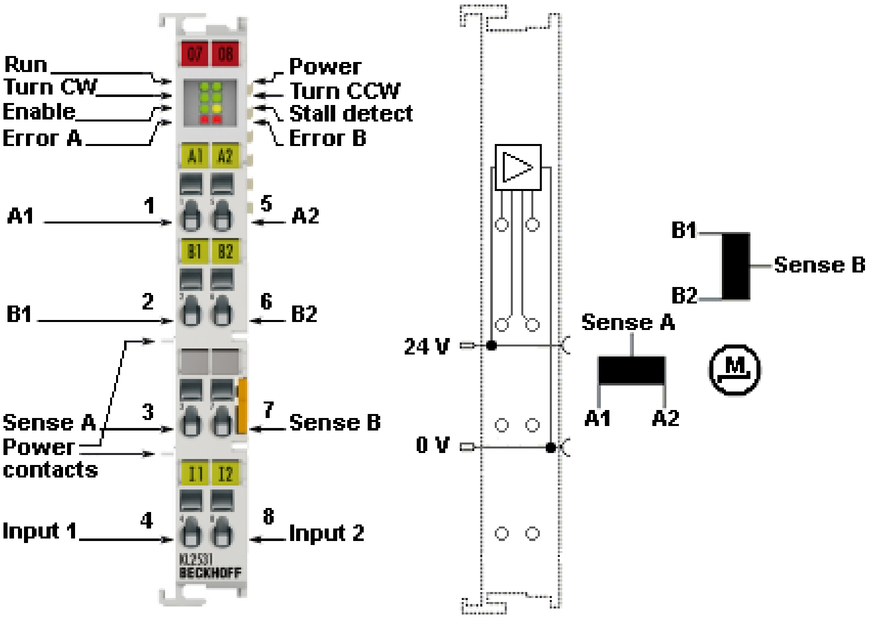


Fig. 1: KL2531

The KL2531 Bus Terminal is intended for the direct connection of different small stepper motors.

The slimline PWM output stages for two motor coils are located in the Bus Terminal together with two digital inputs (24 V) for limit switches. The KL2531 can be adjusted to the motor and the application by changing just a few parameters.

64-fold microstepping ensures particularly quiet and precise motor operation. In many applications, integrated monitoring of the mechanical load makes an encoder system or limit switch unnecessary.

The Bus Terminal is available in the KL2531 version for standard wiring [▸ 41] and the KS2531 for permanent wiring. [▸ 42]

2.2 KL2531 - Technical Data

Technical data	KL2531-0000, KS2531-0000
Number of digital outputs	1, for a stepper motor (2 phases)
Power supply for output stage (via power contacts)	8 V_{DC} to 24 V_{DC}
Output current	max. 1.5 A (overload- and short-circuit-proof)
Current controller frequency	approx. 25 kHz
Maximum step frequency	125 kHz
Step pattern	Full step, half step, 4 / 8 / 16 / 32 / 64-fold microstepping
Resolution	max. 64 steps, microstepping
Resolution per revolution	approx. 5000 positions in typical applications
Diagnostics LED for	Error segment A, error segment B, step loss/standstill, power, enable, direction of rotation, digital inputs
Number of digital inputs	2
Rated voltage	24 V_{DC} (-15%/+20%)
Signal voltage "0"	-3 V ... 5 V
Signal voltage "1"	15 V ... 30 V
Input filter	0.2 ms
Input current	typically 5 mA
Electrical isolation	500 V (K-Bus/signal voltage)
Power supply for the electronics	via the K-bus
Current consumption from K-bus	typically 60 mA
Bit width in the input process image	1 x 8 bit status, 2 x 16 bit data
Bit width in the output process image	1 x 8 bit control, 2 x 16 bit data
Pluggable wiring	for all KSxxxx Bus Terminals
Dimensions (W x H x D)	approx. 15 mm x 100 mm x 70 mm (width aligned: 12 mm)
Weight	approx. 55 g
Permissible ambient temperature range during operation	0 °C ... + 55°C
Permissible ambient temperature range during storage	-25 °C ... + 85°C
Permissible relative air humidity	95%, no condensation
Mounting [▶ 35]	on 35 mm mounting rail conforming to EN 60715
Vibration / shock resistance	conforms to EN 60068-2-6/EN 60068-2-2-7, see also Installation instructions for enhanced mechanical load capacity [▶ 37]
EMC immunity/emission	according to EN 61000-6-2 / EN 61000-6-4 / EN 61800-6-3* *) standard: category C3. Additional filters are required for categories C1 and C2!
Protection class	IP20
Permissible installation position	without fan cartridge ZB8610: standard installing position with fan cartridge ZB8610: standard installing position, other installing positions (example 1 & 2), see notice [▶ 38]
Approval	CE, cULus [▶ 44]

2.3 KL2531 - LED Displays

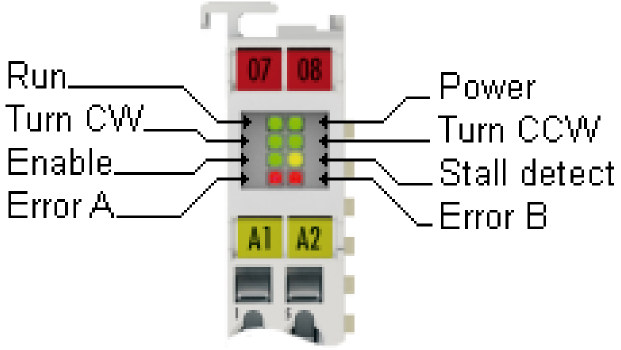


Fig. 2: KL2531 LEDs

LED	Display	
Run (green)	on	Data transmission on the K-bus is active
Turn CW (green)	on	Motor turns clockwise
Enable (green)	off	Motor control is disabled (bit SB.0 [▸ 74] is not set in the status byte) or KL2531 is not ready to operate
	on	Motor control is disabled (bit SB.0 [▸ 74] is set in the status byte) and KL2531 is ready to operate
Error A (red)	flashes (1 Hz)	Configuration error (bit SW.15 [▸ 77] is set in the status word): • Value for min. velocity v_{min} is 0 (R38 [▸ 85]) • Value for max. velocity v_{max} is 0 (R39 [▸ 85]) • Value for max. acceleration a_{max} is 0 (R40 [▸ 85]) • Value for coil current I_S with $a > a_{th}$ is 0 (R42 [▸ 85]) • Value for coil current I_S with $a \leq a_{th}$ is 0 (R43 [▸ 85])
Power (green)	off	Supply voltage ($24V_{DC}$) not available or Motor control is disabled (bit SB.0 [▸ 74] is not set in the status byte)
	on	Supply voltage ($24V_{DC}$) is available
Turn CCW (green)	on	Motor turns counterclockwise
Stall detect (yellow)	on	Motor control is disabled (bit SB.0 [▸ 74] is not set in the status byte)
	flashes (1 Hz)	The current load angle [▸ 18] (SB.3 to SB.1 [▸ 74]) is greater than the configured load angle threshold (R47 [▸ 85]).
Error B (red)	flashes (1 Hz)	Supply or load error Supply voltage (24 V_{DC}) too low (bit SW.9 [▸ 77] in the status word is set) • Open load at bridge A (bit SW.11 [▸ 77] is set in the status word) • Open load at bridge B (bit SW.12 [▸ 77] is set in the status word) • Overcurrent at jumper A (bit SW.13 [▸ 77] is set in the status word) • Overcurrent at jumper B (bit SW.11 [▸ 77] in the status word is set)

2.4 KL2541 - Introduction

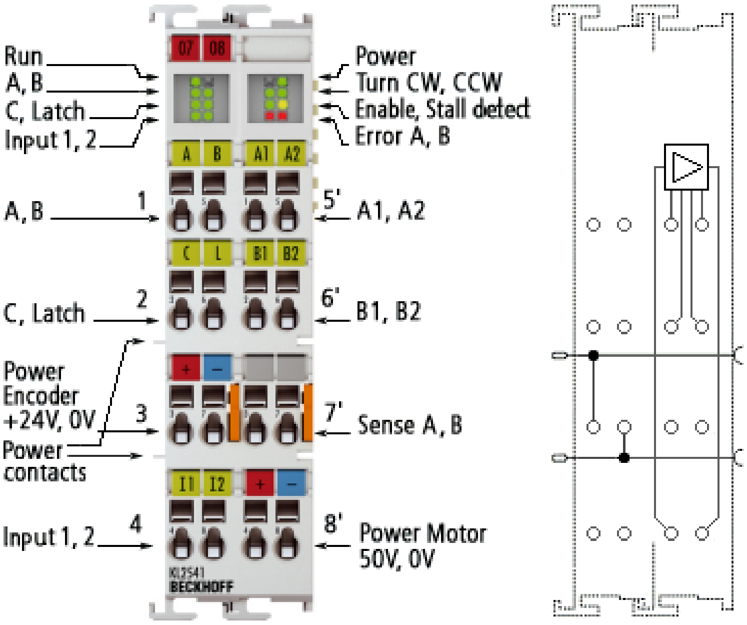


Fig. 3: KL2541-0000

The KL2541 Bus Terminal is intended for stepper motors with medium performance range.

The PWM output stages cover a wide range of voltages and currents. Together with two digital inputs (5 V / 24 V) for limit switches and an incremental encoder interface (5 V / 24 V), they are located in the Bus Terminal. The incremental encoder interface as well as both digital inputs are also usable for 5 V signal level. The KL2541 can be adjusted to the motor and the application by changing just a few parameters.

64-fold microstepping ensures particularly quiet and precise motor operation. Together with a stepper motor, the KL2541 represents an inexpensive small servo axis.

The Bus Terminal is available in the KL2541-0000 version for standard wiring [▸ 41] and the KS2541-0000 for pluggable wiring. [▸ 42]

Moreover, a variation with 5 V encoder supply is available with KL2541-0006.

2.5 KL2541 - Switching behavior

The KL2541 stepper motor terminal is optimized for the fastest signal transmission from the input to the evaluation unit.

The duration, which requires a signal change as a rising or falling edge from the terminal point on the terminal front side to the logic of the central processing unit, is specified for the KL2541 at TON and TOFF < 10 µs, for rising (TON) as well as for falling edge (TOFF). Due to this low absolute cycle time, the temperature drift of the cycle time is very low.

Since additional noise filters at the input would significantly increase the signal propagation delay in the KL2541, rapid level changes or pulses in the µs range arrive at the evaluation unit unfiltered and undamped.

In order to avoid external interference, we strongly recommend to connect the signal inputs of the KL2541 via shielded cables.

The sensor / signal generator must be able to generate a sufficiently steep signal edge. The power supply unit should have sufficient reserves / buffers to allow the signal to reach the terminal with sufficient edge steepness despite capacitive / inductive losses on the supply line.

2.6 KL2541 - Technical data

Technical data	KL2541-0000, KS2541-0000	KL2541-0006
Number of digital outputs	1, for a stepper motor (2 phases)	
Power supply for output stage (via terminal points 4' and 8')	8 V_{DC} to 50 V_{DC}	
Output current	max. 5 A (overload- and short-circuit-proof)	
Maximum step frequency	125 kHz	
Step pattern	Full step, half step, 4 / 8 / 16 / 32 / 64-fold microstepping	
Resolution	max. 64 steps, microstepping	
Resolution per revolution	approx. 5000 positions in typical applications	
Current controller frequency	approx. 25 kHz	
Number of encoder inputs	4, for an encoder system	
Encoder voltage "0"	-3 V ... 2 V	
Encoder voltage "1"	2,5 V ... 30 V	
Encoder supply	24 V_{DC}	5 V_{DC}
Pulse frequency	max. 400,000 increments/s (4-fold evaluation)	
Number of digital inputs	2, for end position	
Rated voltage	24 V_{DC} (-15%/+20%)	
Signal voltage "0"	-3 V ... 2 V	
Signal voltage "1"	2.5 V ... 30 V	
Input filter	0.2 ms	
Input current	typically 5 mA	
Diagnostics LED for	Error segment A, error segment B, step loss/standstill, power, enable, direction of rotation, encoder signals, digital inputs	
Electrical isolation	500 V (K-Bus/signal voltage)	
Power supply for the electronics	via the K-bus and through the power contacts	
Current consumption from K-bus (5 V)	typically 100 mA	
Current consumption from the power contacts	typically 35 mA	
Power loss	typically 4 W	
Bit width in the input process image	1 x 8 bit status, 2 x 16 bit data	
Bit width in the output process image	1 x 8 bit control, 2 x 16 bit data	
Pluggable wiring	for all KSxxxx Bus Terminals	
Dimensions (W x H x D)	approx. 27 mm x 100 mm x 70 mm (width aligned: 24 mm)	
Weight	approx. 100 g	
Permissible ambient temperature range during operation	0 °C ... + 55°C	
Permissible ambient temperature range during storage	-25 °C ... + 85°C	
Permissible relative air humidity	95%, no condensation	
Mounting [▶ 35]	on 35 mm mounting rail conforming to EN 60715	
Vibration / shock resistance	conforms to EN 60068-2-6/EN 60068-2-2-7, see also Installation instructions for enhanced mechanical load capacity [▶ 37]	
EMC immunity/emission	conforms to EN 61000-6-2 / EN 61000-6-4, EN 61800-6-3* *) standard: category C3. Additional filters are required for categories C1 and C2!	
Protection class	IP20	
Permissible installation position	without fan cartridge ZB8610: standard installing position with fan cartridge ZB8610: standard installing position, other installing positions (example 1 and 2), see notice [▶ 38]	
Approval	CE, cULus [▶ 44]	

2.7 KL2541 - LED Displays

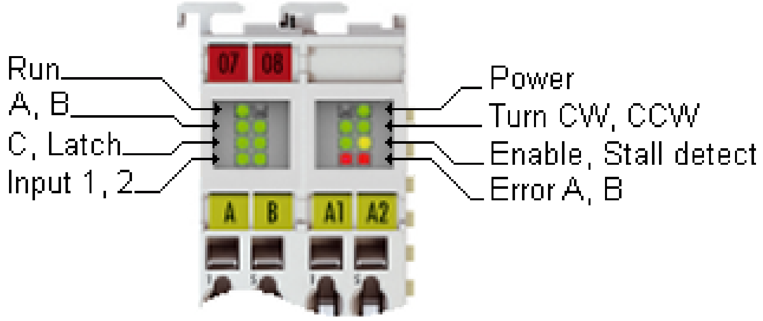


Fig. 4: KL2541 - LED

Left LED prism

LED	Display	
Run (green)	on	Data transmission on the K-bus is active
A (green)	on	A signal is present at encoder input A.
C (green)	on	A signal is present at encoder input C.
Input 1 (green)	on	A signal is present at digital input E1.
-	reserved	
B (green)	on	A signal is present at encoder input B.
Latch (green)	on	A signal is present at the latch input of the encoder.
Input 2 (green)	on	A signal is present at digital input E2.

Right LED prism

LED	Display	
-	reserved	
Turn CW (green)	on	Motor turns clockwise
Enable (green)	off	Motor control is disabled (bit SB.0 [▸ 74] is not set in the status byte) or KL2541 is not ready to operate
	on	Motor control is disabled (bit SB.0 [▸ 74] is set in the status byte) and KL2541 is ready to operate
Error A (red)	flashes (1 Hz)	Configuration error, e.g.: - Value for motor full steps is 0 (R33 [▸ 84]) - Value for encoder increments is 0 (R34 [▸ 84]) - Value for min. velocity v_{min} is 0 (R38 [▸ 85]) - Value for max. velocity v_{max} is 0 (R39 [▸ 85]) - Value for max. acceleration a_{max} is 0 (R40 [▸ 85]) - Value for coil current I_S for $a > a_{th}$ is 0 (R42 [▸ 85]) - Value for coil current I_S with $a \leq a_{th}$ is 0 (R43 [▸ 85])
Power (green)	off	Supply voltage (50 V_{DC}) not available or motor control is disabled (bit SB.0 [▸ 74] is not set in the status byte)
	on	Supply voltage (50 V_{DC}) present
Turn CCW (green)	on	Motor turns counterclockwise
Stall detect (yellow)	on	Motor control is disabled (bit SB.0 [▸ 74] is not set in the status byte)
	flashes (1 Hz)	The current load angle [▸ 18] (SB.3 to SB.1 [▸ 74]) is greater than the configured load angle threshold (R47 [▸ 85]).
Error B (red)	flashes (1 Hz)	Supply or load error - Supply voltage (24 V_{DC}) too low (bit SW.9 [▸ 77] in the status word is set) - Open load at bridge A (bit SW.11 [▸ 77] is set in the status word) - Open load at bridge B (bit SW.12 [▸ 77] is set in the status word) - Overcurrent at bridge A (bit SW.13 [▸ 77] is set in the status word) - Overcurrent at bridge B (bit SW.11 [▸ 77] is set in the status word)

2.8 Basic function principles

The KL2531 and KL2541 Stepper Motor Terminals integrate a compact motion control solution for stepper motors up to 200 W in the smallest design.

Stepper motor

Stepper motors are electric motors and are comparable with synchronous motors. The rotor is designed as a permanent magnet, while the stator consists of a coil package. In contrast to synchronous motors, stepper motors have a large number of pole pairs. In the minimum controller configuration, the stepper motor is moved from pole to pole, or from step to step.

Stepper motors have been around for many years. They are robust, easy to control, and provide high torque. In many applications, the step counting facility saves expensive feedback systems. Even with the increasingly widespread use of synchronous servomotors, stepper motors are by no means "getting long in the tooth". They are considered to represent mature technology and continue to be developed further in order to reduce costs and physical size, increase torque and improve reliability.

The development of the KL2531 and KL2541 Bus Terminals for the Beckhoff Bus Terminal system opens up new application areas. Microstepping and the latest semiconductor technology offer many advantages:

- smoother operation
- avoidance of resonance
- reduced energy consumption
- lower thermal load on the motor
- minimum electromagnetic emissions
- long cable lengths
- simpler handling
- reduced size of the power electronics
- simple integration into higher-level systems
- integrated feedback system

Two stepper motor terminals for optimum performance

The KL2531 and KL2541 Stepper Motor Terminals differ in terms of performance class.

- With a size of only 12 mm, the KL2531 covers the lower performance range. The supply voltage can be up to 24 V_{DC}. The device is designed for simple integration into the 24 V_{DC} control voltage system. With a peak current of 1.5 A per phase, a large number of small drives and adjustable axes can be supplied.
- The KL2541 offers higher performance class comparable to that of small servo drives. With a peak current of 5 A, the KL2541 can generate an impressive torque of 5 Nm in conjunction with a standard stepper motor, for example. The supply voltage of up to 50 V_{DC} permits high speeds with good torque and therefore high mechanical output up to approx. 200 W. The KL2541 features an integrated incremental encoder interface for accommodating all connection wires for a drive but is nevertheless very compact with a width of only 24 mm.

Both stepper motor terminals provide two controlled sine/cosine currents. 25 kHz current control enables smooth current output without resonance. Highly dynamic, low-inductance motors run just as well as stepper motors with small rotor mass. The current resolution is 64 steps per period (64-fold microstepping). The standard motor with a 1.8° step angle runs very smoothly and can be set to up to 12,800 electronic positions per turn. Experience shows that approx., 5000 positions are realistic in terms of the mechanics.

Typical stepper motor problems such as pronounced resonance are therefore a thing of the past. Microstepping and associated set values ensure that rotor jerk is avoided. Also, the rotor no longer tends to oscillate around each indexing position. Mechanical measures such as vibration dampers against resonance or gear reduction for increasing precision are no longer required. This allows the burden from costs and development effort to be lower.

The new stepper motor terminals also reduce development time on the control side. Both Bus Terminals can be used just like standard Bus Terminals in all common fieldbuses. Interface programming is therefore no longer required. Start, stop or resonance frequencies are no longer an issue. For simple positioning tasks, both Bus Terminals can automatically position the drive, taking account of an acceleration ramp and the maximum frequency.

The option of detecting the rotor position via the voltage returned by the stepper motor is not yet used widely. The KL2531 and KL2541 Bus Terminals offer status feedback that reflects the motor load with a resolution of 3 bits. This type of feedback is not suitable for "real" position control. However, since the stepper motor basically follows its control and simply stops in the event of overload, the technique is acceptable in practice: The motor will reach the specified position, as long as it is not overloaded. The position value counted in the Bus Terminal is "O.K."

Realization of more demanding positioning tasks

More demanding positioning tasks can be realized via the TwinCAT automation software from Beckhoff. Like other axes, the two stepper motor terminals are integrated via the TwinCAT System Manager and can be used like standard servo axes. Special stepper motor features, such as speed reduction in the event of large lag errors, are automatically taken into account via the *stepper motor axis* option. The effort for changing from a servomotor to a stepper motor - and back - is no greater than changing from one fieldbus to another one under TwinCAT.

The output stages of the stepper motor terminals have an overload protection in the form of an overtemperature warning and switch-off. Together with short circuit detection, diagnostic data are accessible in the process image of the controller. In addition, this status is displayed by the Bus Terminal LEDs, along with other information. The output stage is switched on via an Enable-Bit. The motor current can be set and reduced via a parameter value.

Optimum adaptation to the motor and the implementation of energy-saving features require minimum programming effort. During the test phase, the KS2000 Configuration Software enables quick and efficient optimization. Since all data are set via software parameters, Bus Terminals can easily be exchanged and parameters stored or transferred to the next project. It is therefore no longer necessary to transfer certain potentiometer settings or to document DIP switch settings.

2.8.1 Stepper motor parameters

Torque

Refers to the maximum motor torque at different speeds. This parameter is usually represented by a characteristic curve. Stepper motors have comparatively high torque in the lower speed range. In many applications, this enables them to be used directly without gearing. Compared with other motors, stepper motors can quite easily provide a holding moment of the same order of magnitude as the torque.

Speed

Stepper motors have low maximum speed, which is usually specified as a maximum step frequency.

Number of phases

Motors with 2 to 5 phases are common. The KL2531 and KL2541 Bus Terminals support 2-phase motors. 4-phase motors are basically 2-phase motors with separate winding ends. They can be connected directly to the Bus Terminal.

Nominal voltage, supply voltage and winding resistance

Under steady-state conditions, the rated current at the rated voltage depends on the winding resistance. This voltage should not be confused with the supply voltage of the power output stage in the Bus Terminal. The KL2531 and KL2541 apply a controlled current to the motor winding. If the supply voltage falls below the nominal voltage, the power output stage can no longer apply the full current, resulting in a loss of torque. It is desirable to aim for systems with small winding resistance and high supply voltage in order to limit warming and achieve high torque at high speeds.

Resonance

At certain speeds, stepper motors run less smoothly. This phenomenon is particularly pronounced if the motor runs without load. Under certain circumstances, it may even stop. This is caused by resonance. A distinction can roughly be made between

- resonances in the lower frequency range up to approx. 250 Hz and
- resonances in the medium to upper frequency range.

Resonances in the medium to upper frequency range essentially result from electrical parameters such as inductance of the motor winding and supply line capacity. They can be controlled relatively easily through high pulsing of the control system.

Resonances in the lower range essentially result from the mechanical motor parameters. Apart from their impact on smooth running, such resonances can lead to significant loss of torque, or even loss of step of the motor, and are therefore particularly undesirable.
In principle, the stepper motor represents an oscillatory system (comparable to a mass/spring system), consisting of the moving rotor with a moment of inertia and a magnetic field that creates a restoring force that acts on the rotor. Moving and releasing the rotor creates a damped oscillation. If the control frequency corresponds to the resonance frequency, the oscillation is amplified, so that in the worst case the rotor will no longer follow the steps, but oscillate between two positions.
Due to their sine/cosine current profile, KL2531 and KL2541 Bus Terminals are able to prevent this effect in almost all standard motors. The rotor is not moved from step to step, so he no longer jumps to the next position, but it moves through 64 intermediate steps, i.e. the rotor is gently moved from one step to the next. The usual loss of torque at certain speeds is avoided, and operation can be optimized for the particular application. This means that the lower speed range, where particularly high torque is available, can be fully utilized.

Step angle

The step angle indicates the angle travelled during each step. Typical values are 3.6°, 1.8° and 0.9°. This corresponds to 100, 200 and 400 steps per motor revolution. Together with the downstream transmission ratio, this value is a measure for the positioning accuracy. For technical reasons, the step angle cannot be reduced below a certain value. Positioning accuracy can only be improved further by mechanical means (transmission). An elegant solution for improving positioning accuracy is the microstepping function offered by the KL2531 and KL2541. It enables up to 64 intermediate steps. The smaller "artificial" step angle has a further positive effect: The drive can be operated at higher velocity, yet with the same accuracy. The maximum speed is unchanged, despite the fact that the drive operates at the limit of mechanical resolution.

2.8.2 Specifying the stepper motor

1. Determine the required positioning accuracy and hence the step resolution. The first task is to determine the maximum resolution that can be achieved. The resolution can be increased via mechanical gear reduction devices such as spindles, gearing or toothed racks. The 64-fold microstepping of the stepper motor terminals also has to be taken into account.
2. Determine mass (m) and moment of inertia (J) of all parts to be moved
3. Calculate the acceleration resulting from the temporal requirements of the moved mass.
4. Calculate the forces from mass, moment of inertia, and the respective accelerations.
5. Convert the forces and velocities to the rotor axis, taking account of efficiencies, moments of friction and mechanical parameters such as gear ratio. It is often best to start the calculation from the last component, usually the load. Each further element transfers a force and velocity and leads to further forces or torques due to friction. During positioning, the sum of all forces and torques acts on the motor shaft. The result is a velocity/torque curve that the motor has to provide.
6. Using the characteristic torque curve, select a motor that meets these minimum requirements. The moment of inertia of the motor has to be added to the complete drive. Verify your selection. In order to provide an adequate safety margin, the torque should be oversized by 20% to 30%. The optimization is different if the acceleration is mainly required for the rotor moment of inertia. In this case, the motor should be as small as possible.
7. Test the motor under actual application conditions: Monitor the housing temperatures during continuous operation. If the test results do not confirm the calculations, check the assumed parameters and boundary conditions. It is important to also check side effects such as resonance, mechanical play, settings for the maximum operation frequency and the ramp slope.
8. Different measures are available for optimizing the performance of the drive: using lighter materials or hollow instead of solid body, reducing mechanical mass. The control system can also have significant influence on the behaviour of the drive. The Bus Terminal enables operation with different supply voltages. The characteristic torque curve can be extended by increasing the voltage. In this case, a current increase factor can supply a higher torque at the crucial moment, while a general reduction of the current can significantly reduce the motor temperature. For specific applications, it may be advisable to use a specially adapted motor winding.

2.8.3 Power rating of the motor power supply

For typical applications the power rating of the motor power supply can be dimensioned such that for the 50% of the current of a stepper motor phase is assumed for the current consumption.

CAUTION

For larger load torques the actual maximum current consumption should be measured accurately!

This is only an estimate for typical applications. In cases where larger load torques occur at the motor, this configuration is insufficient. The actual maximum current consumption occurring must be measured and the motor power supply dimensioned accordingly.

Example for typical applications

A KL2541 that operates a stepper motor with a maximum coil current of 5 A in a typical application can generally be operated with a 48 V/2.5 A motor supply.

2.8.4 Further parameters

Load angle

The load angle provides information about the current mechanical load at the motor axis. It is shown as a 3-bit value (SB.1-SB.3 [▶ 74]) and updated after each full step. Since the load angle is determined via the motor current, it directly depends on the following factors:

- Velocity - A medium velocity is an advantage for an analyzable load angle; high or low velocities result in a high load angle.

- Resonances - Motor resonances generate high mechanical load at the motor and distort the measuring result.
- Acceleration - Acceleration phases also generate high load at the motor.
- Mixed Decay - If this option is enabled, the motor current is actively impressed. Deactivation via CW.9 [▸ 76] therefore has a positive effect on load angle resolution.
- Motor current - The set coil current has a direct influence on the load angle resolution, i.e. the smaller the current, the smaller the resolution.

For each application, the user should therefore determine the optimum velocity for achieving a satisfactory load angle resolution.

Current table

It is conceivable that in some applications it may be necessary to adapt the current (which usually is sinusoidal) to the stepper motor. To this end feature bit R32.7 [▸ 84] has to be enabled. The user should save the adjusted current table in Register page 1 [▸ 88].

Mixed Decay

The Mixed Decay parameter can be used to refine and reduce the coil current. This is achieved by the auxiliary transistor actively impressing a coil current onto a half bridge during the second half of the microstep phase through pulsing. In microstep mode, this has a positive influence on the motor movement. The motor operates more smoothly and can by positioned more accurately. Mixed Decay should be switched off for low velocities and at standstill.

Mixed Decay can be disabled via control bit CW.9 [▸ 76].

⚠ CAUTION	**Position shift on deactivation of Mixed Decay possible** When Mixed Decay is deactivated, in the worst case the position may shift by a few microsteps due to the change in coil currents!

Automatic / manual current reduction

The stepper motor terminal offers the user the option of reducing the current in order to prevent unnecessary heating of the motor:

- automatic, specified via register R44 [▸ 85]
- manual, specified via register R45 [▸ 85]

While the motor is at standstill (v=0), the holding current from R44 is set automatically. This value refers to the set terminal coil current, not to the rated terminal current!

In order to achieve maximum control for the user, this value can be set to 100 % while R45 is set to 50 %, for example. A holding current can be applied manually onto the motor windings by setting control bits CB.3 [▸ 74] or CW.11 [▸ 76]. This can be done at standstill or while the motor is in operation.

Setting the target position (via register)

The user can set or delete the target position value. Registers R2 [▸ 81] and R3 [▸ 81] are used as reference:

- a rising edge at CW.13 [▸ 76] deletes the set target position (higher priority than CW.10 if CW.10 and CW.13 have the same setting)
- a rising edge at CW.10 [▸ 76] sets the target position to the value from register R2 and R3

Acknowledgement occurs via status bit SW.2 [▸ 77].

Path control

For positioning taken over from a PLC, path control [▶ 28] is the optimum solution. In this operation mode, a 32 bit position value and various parameter such as velocity and acceleration are specified for the terminal. Once enabled, the terminal automatically travels to the target position.

Detecting position errors

The terminal determines the relative position error of the motor via an external sensor (e.g. inductive proximity limit switch) connected to digital input 2 (only for internal positioning). Register RP0.R51 [▶ 86] is used to specify the number of pulses per revolution specified (if a pinion is used, for example). By setting bit CW.7 [▶ 76] of the control word, the user can show the error in the process data and subsequently analyses it with the PLC.

The terminal uses these parameters to calculate a position offset:

1.1	IF = MS x Reg33 / Reg51	Pulse factor
1.2	∆Pos = Pos - IC x IF	Relative position error

Legend:

IF	Pulse factor	[IF] = microsteps per pulse
IC	Pulse counter	Pulses counted at digital input 2
MS	Microsteps per full step	MS = 2^{R46}
Reg33	Full motor steps	
Reg46	Step size per quarter period	
Reg51	Number of pulses per revolution	
Pos	Current position (set value)	[Pos] = microsteps
∆Pos	Relative position error	[∆Pos] = microsteps

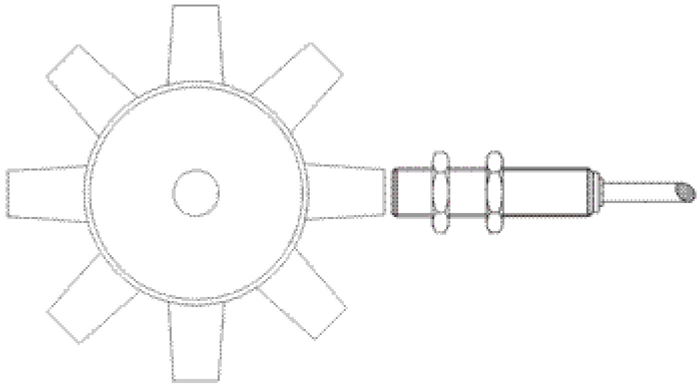


Fig. 5: Pinion with inductive proximity limit switch (register R51 = 8)

The user must analyse the relative position error individually. No direct conclusions can be drawn for the actual number of lost microsteps. This is due to the fact that the terminal does not take the initial error (number of microsteps between the start of the motor and the first sensor pulse) into account. In addition, it is not possible to include the tolerance (delay), with which the sensor switches the 24 V.

These preliminary considerations results in a velocity dependence, which must be taken into account in the evaluation. At constant velocity the error fluctuates around a few microsteps, but remains the same on average. In practice, the small deviation is irrelevant, since in the event of an error the motor:

- either loses many steps at once (this is clearly indicated by a suddenly increasing error), or
- stops altogether, which is indicated by a steadily increasing position error.

Encoder interface

The encoder operates with fourfold evaluation.

Latch functions

The internal encoder offers the option of registering one or several latch events. A latch event can be generated through the input signals C, latch/gate, input 1 or input 2. By default, the terminal only stores one latch value. The latch array can be enabled and the number of latch values increased by setting the feature bit R32.9 [▶ 84] and paramétrisation of register R37 [▶ 85], so that several latch values can be stored.

The terminal response to the latch events is enabled as follows:

- Setting control bit CW.0 [▶ 76] activates a rising edge at the C input (maximum priority if several control bits are set at the same time)
- Setting the control bits CW.3 [▶ 76] activates the rising edge at the latch/gate input (second-highest priority)
- Setting the control bits CW.4 [▶ 76] activates the falling edge at the latch/gate input (third-highest priority)
- Setting the control bits CW.1 [▶ 76] activates the rising edge at digital input E1 (fourth-highest priority)
- Setting the control bits CW.2 [▶ 76] activates the rising edge at digital input E2 (lowest priority)

By activating R32.8 [▶ 84] a latch event may also be used for deleting the current position. To this end, one of the above-mentioned events must be enabled first, followed by control bit CW.13 [▶ 76]. The current position is deleted during the next latch event.

Once the user has enabled the function, the terminal saves the current position value at the next latch event and indicates this by setting status bit SW.4 [▶ 77]. If the latch array is activated, this only happens once the number of latch events specified in R37 has occurred. Reading of latch values must be initiated by setting CW.5 [▶ 76], which causes the low-order word to be shown in the DataIN process data (the terminal indicates this via status bit SW.5 [▶ 77]). The high-order word can only be read from register R5 [▶ 81] via register communication. The following latch values can be retrieved by changing control bit CW.6 [▶ 76]. The terminal acknowledges this by changing status bit SW.6 [▶ 77]. The next latch value now applies and is shown in DataIN. The last latch value has been reached once terminal SW.6 no longer changes according to CW.6.

Note	**Enable must be maintained!** The enable that was set previously must be retained while reading out the latch value. The latch values are lost if the enable is removed!

Digital inputs

The digital inputs can be configured individually for N/C contacts. To this end, for input 1 bit RP0.R52.14 [▶ 86] and for input 2 bit RP0.R52.15 [▶ 86] of feature register 2 [▶ 86] is set to 1_{bin}. In delivery state both inputs are configured for N/O contacts.

2.8.5 Error indication

The stepper motor terminal offers various diagnostic options for the user. These error messages are subdivided into configuration errors, hardware warnings and hardware errors.

Configuration errors

In the event of an illogical configuration, i.e. if one of the registers R33, R34 (KL2541 only), R38, R39, R40, R42, R43 or R50 is zero, the bit ConfigError (SW.15 [▶ 77]) is set in the status word. From these registers, the terminal calculates further internal path control parameters. If these calculations return values outside the permissible range of values, i.e. if a particular target position can never be reached with the specified velocities and accelerations, this error bit is also set. It is reset automatically once the calculations return a correct result by changing the register values. These errors do not cause status bit SB.6 [▶ 74] to be set!

Hardware warnings

The bit SB.5 [▸ 74] (warning) is set in the status byte if the internal terminal temperature reaches 80 °C. SB.5 is automatically reset when the temperatures falls below 60 °C again.

Hardware error

The motor power is switched off and bit SB1.6 [▸ 74] in the status byte is set if one of the following errors occurs.

- Overtemperature (SW.8 [▸ 77])
- Undervoltage (UnderVoltage SW.9 [▸ 77])
- Open load, bridge A (OpenLoad Br A, SW.10 [▸ 77])
- Open load, bridge B (OpenLoad Br B, SW.11 [▸ 77])
- Overcurrent bridge A (OverCurrent Br A, SW.12 [▸ 77])
- Overcurrent bridge B (OverCurrent Br B, SW.13 [▸ 77])
- Outage of the 24 V control voltage (NoControlPower, SW.14 [▸ 77])
  (The terminal is automatically reinitialized when the control voltage returns)

If an error occurs, it first has to be rectified and subsequently acknowledged and thus cancelled by setting bit CB.6 [▸ 74] in the control byte.

2.9 Temperature monitoring

The KL2531 and KL2541 stepper motor terminals feature integrated monitoring for the temperature monitoring inside the housing and the stepper controller chip.

Temperature warning

The bit SB.5 [▸ 74] (warning) is set in the status byte if the temperature in the terminal housing reaches 80 °C. SB.5 is automatically reset when the temperatures falls below 60 °C again.

The current temperature inside the housing can be read from the temperature register R6 [▸ 82].

Automatic shutdown

If the temperature exceeds 125 °C,

- the stepper controller chip switches the motor off automatically. In addition:
- the bit SW.8 [▸ 77] (overtemperature) is set in the status word,
- the LED *error B* flashes,
- the bit SB.6 [▸ 74] (error) is set in the status byte
  (Bit SB.6 [▸ 74] of the status bytes can be reset by setting bit CB.6 [▸ 74] in the control byte.)

❗ **Attention**	**Check temperature monitoring!** Check the temperature monitoring from your user program, if the KL2531 and KL2541 stepper motor terminals are operated at their performance limits!

Temperature diagram for the KL2541

The following diagram is applicable from hardware version [▸ 7] 10 of the KL2541.

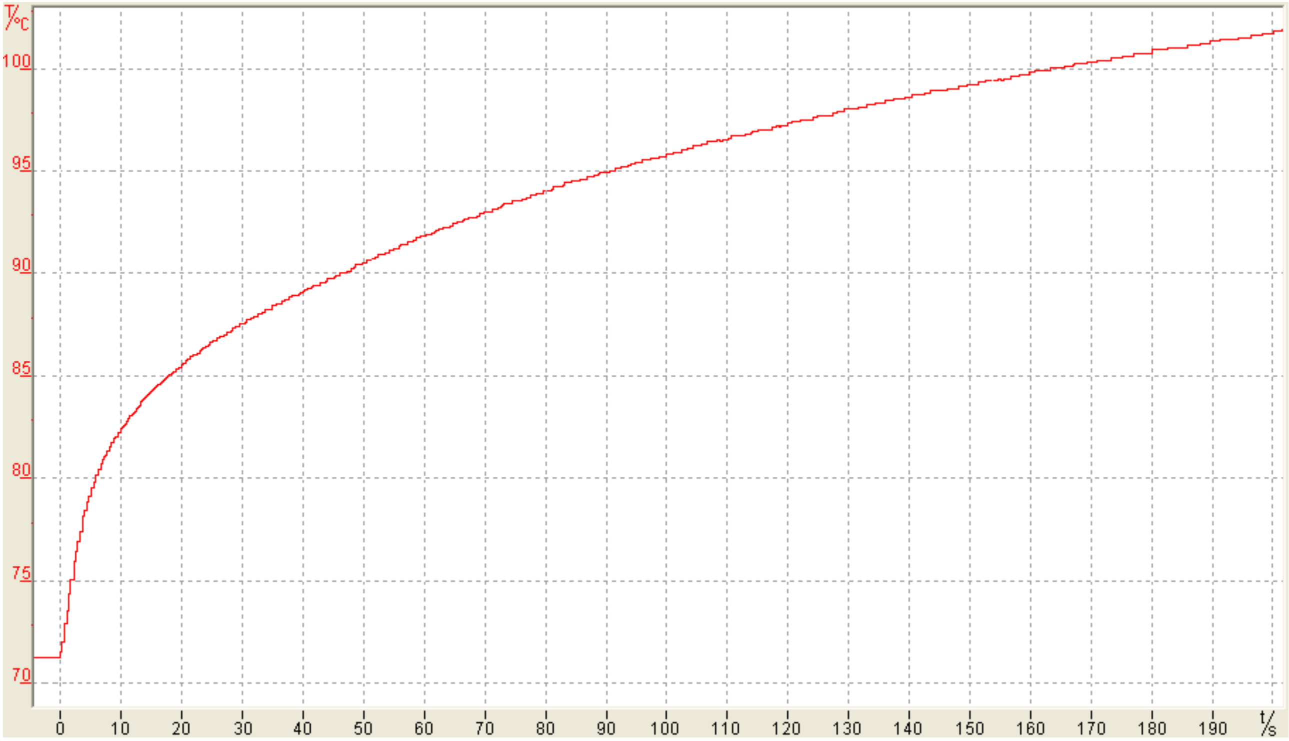


Fig. 6: KL2541 - temperature inside the housing (worst case)

This diagram shows the temperature inside the housing (worst case) of the KL2541 at 55 °C ambient temperature and maximum motor current:

- The x-axis shows the time in seconds.
- The y-axis shows the temperature in °C.

The KL2541 is disabled up to the start of the temperature increase (t = 0 s). Then, a stepper motor is operated with 5 A.

Based on this diagrams the user can estimate how long the KL2541 can be operated at maximum ambient temperature and maximum load current, until the automatic switch-off feature of the stepper controller chip shuts down the motor automatically (125 °C).

2.10 Microstepping

Microstepping can be specified via registers R46 [▸ 85] and R48 [▸ 85] to R63 [▸ 87] or via the current table [▸ 63] dialog in the KS2000 configuration software.

Microstepping	Steps per quarter period
Full step	1
Half step	2
1/4 step	4
1/8 step	8
1/16 step	16
1/32 step	32
1/64 step	64

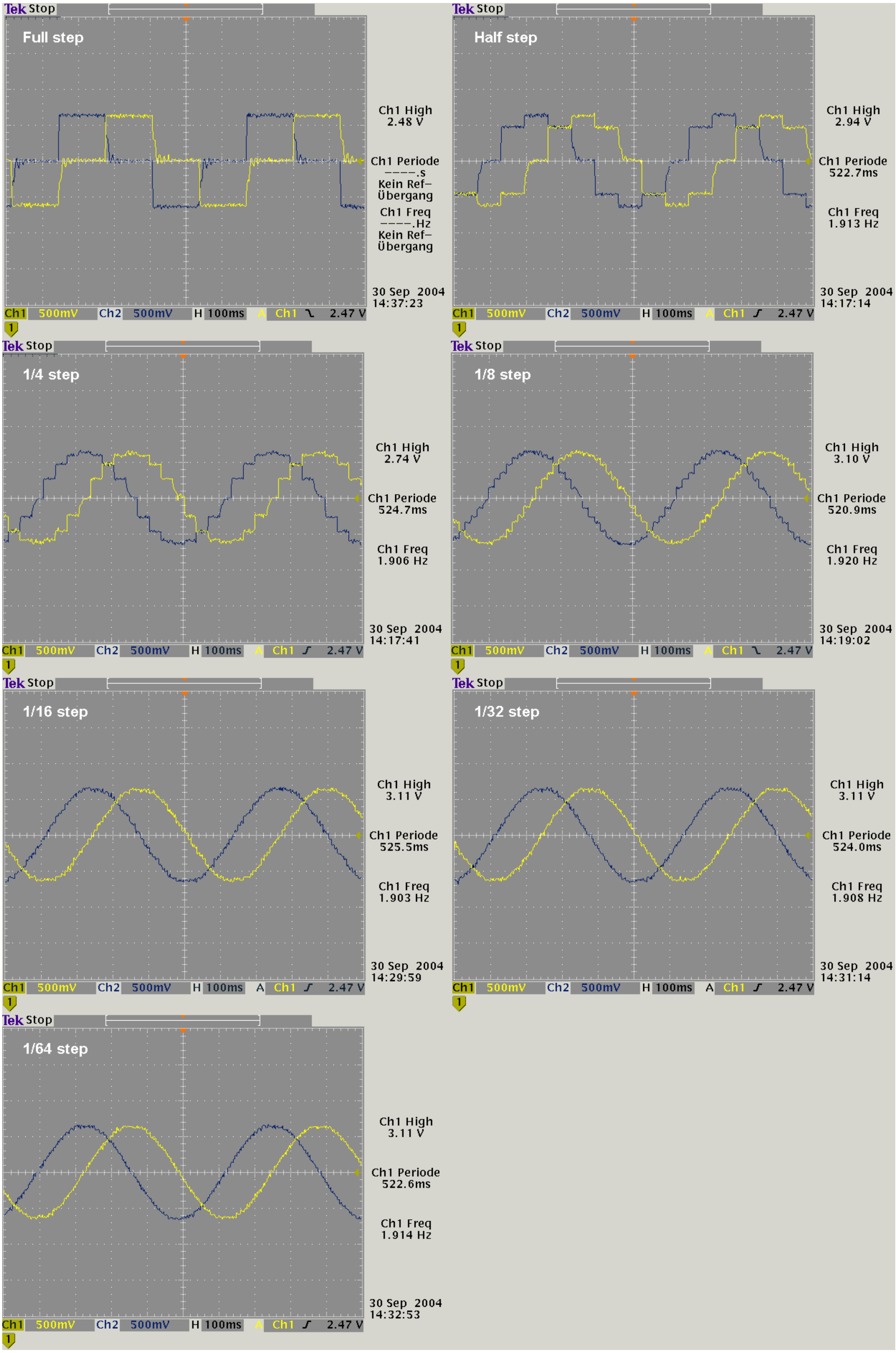


Fig. 7: Current microstepping

2.11 Operation modes

The operating modes can be selected with the configuration software KS2000 [▶ 58] or through register communication via the feature register RP0.R32 [▶ 84].
The KL2531 and KL2541 Stepper Motor Terminals support the following operating modes:

- Velocity, direct [▶ 25]
- Velocity, with ramps [▶ 26]
- Position tracking [▶ 27]
- Travel distance control
    - Manual [▶ 28](velocity, with ramps)
    - Motion command (simple) [▶ 28]
    - Motion command (multiple) [▶ 30]
    - Auto-start function [▶ 32]
    - Auto-stop function [▶ 32]
    - fast-stop function [▶ 32]
    - Referencing (via the digital inputs) [▶ 33]
    - Self-adjustment [▶ 34]

Operation mode		for operation with	Range of values for DataOUT
Velocity, direct		NC, cyclic velocity interface	-7FFF$_{hex}$ to +7FFF$_{hex}$
Velocity, with ramps		PLC	-7FFF$_{hex}$ to -0010$_{hex}$ +0010$_{hex}$ to +7FFF$_{hex}$
Position tracking (in preparation)		NC, cyclic position interface	-7FFF$_{hex}$ to +7FFF$_{hex}$
Travel distance control	Manual	PLC	-7FFF$_{hex}$ to -0010$_{hex}$ +0010$_{hex}$ to +7FFF$_{hex}$
	Motion command		0000$_{hex}$

2.11.1 Velocity, direct

This mode is intended for the cyclic velocity interface of a numeric controller (NC). In this mode, the NC specifies a set velocity. Ramps for start-up and deceleration of the motor are also controlled by the NC.

DataOUT

The terminal analyses 15 bits and the sign (VZ):

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Contents	VZ	Velocity value														

Range of values: -7FFF$_{hex}$ to +7FFF$_{hex}$ (-32767$_{dec}$ to +32767$_{dec}$)

2.11.2 Velocity, with ramps

This mode is intended for simple motor control from a conventional PLC (not NC).

The stepper motor terminal follows the velocity specifications for the PLC, taking into account ramps, which the terminal determines based on the parameters *maximum velocity* (RP0.R39 [▶ 85]), *maximum acceleration* (RP0.R40 [▶ 85]) and *minimum velocity* (RP0.R38 [▶ 85]). These parameters can be deposited in the terminal registers via register communication or parameterized via the KS2000 [▶ 58] configuration software.

DataOUT

The terminal analyses 11 bits and the sign (VZ), which means in this mode the resolution is by a factor 16 poorer than in *velocity, direct* mode:

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Contents	VZ	Velocity value											0	0	0	0

Range of values: -7FFF$_{hex}$ to -10$_{hex}$ and +10$_{hex}$ to +7FFF$_{hex}$ (-32767$_{dec}$ to -16$_{dec}$ and +16$_{dec}$ to +32767$_{dec}$)

Calculation of the velocity (in microsteps)

The stepper motor terminal converts the value contained in the output data word into a velocity according to the following formula.

In the operating modes *velocity, with ramps* and *path control* the maximum velocity specified through register RP0.R39 [▶ 85] limits the motor speed.

1.1	R = DataOUT / 32767 x f x 2047 / 262144	DataOUT = R / f x 262144 x 32767 / 2047
Simplified formulas:		
1.2	R = DataOUT x 3,812951	DataOUT = R / 3,812951
1.3	R_{FS} = DataOUT x 3,812951 / MS	DataOUT = R_{FS} x MS / 3,812951
1.4	RPS = DataOUT x 3,812951 / (R33 x MS)	DataOUT = RPS x R33 x MS / 3,812951

Conversion to full steps / revolutions per second

2.1	R_{FS} = R / MS; R_{FS} = R / 2^{R46}
2.2	RPS = R_{FS} / R33

with

R	Velocity in microsteps	[R] = microsteps / s
R_{FS}	Velocity in full steps	[R_{FS}] = Full steps / s
RPS	Revolutions per second	[RPS] = N / s
MS	Microsteps	MS = 2^{R46}
f	Internal clock frequency	f = 16 MHz
RP0.R33 [▶ 84]	Value in register 33	Number of full step motor steps per revolution
RP0.R46 [▶ 85]	Value in register 46	Number of steps/quarter period
DataOUT [▶ 73]	Value in the output data word	

Calculation of the register values (Reg) for R38, R39, R40, R41, R50, R58

Minimum and maximum velocity

3.1	Reg = R / f x 262144	Velocity in MS / s
3.2	Reg = R_{FS} x MS / f x 262144	Velocity in FS / s
Simplified formulas:		
3.3	Reg = R x 0.016384	Velocity in MS / s
3.4	Reg = R_{FS} x MS x 0.016384	Velocity in FS / s

with:

R	Velocity in microsteps	[R] = microsteps / s
R_{FS}	Velocity in full steps	[R_{FS}] = Full steps / s
RPS	Revolutions per second	[RPM] = N / s
MS	Microsteps	MS = 2^{R46}
f	Internal clock frequency	f = 16 MHz
Reg	Register value R38 or R39	

Maximum acceleration and acceleration threshold

4.1	Reg = ΔR x 2^{38} / f^2	Acceleration in MS/s^2
4.2	Reg = ΔR_{FS} x MS x 2^{38} / f^2	Acceleration in FS/s^2
Simplified formulas:		
4.3	Reg = ΔR x 1,073742 x 10^{-3}	Acceleration in MS/s^2
4.4	Reg = ΔR_{FS} x MS x 1.073742 x 10^{-3}	Acceleration in FS/s^2

Conversion to full steps

5.1	ΔR_{FS} = ΔR / MS ΔR_{FS} = R / 2^{R46}

with:

ΔR	Acceleration	[ΔR] = microsteps / s^2
ΔR_{FS}	Acceleration in full steps	[ΔR_{FS}] = Full steps / s^2
MS	Microsteps	MS = 2^{R46}
f	Internal clock frequency	f = 16 MHz
Reg	Register value R40 or R41	
a_{max}	Maximum acceleration (RP0.R40 [▸ 85])	[a_{max}] = 1

2.11.3 Position tracking

(in preparation)

This mode is intended for the cyclic position interface of a numeric controller (NC). The NC specifies a 16-bit set position. The terminal tries to track this position via a velocity calculated by the internal position controller. The two parameters in register RP0.R48 [▸ 85] (Kp factor) and RP0.R49 [▸ 85] (Kv factor) are used for this purpose.

2.11.4 Path control

This mode is intended for simple positioning from a conventional PLC (not NC). *Path control* mode offers the following variants:

- Manual
- Motion command (simple)
- Specifying the set position via process data (KL2531-1000 and KL2541-1000 only)
- Motion command (multiple)
- Auto-start function
- Quick-start function
- Referencing
- Self-adjustment

2.11.4.1 Manual

Manual mode is identical to velocity interface with ramps [▶ 26]. The terminal automatically selects this mode if no motion command is active (CB.2 [▶ 74] = 0).

2.11.4.2 Motion command (simple)

With simple path control, a position value is specified for the terminal, from which the current position may only deviate by up to +/- 2^{31} position increments. The controller can read the current position from registers R0 [▶ 81] and R1 [▶ 81]. While a motion command is active, the controller can already parameterize the terminal for the next motion command. The new parameters are written to the RAM section of the register (to this end, the code word register must be deleted!). The procedure is described in more detail in the example below.

To ensure meaningful and effective motion, the following registers can be re-parameterized:

- R2 [▶ 81]: Target position (low-order word)
- R3 [▶ 81]: Target position (high-order word)
- RP0.R38 [▶ 85]: Minimum velocity
- RP0.R39 [▶ 85]: Maximum velocity
- RP0.R40 [▶ 85]: Maximum acceleration
- RP0.R58 [▶ 87]: Maximum deceleration (must be activated through bit RP0.R52.1 [▶ 86] in feature register 2)

Sample

In the following example, a motion command is started, and a new position is specified during the motion. A second motion command is then started. The motor is then moved manually and stopped again.

1. In the control byte, set bit CB.0 [▶ 74] to 1 for enabling motor control.
2. Enter the target position (32 bit) in registers R2 [▶ 81] and R3 [▶ 81] via register communication.
3. Enter the remaining parameters in registers R37 [▶ 85], R38 [▶ 85], R39 [▶ 85], R40 [▶ 85] and R58 [▶ 87] (all register page 0) via register communication, if changes are required.
4. The process output data DataOUT [▶ 73] now have to be zero for the motion command to be started.
5. In the control byte, set bit CB.2 [▶ 74] to 1 for starting the motion command. The stepper motor terminal will now independently move the motor to the target position. Leave bit CB.2 [▶ 74] set to 1 until the motor has reached the desired position. To abort the motion command delete bit CB.2 [▶ 74].
6. During the active motion command enter the new parameters in registers R37, R38, R39, R40 and R58 via register communication, if they are to be modified.

7. When the target position is reached, the terminal reports this by setting bit SW.3 [▸ 77] in the status word.
8. Deleting bit CB.2 [▸ 74] in the control byte completes the first motion command.
9. The terminal acknowledges this by deleting bit SW.3 [▸ 77].
10. Set bit CB.2 [▸ 74] to 1 for starting the new motion command.
11. When the second target position is reached, the terminal reports this by setting again bit SW.3 [▸ 77].
12. Deleting bit CB.2 [▸ 74] in the control byte completes the second motion command.
13. The terminal acknowledges this by deleting again bit SW.3 [▸ 77].
14. Enter a velocity value in DataOUT via process data communication for moving the motor manually.
15. Delete the velocity value in DataOUT in order to stop the motor.

2.11.4.3 Specifying the set position via process data (KL2531-1000 and KL2541-1000 only)

Alternatively, in the special KL2541-1000 version you can specify the target position via the process data. The procedure is as follows:

1. Save the current velocity value.
2. Switch the process output data DataOUT [▸ 73] by setting bit CW.14 [▸ 76] (position interface active).
3. Write the new set position to the process output data DataOut.
4. Apply the new set position by setting bit CW.10 [▸ 76].
5. Reset the target velocity to the saved velocity value.
6. Reset the process output data DataOUT [▸ 73] by cancelling bit CW.14 [▸ 76] (velocity interface active).
7. Start the new motion command by setting bit CB.2.

2.11.4.4 Motion command (multiple)

With multiple path control, up to 16 position values, 16 velocities (from firmware 3E) and 16 accelerations (from firmware 3E) are specified for the terminal, which are stored internally in different tables. A position is comprised of two registers, which are stored in registers R32 to R63 of register page 2 [▸ 89]. Each of these position values may only deviate from the current position by up to +/- 2^{31} positions increments. The PLC can read the current position from registers R0 [▸ 81] and R1 [▸ 81]. The velocities are stored in registers R32 to R47, the accelerations in registers R48 to R63 on register page 3 [▸ 90].
During an active motion command the controller can parameterize the terminal for the next motion command (except R39 and R40, which are overwritten when the next motion command starts). The new parameters are stored in the RAM section of the terminal (the code word register must be cleared for this purpose!).

A multiple motion command is started by writing into register R7

- with command 0x0500 [▸ 83] (the internal position index is incremented after each motion command until position 15 is reached, i.e. the table is processed in forward direction), or
- with command 0x0510 [▸ 83] (the internal position index is decremented after each motion command until position 0 is reached, i.e. the table is processed backwards).

The start position is specified by adding an offset to command code 0x0500 [▸ 83] or 0x0510 [▸ 83] (e.g. 0x0504 -> start position 4 forward; 0x0518 -> start position 8 backward). A multiple motion command can be stopped with the command 0x0540 [▸ 83]. The procedure is described in more detail in the sample below.

Up to 16 possible target positions can be specified in the following registers:

- RP2.R32 [▸ 89]: Target position 0 (low-order word)
- RP2.R33: Target position 0 (high-order word)
- RP2.R34: Target position 1 (low-order word)
- RP2.R35: Target position 1 (high-order word)
- etc.

Up to 16 possible velocities can be specified in the following registers:

- RP3.R32 [▸ 90]: velocity 0
- RP3.R33: Velocity 1
- etc.

Up to 16 possible accelerations can be specified in the following registers:

- RP3.R48 [▸ 91]: Acceleration 0
- RP3.R49: Acceleration 1
- etc.

The following registers can be re-parameterized for meaningful and effective motion with multiple positioning:

- RP0.R38 [▸ 85]: Minimum velocity
- RP0.R58 [▸ 87]: maximum deceleration (must be activated through bit RP0.R52.1 [▸ 86])

Sample

In the following sample a multiple motion command is started at index 5, after which three positions are approached (positions 5, 6 and 7). The internal position index is then set to 3 set, and two positions are approached (positions 3 and 2). The multiple motion command is then stopped.

1. Enter the target positions (max. 16 positions, each 32 bit) in registers RP2.R32 to RP2.R63 [▸ 89] using register communication.
2. Set bit CB.0 [▸ 74] of the control byte to 1_{bin}, in order to enable the motor control.
3. Enter the other parameters in registers RP0.R37 [▸ 85], RP0.R38 [▸ 85], RP0.R39 [▸ 85], RP0.R40 [▸ 85] and RP0.R58 [▸ 87] (all register page 0) via register communication, if these are to be re-parameterized.
4. The process output data DataOUT [▸ 73] now have to be zero for the motion command to be started.
5. Enter the start command 0x0505 [▸ 83] (0x0500 + offset 5) in register R7 [▸ 82] via register communication, in order to set the internal position index to position 5 as the first target position.
6. In the control byte set bit CB.2 [▸ 74] to 1_{bin} to start the first motion command.
7. The stepper motor terminal will now independently move the motor to the target position. Leave bit CB.2 [▸ 74] set to 1_{bin} until the motor has reached the desired position. If the motion command is aborted, reset bit CB.2 [▸ 74] to 0_{bin}.
8. During the active motion command enter the new parameters in registers R37, R38, R39, R40 and R58 via register communication, if they are to be modified.
9. The terminal now signals reaching of the target position by setting bit SW.3 [▸ 77] of the status word to 1_{bin}.
10. Deleting bit CB.2 [▸ 74] in the control byte completes the first motion command.
11. The terminal acknowledges this by resetting bit SW.3 [▸ 77] to 0_{bin}. The position index is automatically incremented by one position (to position 6).
12. In the control byte set bit CB.2 [▸ 74] to 1_{bin} to start the second motion command.
13. Cancel bit CB.2 [▸ 74] in order to complete the second motion command, once the terminal has signaled that position 6 has been reached by resetting bit SW.3 [▸ 77] to 0_{bin}.
14. Set bit CB.2 [▸ 74] to 1_{bin} to start the third motion command.
15. Cancel bit CB.2 [▸ 74] in order to complete the third motion command, once the terminal has signaled that position 7 has been reached by resetting bit SW.3 [▸ 77] to 0_{bin}.
16. Enter the start command 0x0513 [▸ 83] (0x5100 + offset 3) in register R7 [▸ 82] via register communication, in order to set the internal position index to position 3 as the next target position.
17. Set bit CB.2 [▸ 74] to 1_{bin} to start the fourth motion command.
18. Cancel bit CB.2 [▸ 74] in order to complete the fourth motion command, once the terminal has signaled that position 3 has been reached by setting bit SW.3 [▸ 77]. The position index is automatically incremented by one position (to position 2).
19. Set bit CB.2 [▸ 74] to 1_{bin} to start the fifth motion command.
20. Cancel bit CB.2 [▸ 74] in order to complete the fifth motion command, once the terminal has signaled that position 2 has been reached by setting bit SW.3 [▸ 77].
21. Enter the Stop command 0x0540 [▸ 83] in register R7 [▸ 82] via register communication to stop the 'multiple' motion command.

2.11.4.5 Auto-start function

The auto-start function is activated by setting bit RP0.R32.0 [▸ 84] in the feature register.

This function offers the user the possibility to carry out fast and simple positioning, independent of the PLC. Digital input 1 is used as preselection for the set position value. Like for a normal motion command, the user must enter the set position in registers R2 [▸ 81] and R3 [▸ 81]. Enable is then affected by setting bit CB.4 [▸ 74] in the control byte. Depending on the state of digital input 1, the stepper motor travels to the following positions:

- the position specified in R2 and R3 if digital input 1 = High-potential (24 V),
- back to position 0 if digital input 1 = Low-potential (0 V).

The digital input may change its state during the motion. The terminal detects such a change immediately, stops the motor and affects travels in the other direction. The terminal signals reaching of the target position by setting bit SW.3 [▸ 77] of the status word.

⚠ **WARNING**	**The motor starts immediately!** It the stepper motor is between position 0 and the upper target position (specified through R2 [▸ 81] and R3 [▸ 81]), it starts immediately when bit CB.4 [▸ 74] is activated! Before the activation it is advisable to cancel the current position and set input 1 to low potential. Ensure that your system state permits this and that hazards for persons or machinery have been ruled out!

2.11.4.6 Auto-stop function

The auto-stop function is activated by setting bit RP0.R32.1 [▸ 84] in the feature register. It is available in all operating modes.

This function offers users the option to stop the motor automatically (e.g. with two limit switches). The function is activated by setting bit CB.5 [▸ 74] in the control byte. If the motor reaches one of the limit switches, which generate a rising edge at digital input 1 or 2, it is stopped with the emergency acceleration (can be configured in register RP0.R50 [▸ 86]). Inputs 1 and 2 can be configured with bits R52.14 [▸ 86] and R52.15 [▸ 86] for N/C contacts (i.e. the active state is LOW). The purpose is to ensure that the motor switches off in the event of a broken wire.

2.11.4.7 Fast-stop function

The fast-stop function is comparable to the auto-stop function, except that this function can be used by the PLC. It does not have to be explicitly registered in the feature register. The function is switched on, resulting in the motor being stopped immediately via the emergency acceleration (register R50 [▸ 86]) by setting bit CB.1 [▸ 74] in the control byte. This can be useful for an active motion command that is configured with a small deceleration, for example.

2.11.4.8 Referencing

The stepper motor terminal enables simple referencing (homing). The digital inputs are used as cam return. If only one reference input is be used, the second input can be disabled with bit RP0.52.4 [▶ 86] or RP0.52.5 [▶ 86]. The procedure is described in more detail in the sample below.

For meaningful referencing these registers can be re-parameterized:

- RP0.R40 [▶ 85]: Maximum acceleration
- RP0.R50 [▶ 86]: Emergency acceleration
- RP0.R53 [▶ 86]: Backward referencing speed $v_{ref,b}$ (backward means decreasing position values)
- RP0.R54 [▶ 86]: Forward referencing speed $v_{ref,f}$ (forward means rising position values)
- RP0.R55 [▶ 86]: Set referencing position (low-order word)
- RP0.R56 [▶ 87]: Set referencing position (high-order word)

Sample

The following sample illustrates referencing.

1. Use register communication to enter the parameters in registers RP0.R40 [▶ 85], RP0.R50 [▶ 86], RP0.R53 [▶ 86], RP0.R54 [▶ 86], RP0.R55 [▶ 86] and RP0.R56 [▶ 87].
2. In the control byte, set bit CB.0 [▶ 74] to 1_{bin} for enabling motor control.
3. Use register communication to enter the command 0x0520 [▶ 83] in register R7 [▶ 82] to prepare the reference run.
4. The process output data DataOUT [▶ 73] must now be zero for referencing to start.
5. The two digital inputs must not be active at the start of the homing, since they are linked through logical OR.
6. In the control byte set bit CB.2 [▶ 74] to 1 to start referencing. The stepper motor terminal now moves the motor in negative direction towards the limit switch with velocity $v_{ref,b}$ (in positive direction with velocity $v_{ref,b}$ if bit RP0.R52.0 [▶ 86] is set).
7. If the terminal detects that the cam is reached through a rising edge at one of the two inputs, it stops the motor with the emergency acceleration.
8. The stepper motor terminal then moves the motor in the opposite direction with velocity $v_{ref,b}$.
9. If a falling edge now appears at the controlled digital input (motor has moved away from the cam), the terminal stops the motor with the emergency acceleration.
10. The terminal now signals successful completion of referencing by setting bit SW.3 [▶ 77] in the status word. At the same time the referencing position configured in RP0.R55 [▶ 86] and RP0.R56 [▶ 87] is set as actual position.
11. Delete bit CB.2 [▶ 74] in the control byte to compete the referencing procedure.
12. The terminal acknowledges this by deleting bit SW.3 [▶ 77].

2.11.4.9 Self-adjustment

The stepper motor terminal has an option for simple, "on the fly" self-alignment. A linear axis provides a useful platform for this purpose.

A switch positioned on the path serves as triggering sensor. If it is activated, it triggers a latch event in the terminal (any latch input can be used for this purpose).

Initially, the latch value is used as reference position. If the triggering sensor is activated again and a latch value is saved, at the next standstill the terminal corrects the difference between the reference position and the value latched during service operation. The procedure is described in more detail in the sample below.

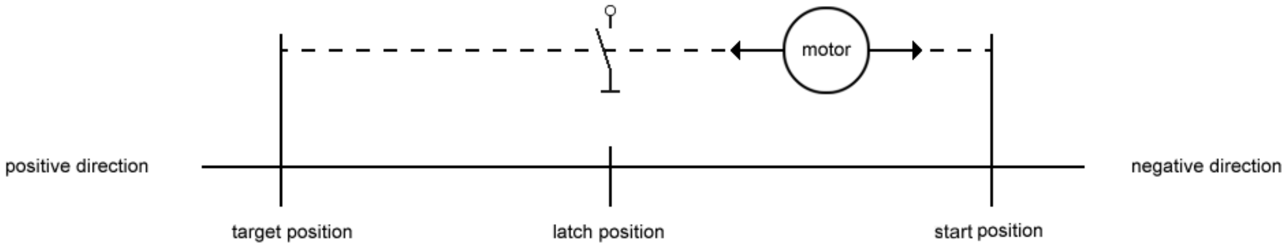


Fig. 8: Self-adjustment

Sample

The following sample illustrates self-alignment.

1. In the control byte, set bit CB.0 [▶ 74] to 1 for enabling motor control.
2. Use register communication to enter the command 0x0530 [▶ 83] in register R7, in order to activate self-adjustment.
3. Activate a latch event [▶ 21] in the control word using bits CW.0 [▶ 76] to CW.4 [▶ 76], in order to latch the reference position.
4. Enter a velocity value in the process data word DataOUT, in order to move the motor to the reference position.
5. The terminal sets bit SW.4 [▶ 77] when the reference position was passed.
6. Set the velocity in the process data word DataOUT to zero and clear the latch enable (at this point the reference position is saved).
7. Enter an opposite velocity value in the process data word DataOUT, in order to move the motor back to the home position.
8. When the home position is reached, in the control word activate the latch event that was previously used for referencing.
9. Now use process data communication to enter a velocity value in DataOUT, in order to move the motor to the latch point again.
10. Use process data communication to set the velocity to zero and clear the latch enable (the current position is now corrected).
11. Use register communication to enter the stop command 0x0540 [▶ 83] in register R7 [▶ 82], in order to activate self-adjustment.

3 Mounting and wiring

3.1 Installation on mounting rails

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

Assembly

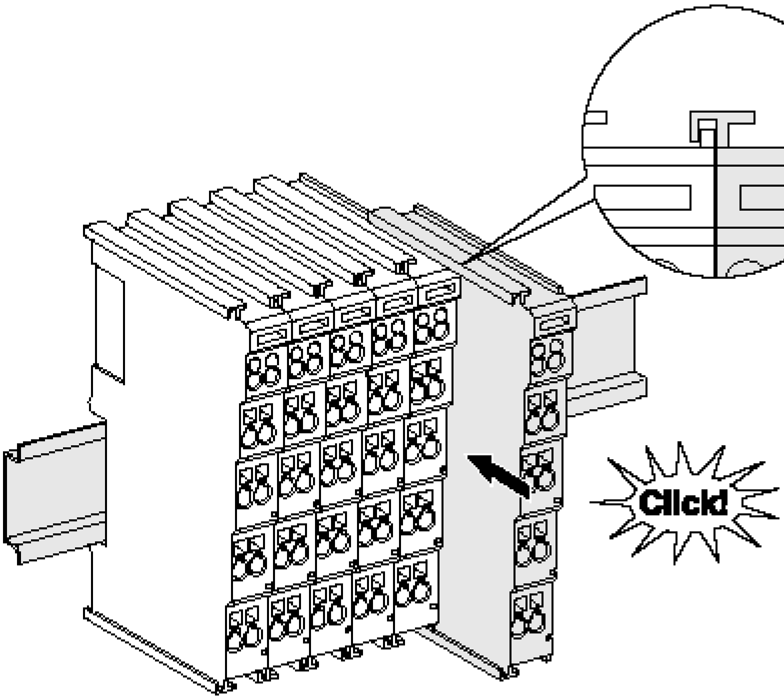


Fig. 9: Attaching on mounting rail

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

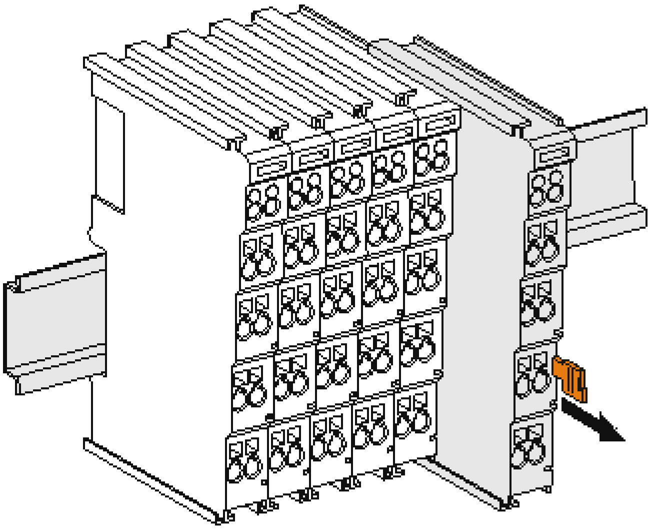


Fig. 10: Disassembling of terminal

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.

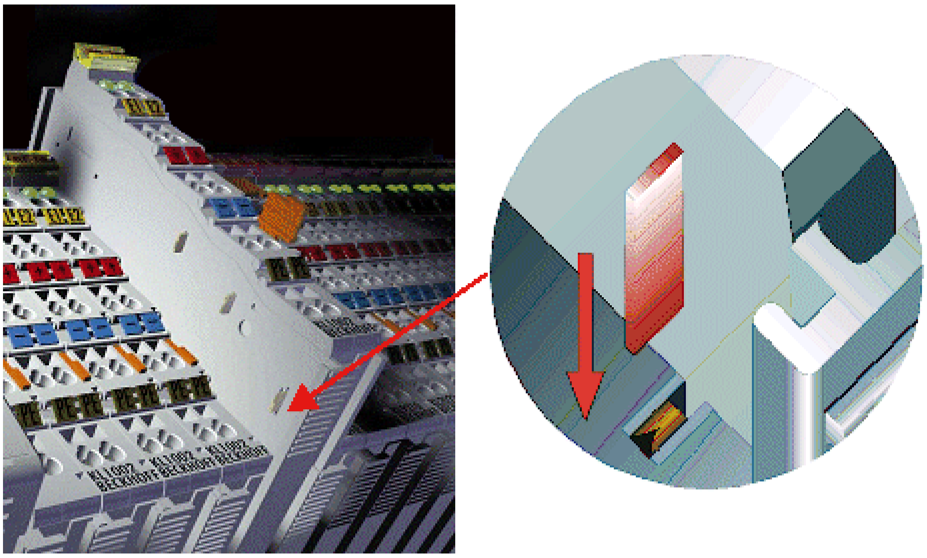


Fig. 11: *Power contact on left side*

Attention

Possible damage of the device

Note that, for reasons of electromagnetic compatibility, the PE contacts are capacitatively coupled to the mounting rail. This may lead to incorrect results during insulation testing or to damage on the terminal (e.g. disruptive discharge to the PE line during insulation testing of a consumer with a nominal voltage of 230 V). For insulation testing, disconnect the PE supply line at the Bus Coupler or the Power Feed Terminal! In order to decouple further feed points for testing, these Power Feed Terminals can be released and pulled at least 10 mm from the group of terminals.

WARNING

Risk of electric shock!

The PE power contact must not be used for other potentials!

3.2 Installation instructions for enhanced mechanical load capacity

WARNING

Risk of injury through electric shock and damage to the device!

Bring the Bus Terminal system into a safe, de-energized state before starting mounting, disassembly or wiring of the Bus Terminals!

Additional checks

The terminals have undergone the following additional tests:

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

Additional installation instructions

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

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

3.3 Installation position for operation with or without fan

❗ **Attention**	**Constraints regarding installation position and operating temperature range** When installing the terminals ensure that an adequate spacing is maintained between other components above and below the terminal in order to guarantee adequate ventilation!

Prescribed installation position for operation without fan

The prescribed installation position requires the mounting rail to be installed horizontally and the connection surfaces of the EL/KL terminals to face forward (see Fig. “*Recommended distances of installation position for operating without fan*“).

The terminals are ventilated from below, which enables optimum cooling of the electronics through convection.

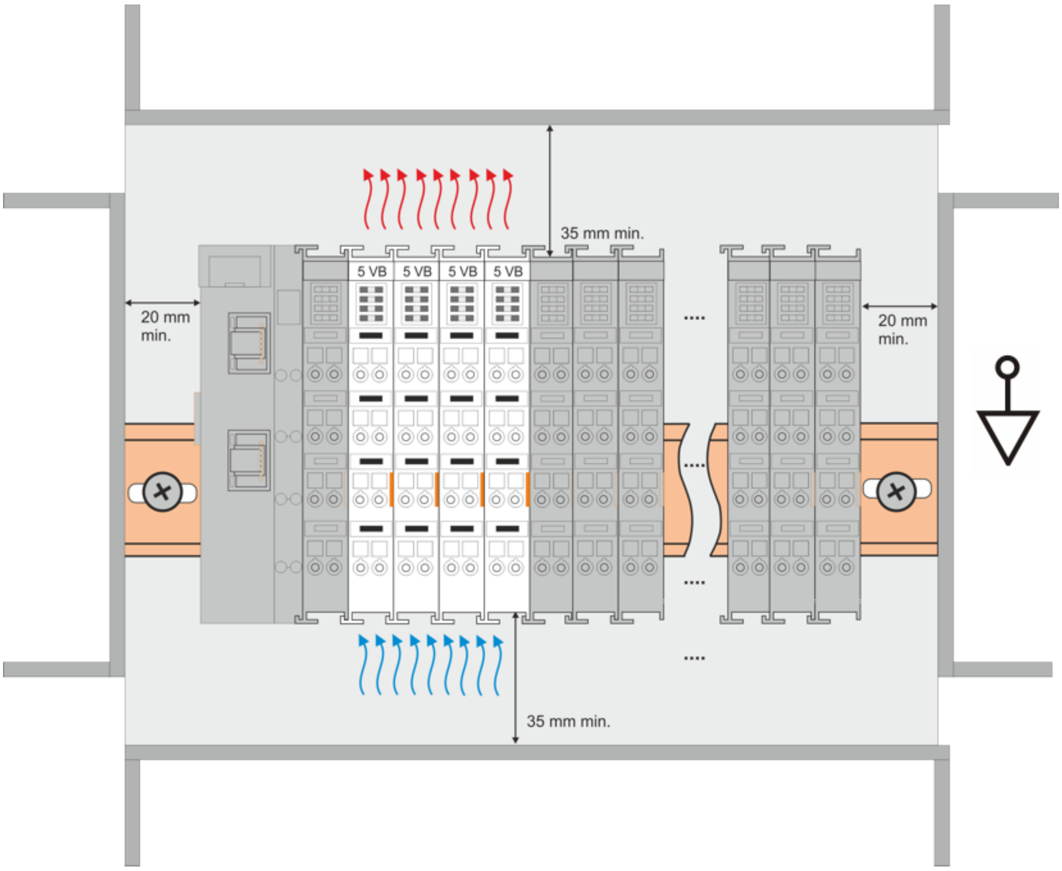


Fig. 12: *Recommended distances of installation position for operating without fan*

Compliance with the distances shown in Fig. “*Recommended distances of installation position for operating without fan*” is recommended.

For further information regarding the operation without fan refer to the Technical Data of the terminal.

Standard installation position for operation with fan

The standard installation position for operation with fan requires the mounting rail to be installed horizontally and the connection surfaces of the EL/KL terminals to face forward (see Fig. *Recommended distances for installation position for operation with fan*).
The terminals are ventilated fan supported (e.g. with fan cartridge ZB8610) from below.

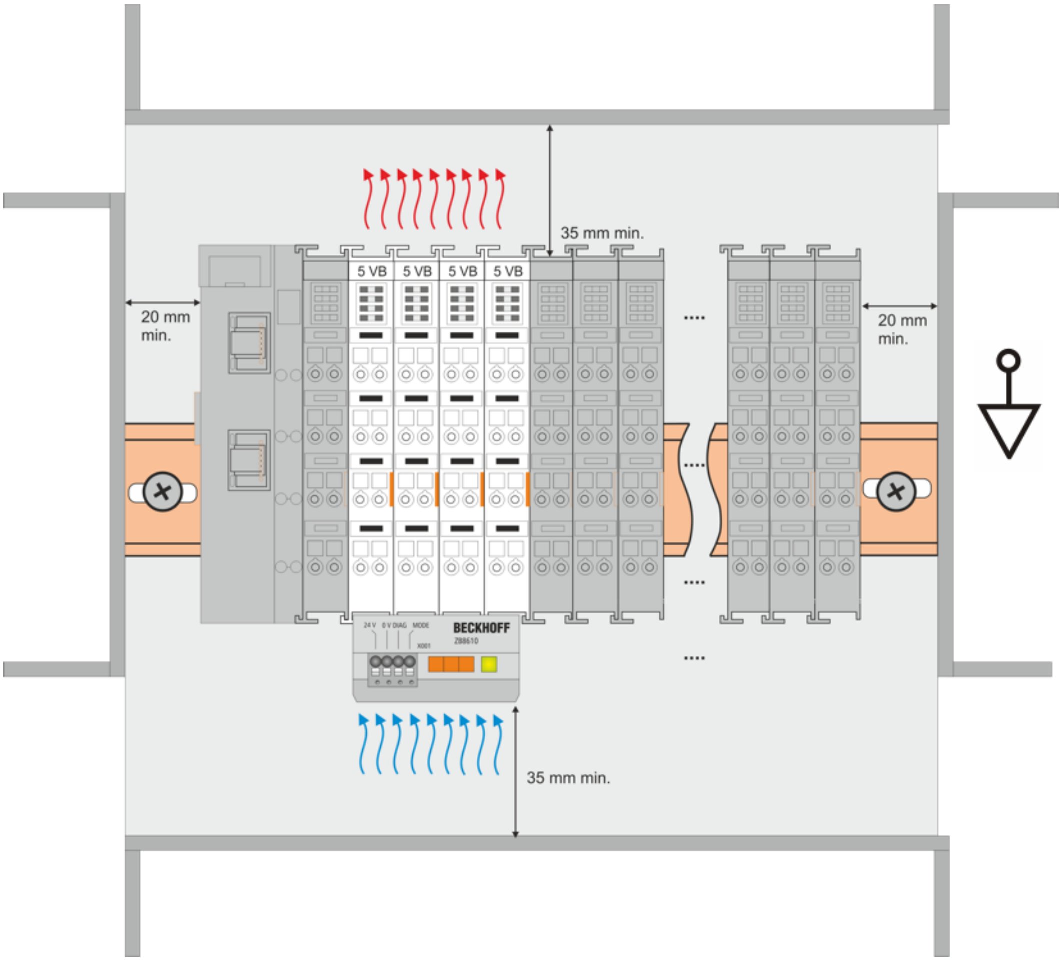


Fig. 13: *Recommended distances for installation position for operation with fan*

Other installation positions

Due to the enforced effect of the fan on the ventilation of the terminals, other installation positions (see Fig. "*Other installation positions, example 1 + 2*") may be permitted where appropriate.

See corresponding notes in the Technical Data of the terminal.

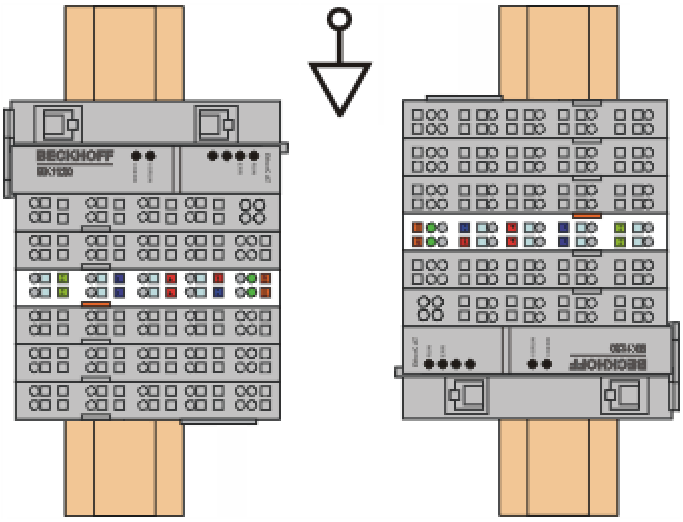


Fig. 14: *Other installation positions, example 1*

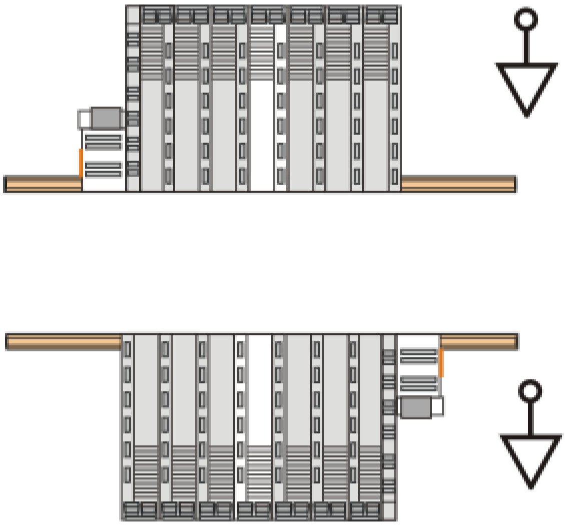


Fig. 15: *Other installation positions, example 2*

3.4 Connection

3.4.1 Connection system

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

Overview

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

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

Standard wiring (ELxxxx / KLxxxx)

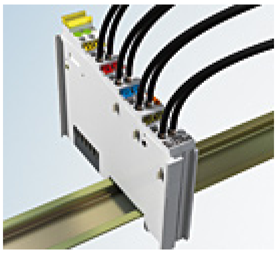


Fig. 16: Standard wiring

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

Pluggable wiring (ESxxxx / KSxxxx)

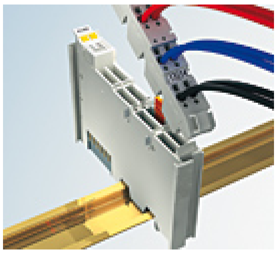


Fig. 17: Pluggable wiring

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

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

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

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

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

High Density Terminals (HD Terminals)

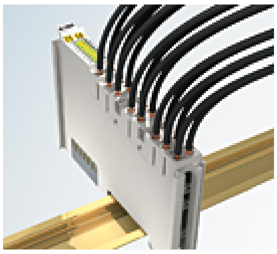


Fig. 18: *High Density Terminals*

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

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

Ultrasonically "bonded" (ultrasonically welded) conductors

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

3.4.2 Wiring

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!

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

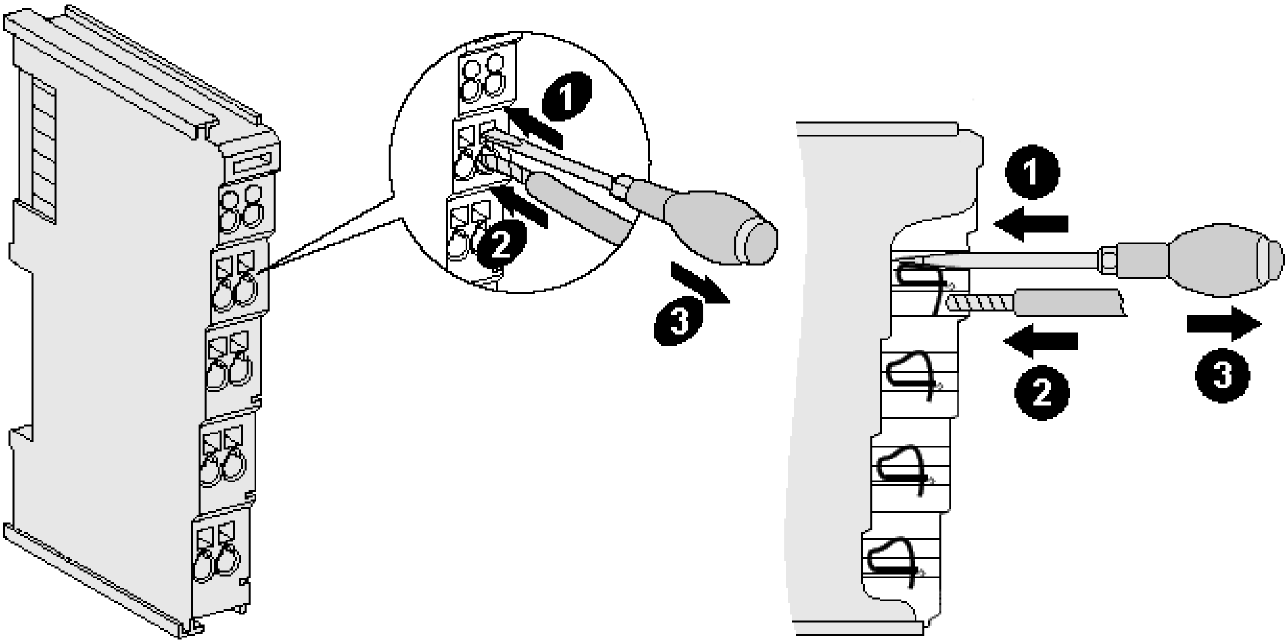


Fig. 19: Connecting a cable on a terminal point

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

1. Open a terminal point by pushing a screwdriver straight against the stop into the square opening above the terminal point. Do not turn the screwdriver or move it alternately (don't toggle).
2. The wire can now be inserted into the round terminal opening without any force.
3. The terminal point closes automatically when the pressure is released, holding the wire securely and permanently.

See the following table for the suitable wire size width.

Terminal housing	ELxxxx, KLxxxx	ESxxxx, KSxxxx
Wire size width (single core wires)	0.08 ... 2.5 mm^2	0.08 ... 2.5 mm^2
Wire size width (fine-wire conductors)	0.08 ... 2.5 mm^2	0,08 ... 2.5 mm^2
Wire size width (conductors with a wire end sleeve)	0.14 ... 1.5 mm^2	0.14 ... 1.5 mm^2
Wire stripping length	8 ... 9 mm	9 ... 10 mm

High Density Terminals (HD Terminals [▶ 42]) with 16 terminal points

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

Terminal housing	High Density Housing
Wire size width (single core wires)	0.08 ... 1.5 mm^2
Wire size width (fine-wire conductors)	0.25 ... 1.5 mm^2
Wire size width (conductors with a wire end sleeve)	0.14 ... 0.75 mm^2
Wire size width (ultrasonically “bonded" conductors)	only 1.5 mm^2
Wire stripping length	8 ... 9 mm

3.4.3 Shielding

Note

Shielding

Encoder, analog sensors and actors should always be connected with shielded, twisted paired wires.

3.5 UL notice - Compact Motion

Application

Beckhoff EtherCAT modules are intended for use with Beckhoff’s UL Listed EtherCAT System only.

Examination

For cULus examination, the Beckhoff I/O System has only been investigated for risk of fire and electrical shock (in accordance with UL508 and CSA C22.2 No. 142).

For devices with Ethernet connectors

Not for connection to telecommunication circuits.

Notes on motion devices

- *Motor overtemperature*
  Motor overtemperature sensing is not provided by the drive.
- *Application for compact motion devices*
  The modules are intended for use only within Beckhoff’s Programmable Controller system Listed in File E172151.
- *Galvanic isolation from the supply*
  The modules are intended for operation within circuits not connected directly to the supply mains (galvanically isolated from the supply, i.e. on transformer secondary).
- *Requirement for environmental conditions*
  For use in Pollution Degree 2 Environment only.

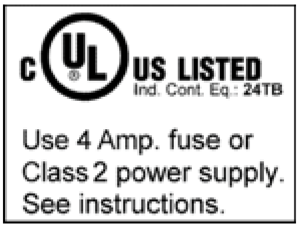
Basic principles

Two UL certificates are met in the Beckhoff EtherCAT product range, depending upon the components:

- UL certification according to UL508
  Devices with this kind of certification are marked by this sign:

Almost all current EtherCAT products (as at 2010/05) are UL certified without restrictions.

- UL certification according to UL508 with limited power consumption
  The current consumed by the device is limited to a max. possible current consumption of 4 A. Devices with this kind of certification are marked by this sign:



Almost all current EtherCAT products (as at 2010/05) are UL certified without restrictions.

Application

If terminals certified *with restrictions* are used, then the current consumption at 24 V_{DC} must be limited accordingly by means of supply

- from an isolated source protected by a fuse of max. 4A (according to UL248) or
- from a voltage supply complying with *NEC class 2*.
  A voltage source complying with *NEC class 2* may not be connected in series or parallel with another *NEC class 2* compliant voltage supply!

These requirements apply to the supply of all EtherCAT bus couplers, power adaptor terminals, Bus Terminals and their power contacts.

3.6 KL2531 - Connection

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

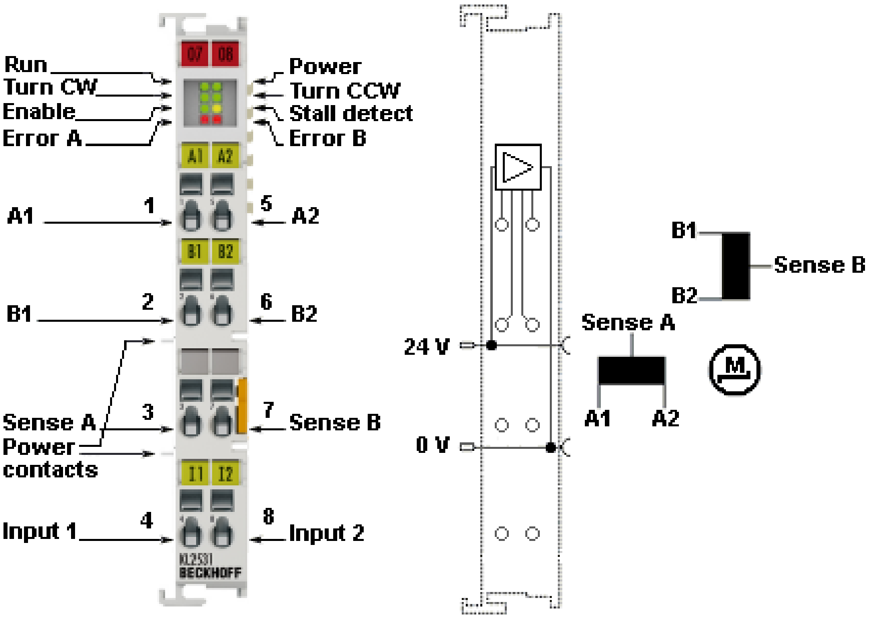


Fig. 20: KL2531-0000 - Connection

Terminal point	No.	Connection
A1	1	Motor winding A
B1	2	Motor winding B
Sense A	3	Motor winding A
Input 1	4	Digital input 1 (24 V_{DC}). The current counter value is saved as a reference mark in the latch register if bit CW.1 [▸ 76] is set in the control word and a rising edge occurs at digital input 1.
A2	5	Motor winding A
B2	6	Motor winding B
Sense B	7	Motor winding B
Input 2	8	Digital input 2 (24 V_{DC}). The current counter value is saved as a reference mark in the latch register if bit CW.2 [▸ 76] is set in the control word and a rising edge occurs at digital input 2.

Note	**Connection examples** Note the connection examples [▸ 47] for the KL2531.

3.7 KL2531 - General connection examples

WARNING

Risk of injury through electric shock and damage to the device!

Bring the Bus Terminals system into a safe, de-energized state before starting mounting, disassembly or wiring of the Bus Terminals.

Attention

Connect the motor strands correctly!

- one motor strand to clamping points A1 and A2,
- the other motor strand to clamping points B1 and B2.

Connection types

The KL2531 Stepper Motor terminal has bipolar output stages and can control bipolar and unipolar motors.

Bipolar motors

Bipolar control (serial) of a bipolar motor **Bipolar control (parallel) of a bipolar motor**

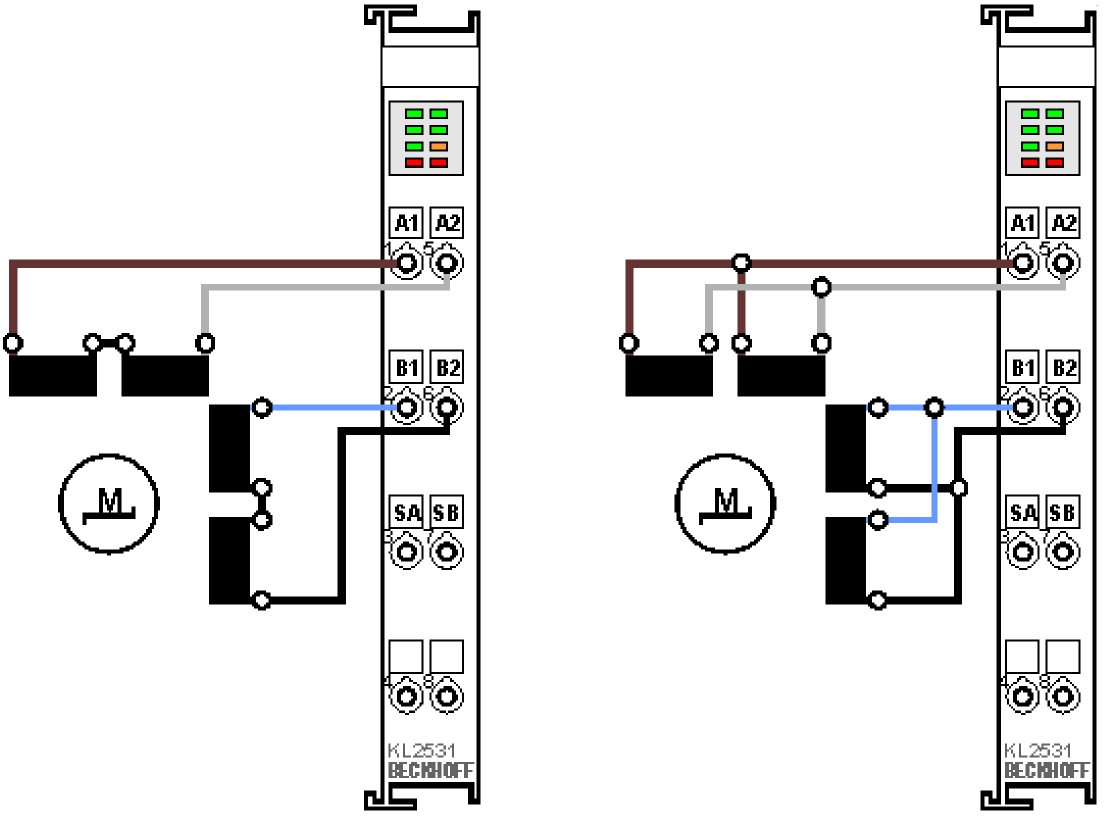


Fig. 21: KL2531 - Bipolar serial (left) and parallel (right) control of a bipolar motor

Note

Documentation for stepper motors from Beckhoff

These two examples show the connection of the bipolar Beckhoff motors AS1010, AS1020 or AS1030.
Further information on stepper motors from Beckhoff can be found in the associated documentation available for download from our website at http://www.beckhoff.com/.

Unipolar motors

Bipolar control of a unipolar motor

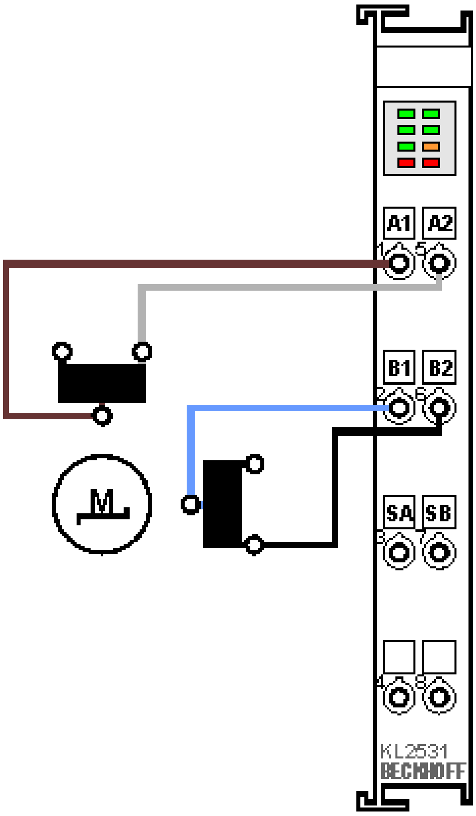


Fig. 22: KL2531 - Bipolar control of a unipolar motor

Only one half of each winding is controlled.

3.8 KL2541 - Connection

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

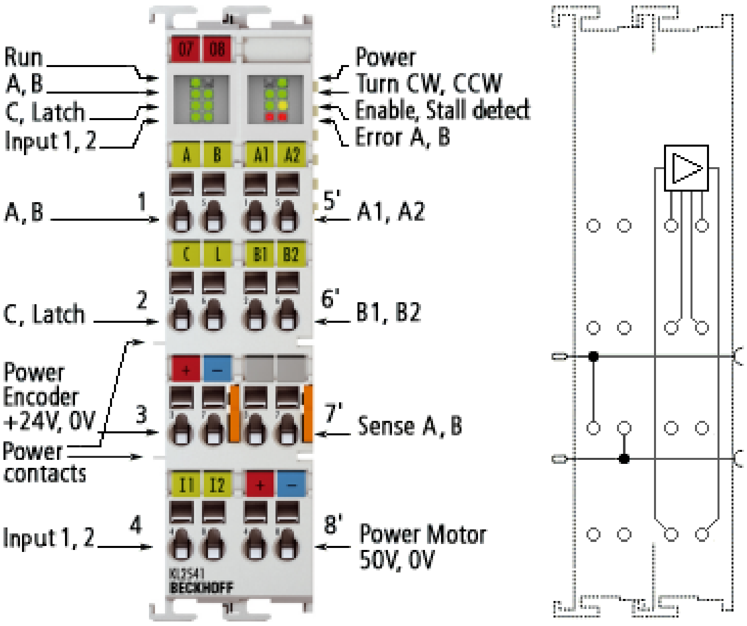


Fig. 23: KL2541-0000 - Connection

Terminal points - left part of the housing

Terminal point	No.	Connection
A	1	Encoder input A
C	2	Encoder input C (zero input). If bit CW.0 [▸ 76] is set in the control word and a rising edge occurs at encoder input C, the current counter value is stored as a reference mark in the latch register.
Encoder supply +	3	KL2541-0000: 24 V Encoder supply (from positive power contact) KL2541-0006: 5 V Encoder supply (generated from positive power contact)
Input 1	4	Digital input 1 (24 V_{DC}). The current counter value is saved as a reference mark in the latch register if bit CW.1 [▸ 76] is set in the control word and a rising edge occurs at digital input 1.
B	5	Encoder input B
Latch/Gate	6	Latch input. The current counter value is stored as a reference mark in the latch register, if in the control word • bit CW.3 [▸ 76] is set and a rising edge occurs at the latch input; or • bit CW.4 [▸ 76] is set and a falling edge occurs at the latch input. The counter can be disabled via this input if • the feature bit RP0.R32.12 [▸ 84] is set and the latch/gate input is on low potential; or • the feature bit RP0.R32.13 [▸ 84] is set and the latch/gate input is on high potential If the counter is disabled, this is reported from the terminal to the controller via status bit SW.7 [▸ 77].
Encoder supply 0 V	7	Encoder supply (from negative power contact)
Input 2	8	Digital input 2 (24 V_{DC}). The current counter value is saved as a reference mark in the latch register if bit CW.2 [▸ 76] is set in the control word and a rising edge occurs at digital input 2.

Terminal points - right part of the housing

Terminal point	No.	Connection for
A1	1'	Motor winding A
B1	2'	Motor winding B
Sense A	3'	Motor winding A
Motor supply	4'	Supply for output stages (maximum +50 V_{DC})
A2	5'	Motor winding A
B2	6'	Motor winding B
Sense B	7'	Motor winding B
Motor supply	8'	Supply for output stages (0 V_{DC})

Note

Connection examples

Note the connection examples [▶ 51] for the KL2541.

Power contacts

The power contacts (+24 V_{DC}) supply the following consumers:

- Incremental encoder (terminal points 3 and 7)
- Digital inputs (terminal points 4 and 8)
- Output driver of the stepper motor terminal

Note

Order of switch-on of the supply voltages

When the K-bus voltage is switched on, the power contacts must already be energized, so that internal circuits (output drivers) can be initialized. If this is not possible (e.g. because the supply is activated via the emergency off circuit), a software reset is required for the terminal, once the system has started up (see command 0x8000 [▶ 82]).
From firmware version 2D, any outage of the control voltage (24 V, from the power contacts) is detected and indicated by bit SW.14 [▶ 77]. Restoration of the control voltage is detected automatically, and an initialization takes place.

WARNING

The motor does not stop if the K-bus voltage fails!

If the K-bus voltage (5 V) fails, the motor controller is not reset! In other words, if the motor is in motion, it is not stopped!

3.9 KL2541 - General connection examples

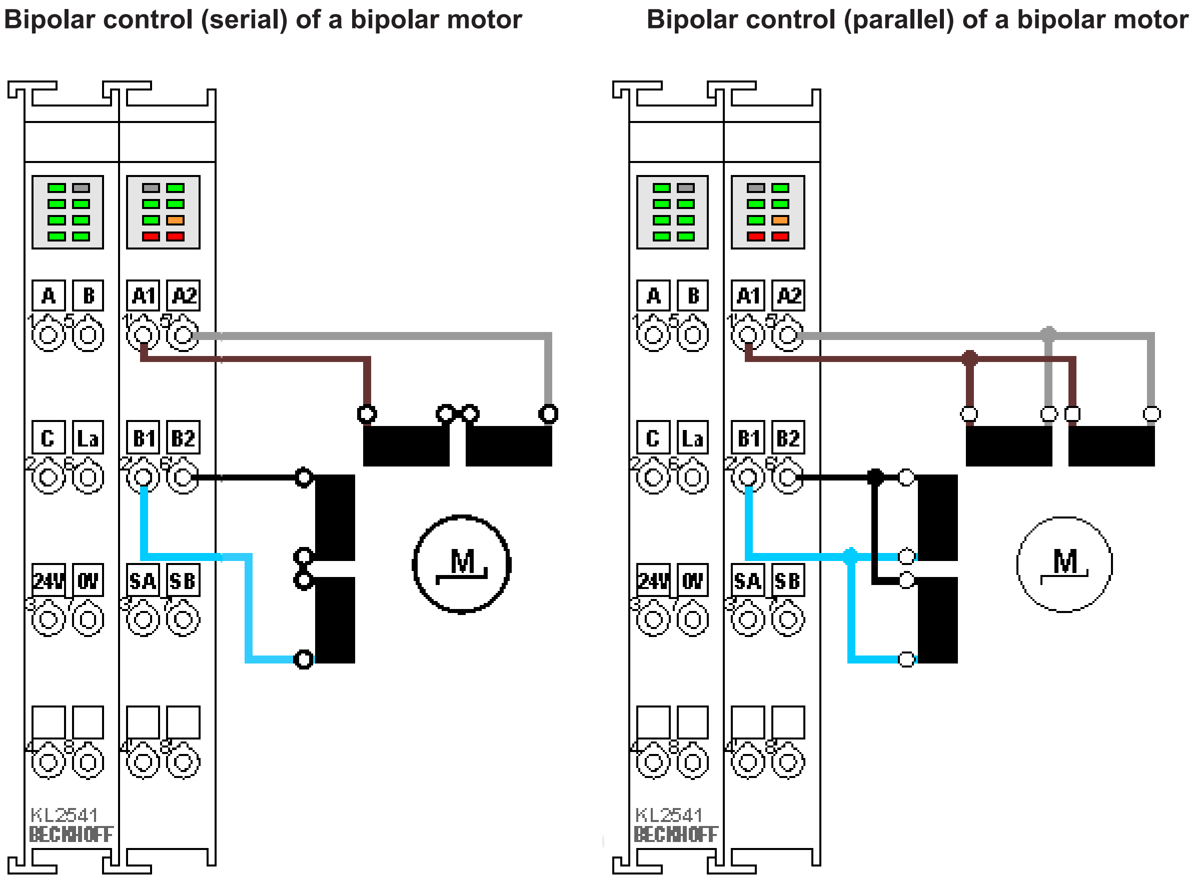
WARNING	**Risk of injury through electric shock and damage to the device!** Bring the Bus Terminals system into a safe, de-energized state before starting mounting, disassembly or wiring of the Bus Terminals.

Attention	**Connect the motor strands correctly!** • one motor strand to clamping points A1 and A2, • the other motor strand to clamping points B1 and B2. ⇨ If you connect a motor phase to terminal points of different output stages (e.g. to A1 und B1), this can damage the output stages of the stepper motor terminal!

Connection types

The KL2541 Stepper Motor terminal has bipolar output stages and can control bipolar and unipolar motors.

Bipolar motors



Bipolar control (serial) of a bipolar motor

Bipolar control (parallel) of a bipolar motor

Fig. 24: KL2541 - Bipolar serial (left) and parallel (right) control of a bipolar motor

Note	**Documentation for stepper motors from Beckhoff** These two examples show the connection of the bipolar Beckhoff motors AS1010, AS1020, AS1030, AS1050 or AS1060. Further information on stepper motors from Beckhoff can be found in the associated documentation available for download from our website at http://www.beckhoff.com/.

Unipolar motors

Bipolar control of an unipolar motor

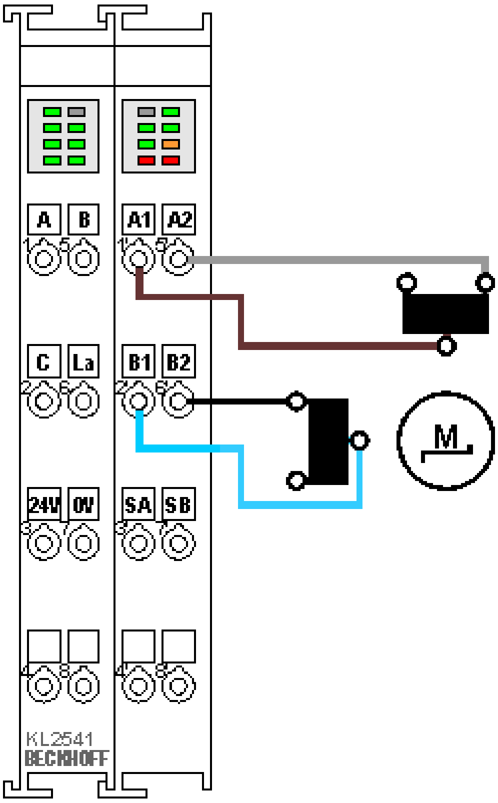


Fig. 25: KL2541 - Bipolar control of an unipolar motor

Only one half of each winding is controlled.

Encoder

Connection of an encoder (24 V)

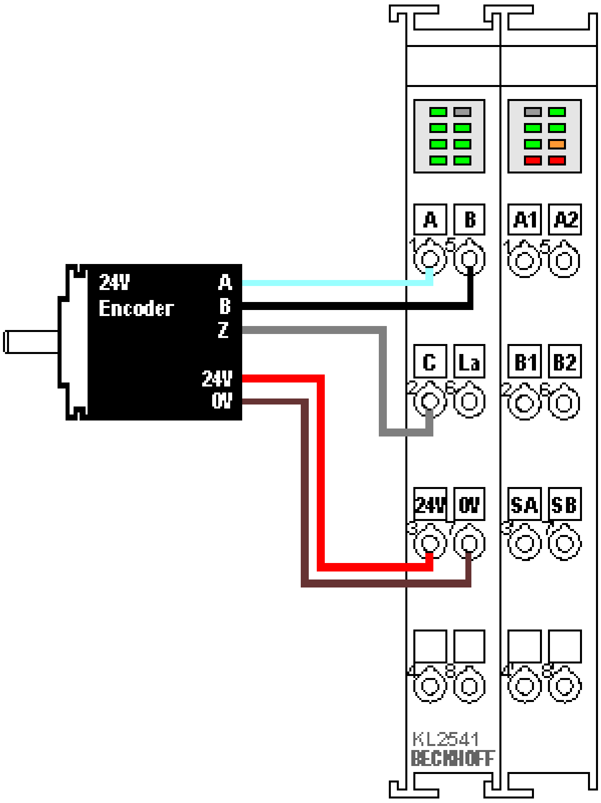


Fig. 26: KL2541 - Connecting an encoder

The encoder is supplied from the power contacts via terminal points 3 (+24 V) and 7 (0 V).

4 Configuration software KS2000

4.1 KS2000 - Introduction

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

Fig. 27: KS2000 configuration software

Configuration

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

Parameterization

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

Commissioning

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

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

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

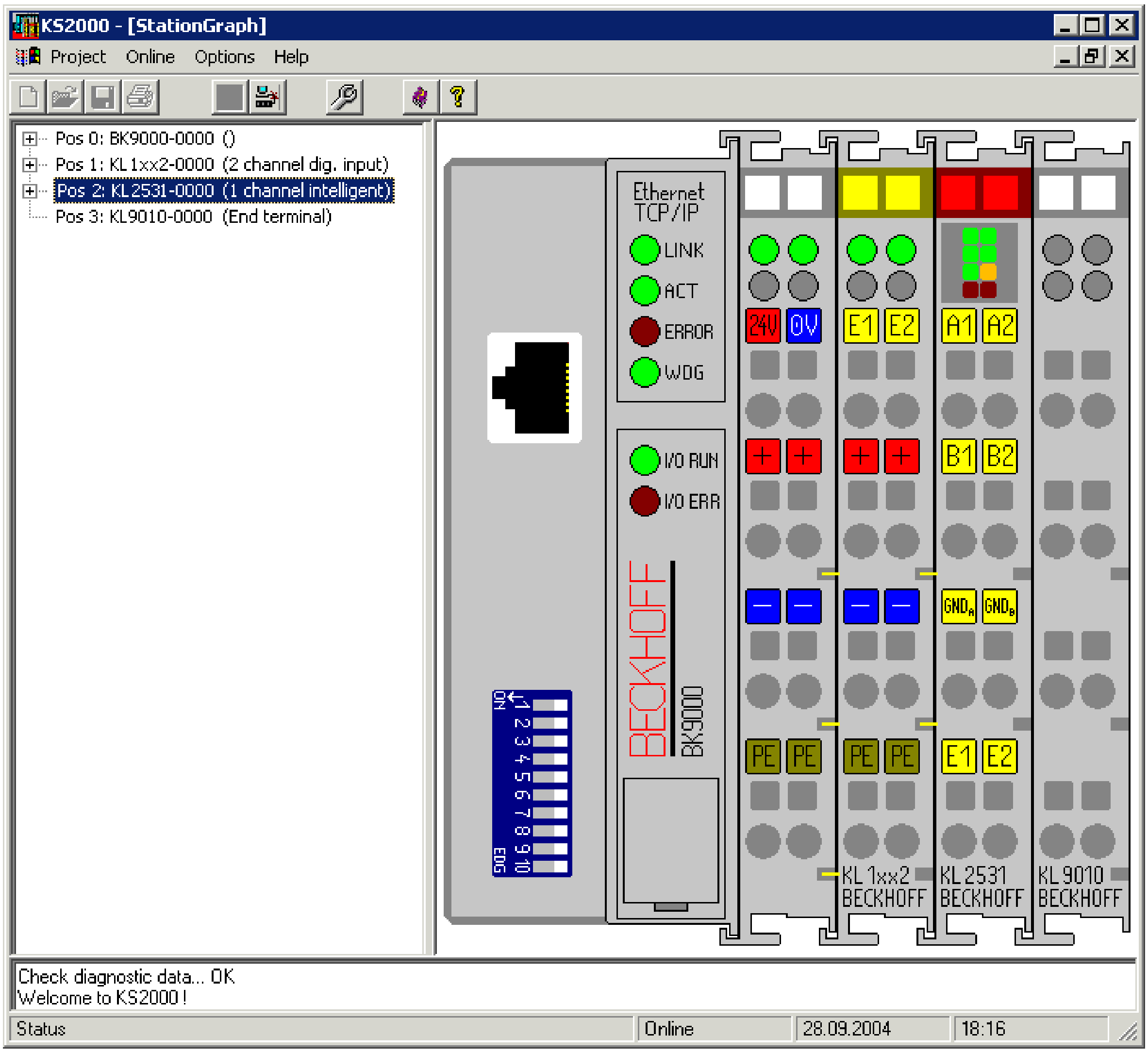
All possibilities in the *online mode* can be used in parallel with the actual fieldbus mode of the terminal station. The fieldbus protocol always has the higher priority in this case.

4.2 Parameterization with KS2000

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

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

- a BK9000 Bus Coupler for Ethernet
- a KL1xx2 digital input terminal
- a single-channel KL2531 Stepper Motor Terminal
- a KL9010 bus end terminal



Pos 0: BK9000-0000 ()
Pos 1: KL1xx2-0000 (2 channel dig. input)
Pos 2: KL2531-0000 (1 channel intelligent)
Pos 3: KL9010-0000 (End terminal)

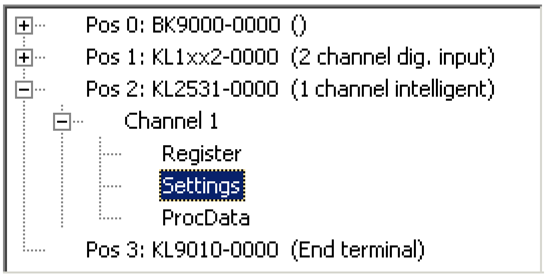
Check diagnostic data... OK
Welcome to KS2000 !

Status | Online | 28.09.2004 | 18:16

Fig. 28: Display of the fieldbus station in KS2000

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

In the tree structure of the left-hand window, click on the plus-sign next to the terminal whose parameters you wish to change (item 2 in the example).



Pos 0: BK9000-0000 ()
Pos 1: KL1xx2-0000 (2 channel dig. input)
Pos 2: KL2531-0000 (1 channel intelligent)
Channel 1
Register
Settings
ProcData
Pos 3: KL9010-0000 (End terminal)

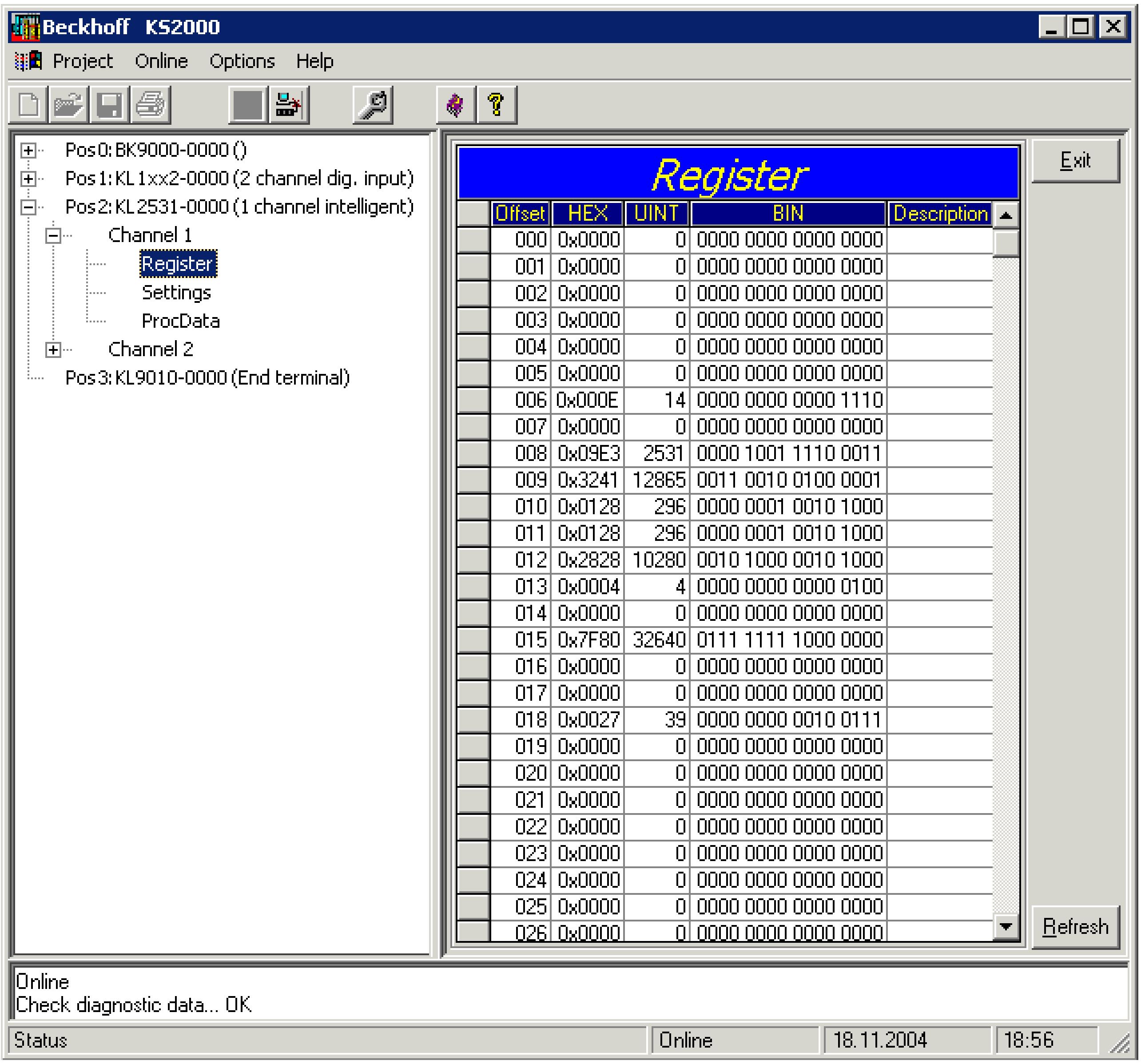
Fig. 29: KS2000 tree branches for channel 1 of the KL2531

For the KL2531 or KL2541, the branches *Register*, *Settings* and *ProcData* are displayed:

- Register [▸ 57] permits direct access to the registers of the KL2531/KL2541.
- Under Settings [▸ 58] you find dialog boxes for parameterizing the KL2531/KL2541.
- ProcData [▸ 71] displays the KL2531/KL2541 process data.

4.3 Register

You can access the registers of the KL2531/KL2541 directly under *Register*. The meaning of the register is explained in the register overview [▶ 78].



Pos0: BK9000-0000 ()
Pos1: KL1xx2-0000 (2 channel dig. input)
Pos2: KL2531-0000 (1 channel intelligent)
Channel 1
Register
Settings
ProcData
Channel 2
Pos3: KL9010-0000 (End terminal)

Register

Offset	HEX	UINT	BIN	Description
000	0x0000	0	0000 0000 0000 0000	
001	0x0000	0	0000 0000 0000 0000	
002	0x0000	0	0000 0000 0000 0000	
003	0x0000	0	0000 0000 0000 0000	
004	0x0000	0	0000 0000 0000 0000	
005	0x0000	0	0000 0000 0000 0000	
006	0x000E	14	0000 0000 0000 1110	
007	0x0000	0	0000 0000 0000 0000	
008	0x09E3	2531	0000 1001 1110 0011	
009	0x3241	12865	0011 0010 0100 0001	
010	0x0128	296	0000 0001 0010 1000	
011	0x0128	296	0000 0001 0010 1000	
012	0x2828	10280	0010 1000 0010 1000	
013	0x0004	4	0000 0000 0000 0100	
014	0x0000	0	0000 0000 0000 0000	
015	0x7F80	32640	0111 1111 1000 0000	
016	0x0000	0	0000 0000 0000 0000	
017	0x0000	0	0000 0000 0000 0000	
018	0x0027	39	0000 0000 0010 0111	
019	0x0000	0	0000 0000 0000 0000	
020	0x0000	0	0000 0000 0000 0000	
021	0x0000	0	0000 0000 0000 0000	
022	0x0000	0	0000 0000 0000 0000	
023	0x0000	0	0000 0000 0000 0000	
024	0x0000	0	0000 0000 0000 0000	
025	0x0000	0	0000 0000 0000 0000	
026	0x0000	0	0000 0000 0000 0000	

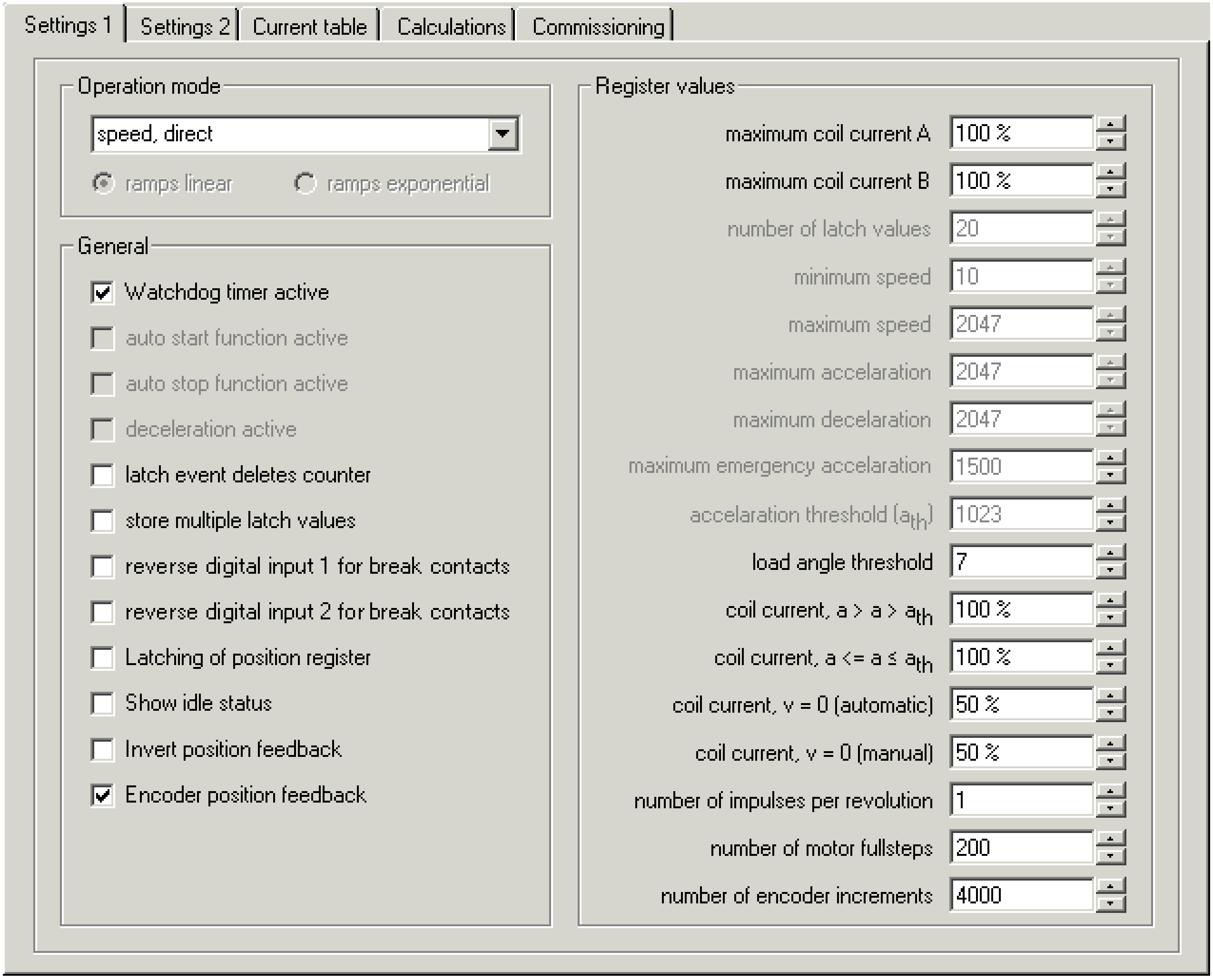
Online
Check diagnostic data... OK

Status | Online | 18.11.2004 | 18:56

Fig. 30: Register view in KS2000

4.4 Settings 1

Settings 1 tab for parametrization the KL2531 and KL2541



Settings 1 | Settings 2 | Current table | Calculations | Commissioning

Operation mode

speed, direct

● ramps linear ○ ramps exponential

General

☑ Watchdog timer active
□ auto start function active
□ auto stop function active
□ deceleration active
□ latch event deletes counter
□ store multiple latch values
□ reverse digital input 1 for break contacts
□ reverse digital input 2 for break contacts
□ Latching of position register
□ Show idle status
□ Invert position feedback
☑ Encoder position feedback

Register values

maximum coil current A	100 %
maximum coil current B	100 %
number of latch values	20
minimum speed	10
maximum speed	2047
maximum accelaration	2047
maximum decelaration	2047
maximum emergency accelaration	1500
accelaration threshold (a_{th})	1023
load angle threshold	7
coil current, a > a_{th}	100 %
coil current, a <= a_{th}	100 %
coil current, v = 0 (automatic)	50 %
coil current, v = 0 (manual)	50 %
number of impulses per revolution	1
number of motor fullsteps	200
number of encoder increments	4000

Fig. 31: Settings via KS2000

Operation mode:

- **Operating mode (** RP0.R32.4 [▶ 84], RP0.R32.3 [▶ 84]**)**
  Select the stepper motor terminal operation mode [▶ 25]:
  - Velocity, direct
  - Velocity, with ramps
  - Position tracking
  - Path control:
    - Manual
    - Motion command

  Depending on the selected operating mode, different parameters for configuring the stepper motor terminals are available under *General*, *Encoder* (see Settings 2 [▶ 61]) and *Register values*.
- **Ramps**
  In *path control* mode, you can additionally specify the ramp type (RP0.R32.5 [▶ 84]):
  - Ramps, linear
  - Ramps, exponential (during deceleration)

  This setting is only active during a motion command. In *velocity, with ramps* mode and in manual *path control* mode, acceleration is invariably based on linear ramps.

General:

- **Watchdog timer active (** RP0.R32.2 [▸ 84] **)**
  Here you can disable the watchdog (default: enabled).
  If the watchdog is active and the stepper motor terminal does not receive process data from the controller for more than 100 ms, the watchdog is triggered and the motor is stopped.
- **Auto-start function active (** RP0.R32.0 [▸ 84] **)**
  Here you can enable the auto-start function [▸ 32].
  This feature is only available in *path control* mode (default: disabled).
- **Auto-stop function active (** RP0.R32.1 [▸ 84] **)**
  Here you can enable the auto-start function [▸ 32].
  This feature is only available in *path control* mode (default: disabled).
- **Deceleration active (** RP0.R52.1 [▸ 86] **)**
  Here you can enable the deceleration ramp.
  This feature is only available in *path control* mode (default: disabled).
- **Latch event deletes counter (** RP0.R32.8 [▸ 84] **)**
  Here you can specify that the occurrence of a latch event when CW.13 [▸ 76] is active clears the counter (default: disabled).
- **Store multiple latch values (** RP0.R32.9 [▸ 84] **)**
  Here you can activate storage of multiple latch values (default: disabled).
  The number of latch values is specified through register R37 [▸ 85].
- **Invert digital input 1 for N/C contacts (** RP0.R52.14 [▸ 86] **)**
  Here you can invert digital input 1 for N/C contacts (default: disabled, configured for N/O contacts).
- **Invert digital input 2 for N/C contacts (** RP0.R52.15 [▸ 86] **)**
  Here you can invert digital input 2 for N/C contacts (default: disabled, configured for N/O contacts).
- **Latching position registers (** RP0.R52.2 [▸ 86] **)**
  Here you can enable the latch function for the position registers R0 [▸ 81] and R1 [▸ 81] (default: disabled).
- **Show idle status (** RP0.R52.3 [▸ 86] **)**
  Here you can enable the display of the idle bit in the status byte (SB.4 [▸ 74]) (default: disabled).
- **Invert position feedback (** RP0.R32.6 [▸ 84] **)**
  Here you can invert the position feedback, if the position value originates from an encoder that is mounted on a shaft opposite the stepper motor (default: disabled).
- **Position feedback with encoder (KL2541 only) (** RP0.R32.11 [▸ 84] **)**
  Here you can deactivate the KL2541 encoder input (default: enabled).
  This also enables internal position detection.

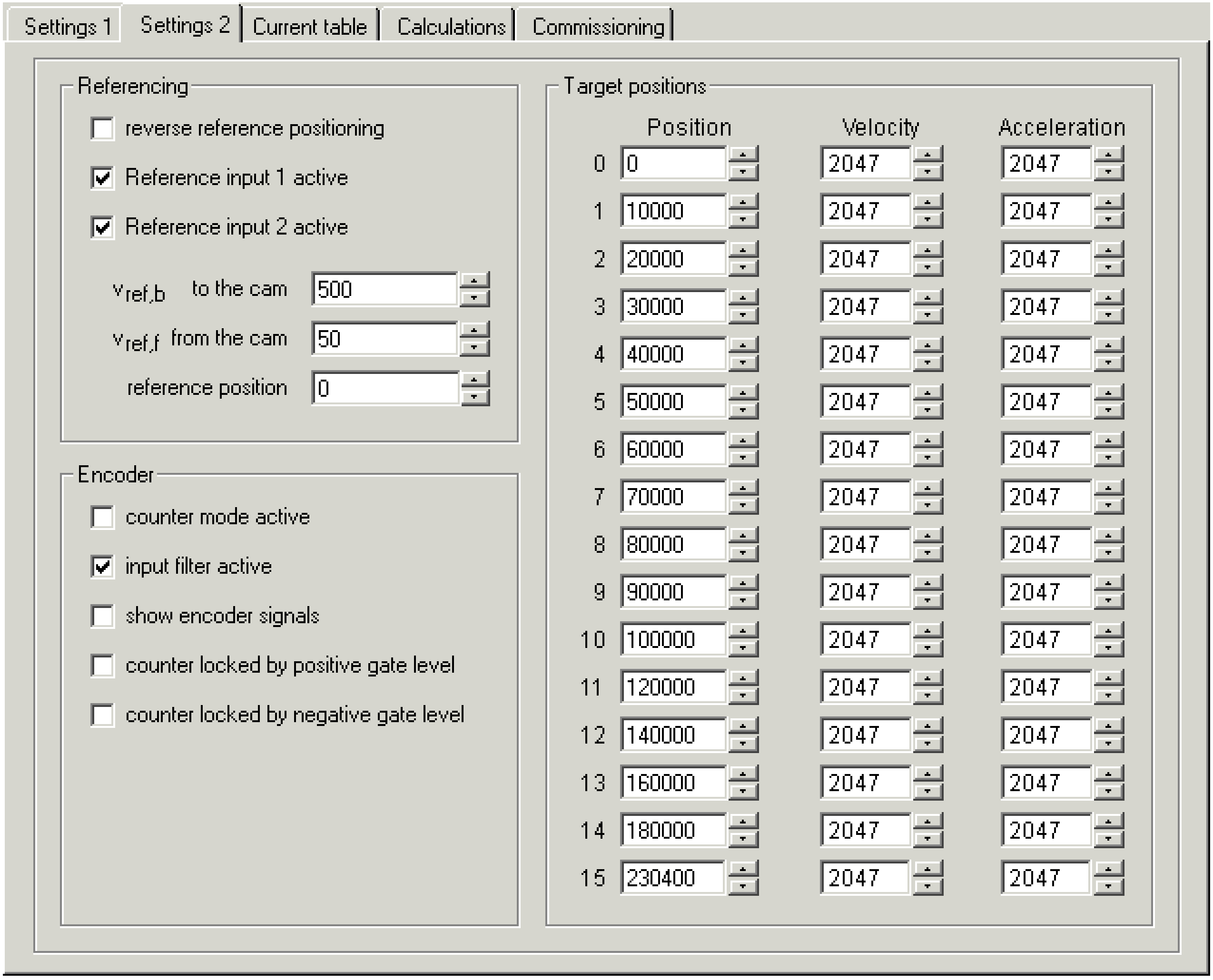
Register values:

- **Max. coil current A (** RP0.R35 [▸ 84] **)**
  Here you can specify the current limit for winding A from 0 to 100 % of rated terminal current (default: 100 %).
  KL2531-0000: 100 % = 1.5 A
  KL2541-0000: 100 % = 5.0 A
- **Max. coil current B (**RP0.R36 [▸ 84]**)**
  Here you can specify the current limit for winding B from 0 to 100 % of rated terminal current (default: 100 %).
  KL2531-0000: 100 % = 1.5 A
  KL2541-0000: 100 % = 5.0 A
- **Number of latch values (** RP0.R37 [▸ 85] **)**
  If the option store multiple latch values [▸ 59] is activated, the number of saved latch values can be specified here (default: 20, max.: 20).

- **Min. velocity v_{max} (RP0.R38 [▸ 85])**
  Here you can specify the coasting velocity for path control [▸ 58] mode (default: 10_{dec}).
  The coasting velocity is the maximum velocity from which the stepper motor can stop directly without step errors. It depends on the holding torque of the motor and mass rotating with its shaft.
- **Max. velocity v_{min}(RP0.R39 [▸ 85])**
  Here you can specify the maximum velocity for the operating modes velocity, with ramps [▸ 58] and path control [▸ 58] (default: 2047, max.: 2047).
- **Max. acceleration a_{acc} (RP0.R40 [▸ 85])**
  Here you can specify the maximum acceleration for the operating modes velocity, with ramps [▸ 58] and path control [▸ 58] (default: 2047, max.: 2047).
- **Max. deceleration a_{dec} (RP0.R58 [▸ 87])**
  Here you can specify the maximum deceleration for the operating modes velocity, with ramps [▸ 58] and path control [▸ 58] (default: 2047, max.: 2047).
- **Max. acceleration for emergency a_e (RP0.R50 [▸ 86])**
  Here you can specify the maximum emergency acceleration for path control [▸ 58] mode (default: 2047, max.: 2047).
  A rising edge of bit CB.1 [▸ 74] results in the motor being brought to a standstill with the emergency acceleration. This happens when a rising edge occurs at digital inputs 1 or 2, if the auto-stop function [▸ 32] is enabled (RP0.R32.1 [▸ 84] = 1_{bin} and CB.5 [▸ 74] = 1_{bin}).
- **Acceleration threshold a_{th} (RP0.R41 [▸ 85])**
  Here you can specify the acceleration threshold (default: 1023, max.: 2047).
- **Load angle threshold (RP0.R47 [▸ 85])**
  Here you can specify the load angle threshold from which the *Stall Detect* LED starts flashing. (default: 7, max.: 7).
  The load angle is a measure for the motor load. It is shown with values between 0 and 7, with 7 representing the maximum load.
- **Coil current for $a > a_{th}$ (RP0.R42 [▸ 85])**
  Here you can specify the coil current in 0 to 100 % for the case that the current acceleration is greater than the acceleration threshold (default: 100 %). This value refers to the set coil current of the terminal.
  a: current acceleration
  a_{th}: acceleration threshold
- **Coil current for $a \leq a_{th}$ (RP0.R43 [▸ 85])**
  Here you can specify the coil current in 0 to 100 % for the case that the current acceleration is less than or equal to the acceleration threshold (default: 100 %). This value refers to the set coil current of the terminal.
- **Coil current for v = 0 (automatic) (RP0.R44 [▸ 85])**
  Here you can specify the automatic holding current (default: 50 %).
  When the motor is at standstill, the coil currents are automatically reduced to this value, in order to prevent unnecessary heating of the motor. This value refers to the set coil current of the terminal.
- **Coil current for v = 0 (manual) (Rp0.R45 [▸ 85])**
  Here you can specify the manual holding current (default: 50 %).
  If control bit CB.3 [▸ 74] or CW.11 [▸ 76] is set, the coil currents can be reduced manually to this value in order to prevent unnecessary heating of the motor. This value refers to the set coil current of the terminal. The automatically and manually reduced coil current values are not added! The manual value has a higher priority than the automatic value!
- **Number of pulses per revolution (motor) (RP0.R51 [▸ 86])**
  Enter the number of sensor with which the motor movement is controlled (default: 1). This factor is taken into account in the calculation of the position error.
- **Full steps per revolution (motor) (RP0.R33 [▸ 84])**
  Enter the number of full steps that the connected stepper motor requires for a complete turn (default: 200).
- **Increments per revolution (encoder), KL2541 only (RP0.R34 [▸ 84])**
  Enter the number of increments issued by the encoder connected to the KL2541 during a complete turn (default: 4000).

4.5 Settings 2

Settings 2 tab for parametrization the KL2531 and KL2541



Referencing

□ reverse reference positioning

☑ Reference input 1 active

☑ Reference input 2 active

$v_{ref,b}$ to the cam: 500

$v_{ref,f}$ from the cam: 50

reference position: 0

Encoder

□ counter mode active

☑ input filter active

□ show encoder signals

□ counter locked by positive gate level

□ counter locked by negative gate level

Target positions

	Position	Velocity	Acceleration
0	0	2047	2047
1	10000	2047	2047
2	20000	2047	2047
3	30000	2047	2047
4	40000	2047	2047
5	50000	2047	2047
6	60000	2047	2047
7	70000	2047	2047
8	80000	2047	2047
9	90000	2047	2047
10	100000	2047	2047
11	120000	2047	2047
12	140000	2047	2047
13	160000	2047	2047
14	180000	2047	2047
15	230400	2047	2047

Fig. 32: Settings 2 via KS2000

Referencing:

These parameters are only available in *path control* mode.

- **Reversing the homing direction (RP0.R52.0 [▶ 86])**
  Here you can reverse the homing direction (default: disabled).
- **Reference input 1 active (RP0.R52.4 [▶ 86])**
  Here you can disable reference input 1 (default: enabled).
- **Reference input 2 active (RP0.R52.5 [▶ 86])**
  Here you can disable reference input 2 (default: enabled).
- **$v_{ref,b}$ on the cam, backward referencing velocity (RP0.R53 [▶ 86])**
  Here you specify the velocity with which the motor approaches the cam during homing (default: 500_{dec}).
  In this case backward means that the terminal controls the motor with falling position values.
- **$v_{ref,f}$ on the cam, forward referencing velocity (RP0.R54 [▶ 86])**
  Here you specify the velocity with which the motor moves away from the cam during homing (default: 50_{dec}).
  In this case forward means that the terminal controls the motor with rising position values.
- **Reference position (RP0.R55 [▶ 86] RP0.R56 [▶ 87])**
  Here you can specify the reference position for homing (default: 0).

Encoder (KL2541 only):

- **Counter mode active (RP0.R32.15 [▶ 84])**
  Here you can activate counter mode (default: disabled). If counter mode is active, the encoder input of the KL2541 acts as a 32-bit forward/backward counter (A-signal=cycle, B-signal=direction, C-signal=enable).
- **Input filter active (RP0.R32.14 [▶ 84])**
  Here you can deactivate the input filter for the encoder signals (default: active).
- **Show encoder signals (RP0.R32.10 [▶ 84])**
  For the KL2541, you can activate status word signal display as follows:
    - of the C input in bit SW.6 [▶ 77]
    - of the gate input in bit SW.7 [▶ 77]

  (default: disabled). In this case, the *LatchDataToggled* acknowledge bit for CW.6 [▶ 76] and *the EncoderDisabled* bit are not used.
- **Positive signal level at the gate blocks the counter (RP0.R32.13 [▶ 84])**
  Here you can activate disabling of the counter through a positive signal level at the KL2541 gate input (default: disabled).
- **Negative signal level at the gate blocks the counter (RP0.R32.12 [▶ 84])**
  Here you can activate disabling of the counter through a negative signal level at the KL2541 gate input (default: disabled).

Target positions:

Here you can specify 16 target positions [▶ 89] for the *path control* mode. These parameters are only available in *path control* mode.

- **Position 0 (RP2.R32 [▶ 89] RP2.R33 [▶ 89])**
  Here you can specify target position 0 (default: 0).
- **Position 1 (RP2.R34 [▶ 89] RP2.R35 [▶ 89])**
  Here you can specify target position 1 (default: 0).

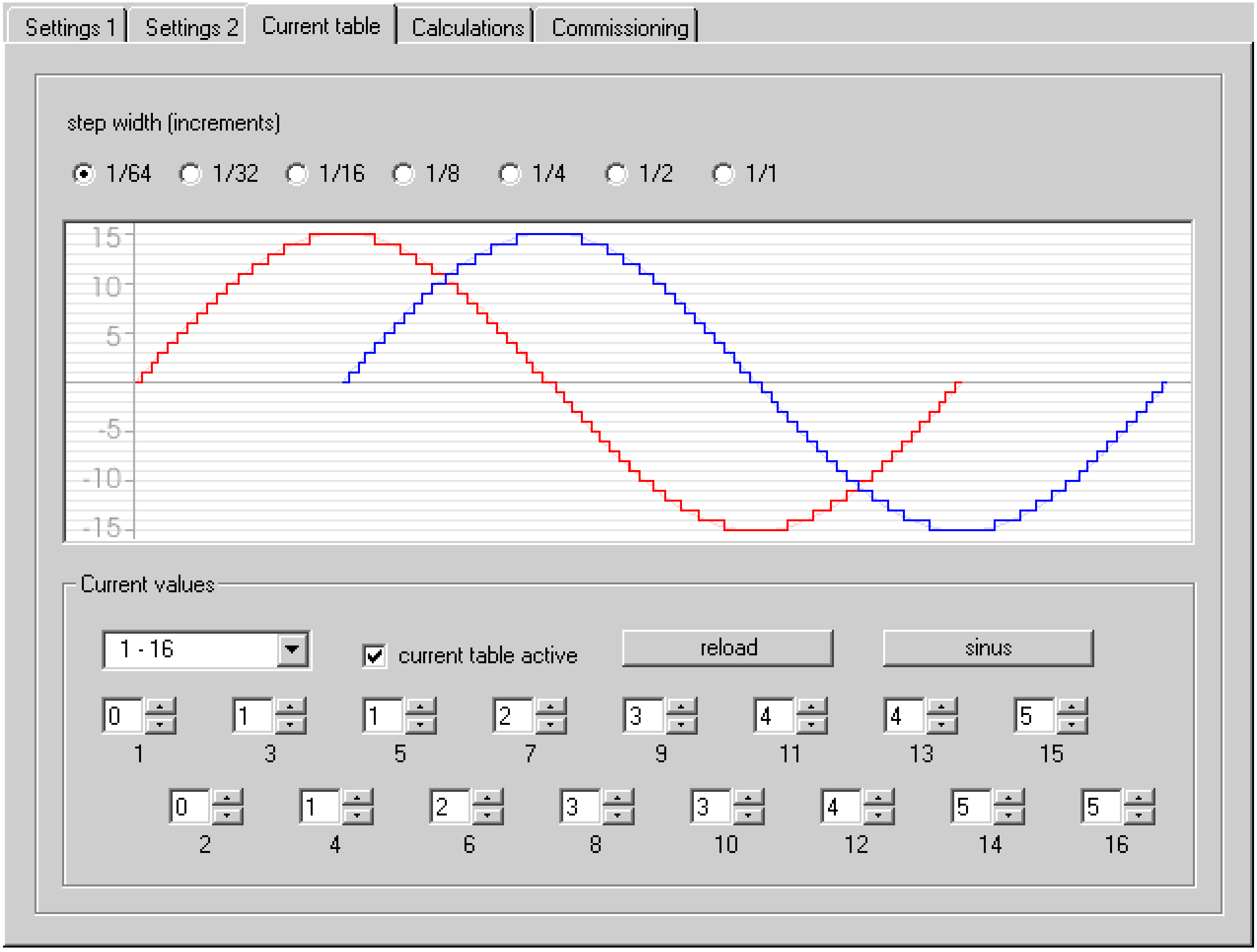
  etc.
- **Velocity 0 (RP3.R32 [▶ 90])**
  Here you can specify the velocity with which target position 0 is approached (default: 2047).
- **Velocity 1 (RP3.R33 [▶ 90])**
  Here you can specify the velocity with which target position 0 is approached (default: 2047).

  etc.
- **Acceleration 0 (RP3.R48 [▶ 91])**
  Here you can specify the acceleration that is used for approaching velocity 0 (default: 2047).
- **Acceleration 1 (RP3.R49 [▶ 91])**
  Here you can specify the acceleration that is used for approaching velocity 1 (default: 2047).

  etc.

4.6 Settings - Current table

Current table tab for parametrization the KL2531 and KL2541.



step width (increments)

● 1/64 ○ 1/32 ○ 1/16 ○ 1/8 ○ 1/4 ○ 1/2 ○ 1/1

Current values

1 - 16

☑ current table active

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	0	1	1	1	2	2	3	3	3	4	4	4	5	5	5

Fig. 33: Current table tab in KS2000

The first quarter period can be specified in up to 64 steps. For each step you can specify a current value between 0 and 15 using the check boxes below.

The current values specified for the first quarter period are automatically used for the remaining 3 quarter periods (with associated change in sign as required).

- **Step size (RP0.R46 [▶ 85])**
  Here you can select the number of steps for the quarter period (see Fig. Step size examples [▶ 65]).

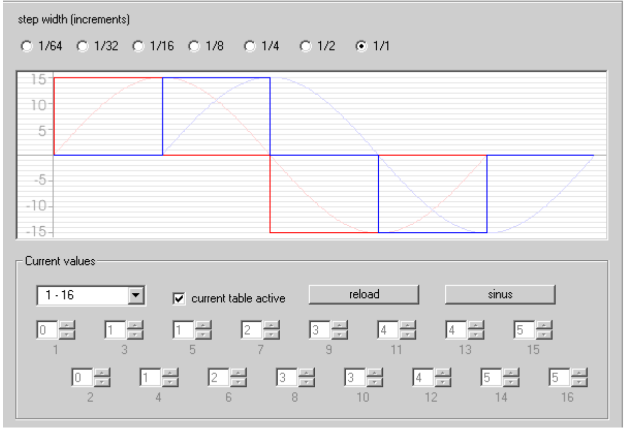
Step size	1/64	1/32	1/16	1/8	1/4	1/2 (half step)	1/1 (full step)
Steps/quarter period	64	32	16	8	4	2	1

- **Current values**
  For step size 1/64 you can specify the current values manually. They are stored in the user current table.
  For all other step sizes, a sinusoidal curve is used.

- **Manual specification of the current values**
  The current values are specified in four groups, each containing 16 current values. Select which group you wish to parameterize using the 16 check boxes below:
  1 - 16
  17 - 32
  33 - 48
  49 - 64
  The values you enter are not transferred to the terminal and stored in the user current table until you press the button.
- **Current table active (RP0.R32.7 [▶ 84])**
  With this options field you can activate the user current table instead of the internal current table.
  - The internal current table describes a sinusoidal curve. Only the step size can be specified.
  - The user current table (active if the step size 1/64 is selected) can be freely edited using the check boxes below.
- **Reload**
  Reads the values (RP1.R32 [▶ 88] to RP1.R47 [▶ 88]) stored in the terminal and displays them.
- **Sine**
  Specifies a sinusoidal curve.

Samples

Full step:



step width (increments)

○ 1/64 ○ 1/32 ○ 1/16 ○ 1/8 ○ 1/4 ○ 1/2 ● 1/1

Current values

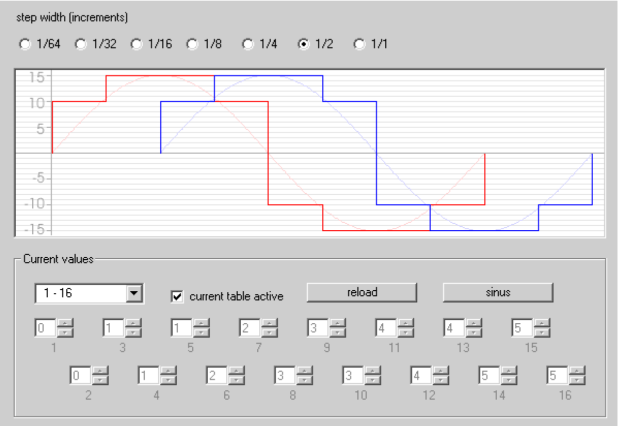
1 - 16

☑ current table active

1	3	5	7	9	11	13	15
0	1	1	2	3	4	4	5

2	4	6	8	10	12	14	16
0	1	2	3	3	4	5	5

Half step:



step width (increments)

○ 1/64 ○ 1/32 ○ 1/16 ○ 1/8 ○ 1/4 ● 1/2 ○ 1/1

Current values

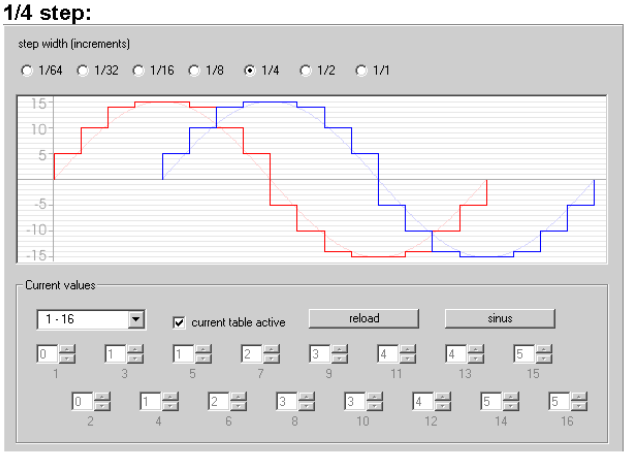
1 - 16

☑ current table active

1	3	5	7	9	11	13	15
0	1	1	2	3	4	4	5

2	4	6	8	10	12	14	16
0	1	2	3	3	4	5	5

1/4 step:



step width (increments)

○ 1/64 ○ 1/32 ○ 1/16 ○ 1/8 ● 1/4 ○ 1/2 ○ 1/1

Current values

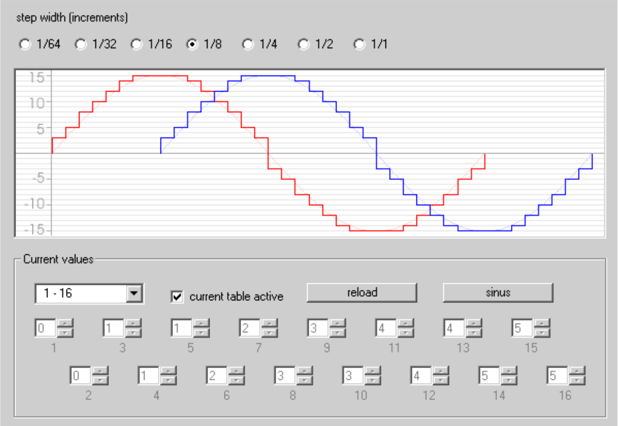
1 - 16

☑ current table active

1	3	5	7	9	11	13	15
0	1	1	2	3	4	4	5

2	4	6	8	10	12	14	16
0	1	2	3	3	4	5	5

1/8 step:



step width (increments)

○ 1/64 ○ 1/32 ○ 1/16 ● 1/8 ○ 1/4 ○ 1/2 ○ 1/1

Current values

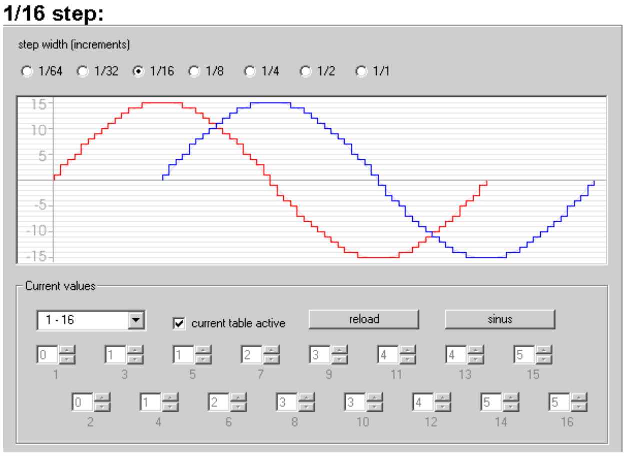
1 - 16

☑ current table active

1	3	5	7	9	11	13	15
0	1	1	2	3	4	4	5

2	4	6	8	10	12	14	16
0	1	2	3	3	4	5	5

1/16 step:



step width (increments)

○ 1/64 ○ 1/32 ● 1/16 ○ 1/8 ○ 1/4 ○ 1/2 ○ 1/1

Current values

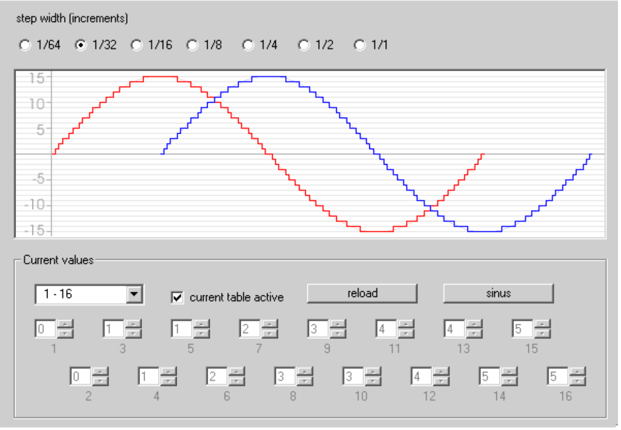
1 - 16

☑ current table active

1	3	5	7	9	11	13	15
0	1	1	2	3	4	4	5

2	4	6	8	10	12	14	16
0	1	2	3	3	4	5	5

1/32 step:



step width (increments)

○ 1/64 ● 1/32 ○ 1/16 ○ 1/8 ○ 1/4 ○ 1/2 ○ 1/1

Current values

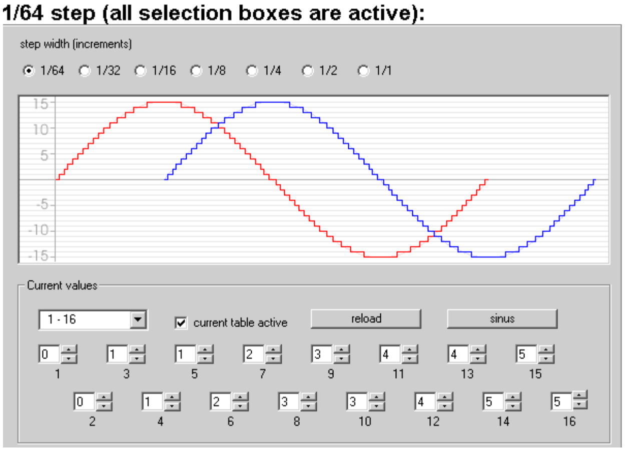
1 - 16

☑ current table active

1	3	5	7	9	11	13	15
0	1	1	2	3	4	4	5

2	4	6	8	10	12	14	16
0	1	2	3	3	4	5	5

1/64 step (all selection boxes are active):



step width (increments)

● 1/64 ○ 1/32 ○ 1/16 ○ 1/8 ○ 1/4 ○ 1/2 ○ 1/1

Current values

1 - 16

☑ current table active

1	3	5	7	9	11	13	15
0	1	1	2	3	4	4	5

2	4	6	8	10	12	14	16
0	1	2	3	3	4	5	5

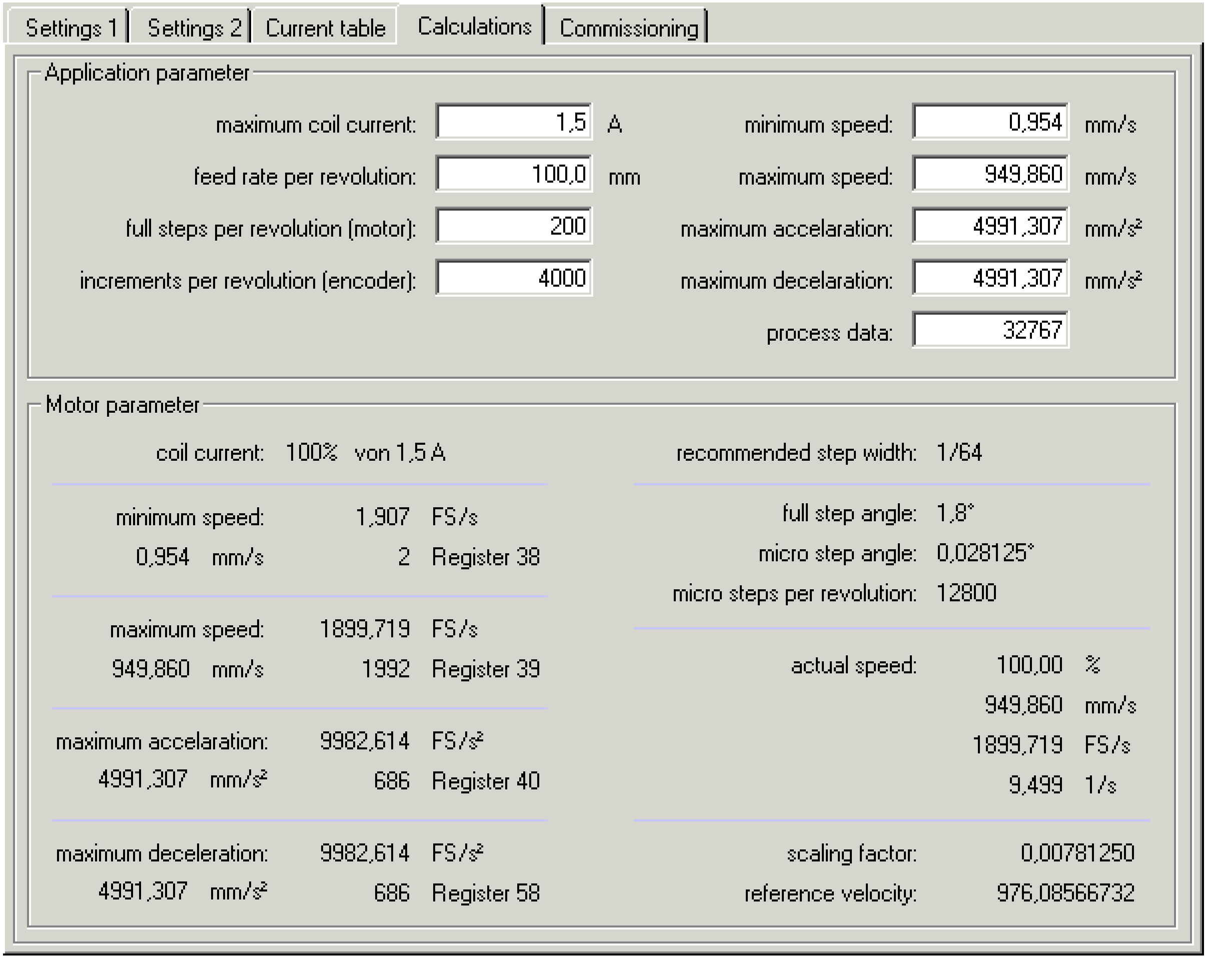
Fig. 34: KL2541 - Examples for current table settings with KS2000

4.7 Settings - Calculations

The Calculations tab is only available in the operating modes velocity, with ramps [▸ 26] and path control [▸ 28].

This dialog calculates the motor parameters from the application parameters (e.g. register values of the stepper motor terminal) and then displays them. This calculations are also available as an Excel table. To use this Excel table, macro execution has to be enabled in Excel!

Excel table (https://infosys.beckhoff.com/content/1033/kl2531_kl2541/Resources/zip/3426289803.zip)



Application parameter

maximum coil current: 1,5 A

feed rate per revolution: 100,0 mm

full steps per revolution (motor): 200

increments per revolution (encoder): 4000

minimum speed: 0,954 mm/s

maximum speed: 949,860 mm/s

maximum accelaration: 4991,307 mm/s²

maximum decelaration: 4991,307 mm/s²

process data: 32767

Motor parameter

coil current: 100% von 1,5 A

minimum speed: 1,907 FS/s
0,954 mm/s 2 Register 38

maximum speed: 1899,719 FS/s
949,860 mm/s 1992 Register 39

maximum accelaration: 9982,614 FS/s²
4991,307 mm/s² 686 Register 40

maximum deceleration: 9982,614 FS/s²
4991,307 mm/s² 686 Register 58

recommended step width: 1/64

full step angle: 1,8°
micro step angle: 0,028125°
micro steps per revolution: 12800

actual speed: 100,00 %
949,860 mm/s
1899,719 FS/s
9,499 1/s

scaling factor: 0,00781250
reference velocity: 976,08566732

Fig. 35: *Calculations* tab

Application parameters:

- **Maximum coil current (**RP0.R35 [▸ 84]**)**
  Current limit for winding A and B. The preset value is calculated from the maximum coil current A (in %) entered in the Settings [▸ 58] dialog and the maximum rated current of the stepper motor terminal.
- **Feed per revolution**
  Enter a value (parameter relating to the connected mechanical system) that is then used for calculating the motor parameters. This value is not transferred to the stepper motor terminal and is only used for illustrating the motor parameters.
- **Full steps per revolution (motor) (**RP0.R33 [▸ 84]**)**
  The value from the Settings [▸ 58] dialog is preset.
- **Increments per revolution (encoder), KL2541 only (**RP0.R34 [▸ 84]**)**
  The value from the Settings [▸ 58] dialog is preset.
- **Min. velocity (**R38 [▸ 85]**)**
  The value from the Settings [▸ 58] dialog is preset. This value is used for *path control* mode and transferred to the Commissioning [▸ 68] dialog.

- **Max. velocity (R39 [▸ 85])**
  The value from the Settings [▸ 58] dialog is preset. This value is used for *path control* mode and transferred to the Commissioning [▸ 68] dialog.
- **Max. acceleration (R40 [▸ 85])**
  The value from the Settings [▸ 58] dialog is preset. This value is used for *path control* mode and transferred to the Commissioning [▸ 68] dialog.
- **Max. deceleration (R58 [▸ 87])**
  The value from the Settings [▸ 58] dialog is preset. This value is used for *path control* mode and transferred to the Commissioning [▸ 68] dialog.
- **Process data**
  This field can be used for entering a process data (decimal) that is then used for calculating the resulting velocity based on the specified motor parameters. This process data is not transferred to the stepper motor terminal and is only used for illustrating the motor parameters.

Motor parameter:

- **Coil current**
  Specifies the maximum coil current in percent (RP0.R35 [▸ 84] and RP0.R36 [▸ 84]) of the maximum rated current of the stepper motor terminal.
- **Min. velocity** Coasting velocity [▸ 60] (in operating mode path control)
  - as feed rate
  - in full steps per second
  - as decimal register value (R38 [▸ 85])
- **Max. velocity**
  Representation of the maximum velocity
  - as feed rate
  - in full steps per second
  - as decimal register value (R39 [▸ 85])
- **Max. acceleration**
  Representation of the maximum acceleration
  - as feed rate
  - in full steps per second squared
  - as decimal register value (R40 [▸ 85])
- **Max. deceleration**
  Representation of the maximum acceleration
  - as feed rate
  - in full steps per second squared
  - as decimal register value (R58 [▸ 87])
- **Recommended step size**
  This is the recommended value for the smallest possible step size with given application parameters. The user can increase the step size if necessary. However, please note that specifying the velocity via the process data requires a new calculation.
- **Full step angle**
  Specifies the resulting angle for a full step.
- **Microstep angle**
  Specifies the resulting angle for a microstep.
- **Microsteps per revolution**
  Specifies the number of microsteps required for a complete turn of the motor shaft.
- **Current velocity**
  Indicates the resulting velocity for value entered under process data [▸ 67] and the given application parameters.

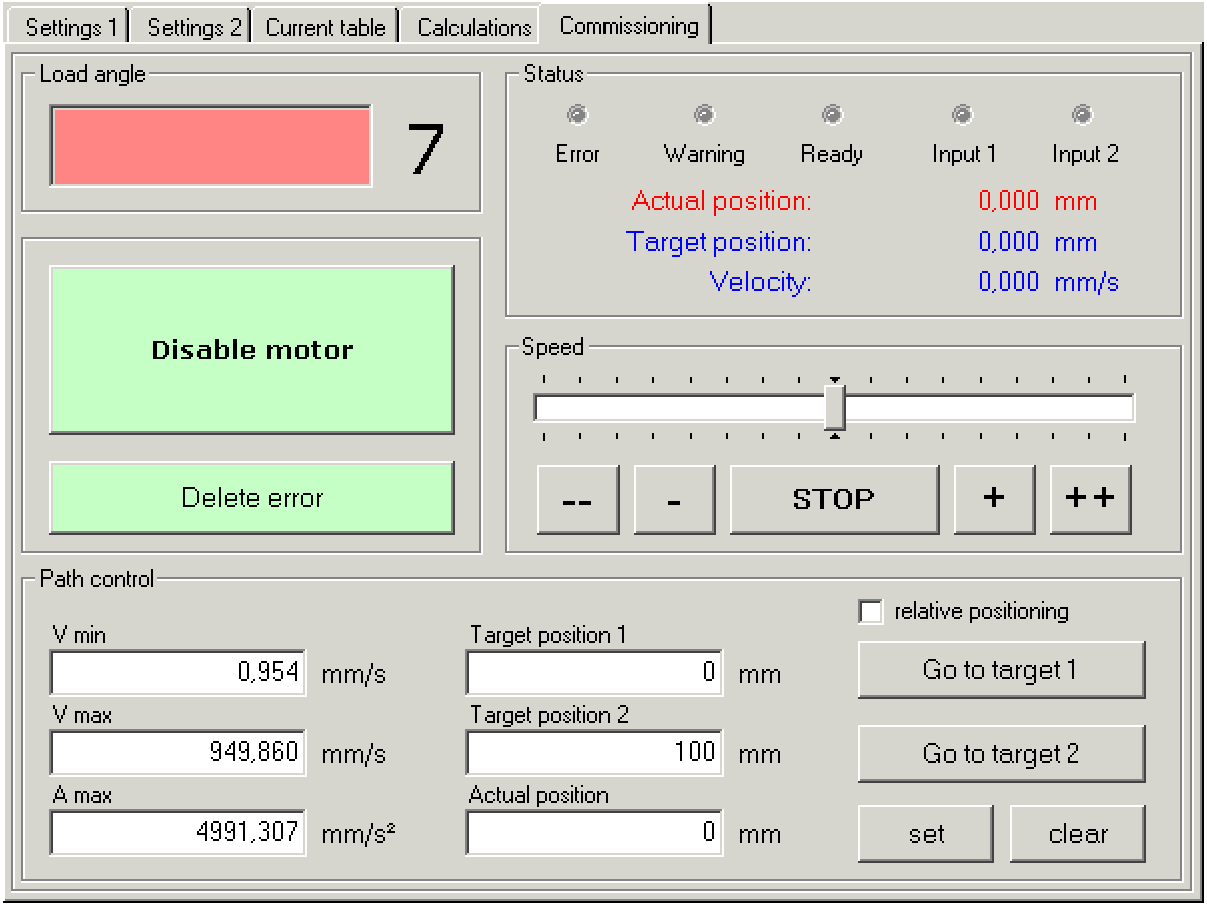
- **Scaling factor (SF)**
  Calculated based on the formula:
  SF = feed rate/revolution / (full steps/revolution x step size)
- **Reference velocity (v_{ref})**
  Calculated based on the formula:
  v_{ref} = SF x $DataOUT_{max}$ x 3.812951 = SF x 32767 x 3.812951
  (See chapter Calculation of the velocity [▸ 26])

4.8 Settings - Commissioning

Commissioning tab for parametrization the stepper motor control via the KL2531/2541.

WARNING	**Manual control of the stepper motor** The stepper motor can be controlled manually via this dialog mask. Ensure that your system state permits manual startup of the stepper motor and that hazards for persons or machinery have been ruled out!

The *Commissioning* tab is only available in path control mode.



Load angle: 7

Actual position: 0,000 mm
Target position: 0,000 mm
Velocity: 0,000 mm/s

V min: 0,954 mm/s
V max: 949,860 mm/s
A max: 4991,307 mm/s²
Target position 1: 0 mm
Target position 2: 100 mm
Actual position: 0 mm
□ relative positioning

Fig. 36: KL2541, KL2531 - *Commissioning* tab

Load angle

The load angle [▸ 18] provides information about the current mechanical load at the motor axis.

Activate / deactivate motor

This button sets or deletes the enable bit CB.0 [▸ 74], thereby enabling or disabling the motor.

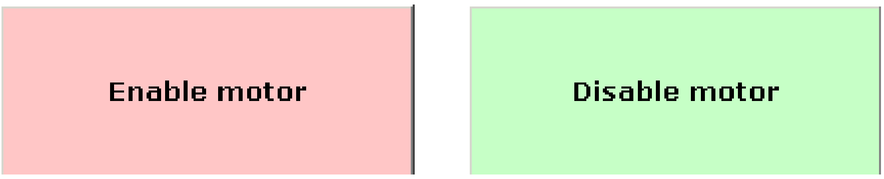


Fig. 37: *Enable motor* and *Disable motor* buttons in the *Commissioning* tab

Clear Error

If an error has occurred, the button turns red. The error can be cancelled by pressing this button.

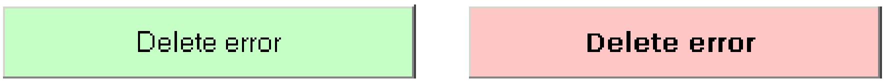
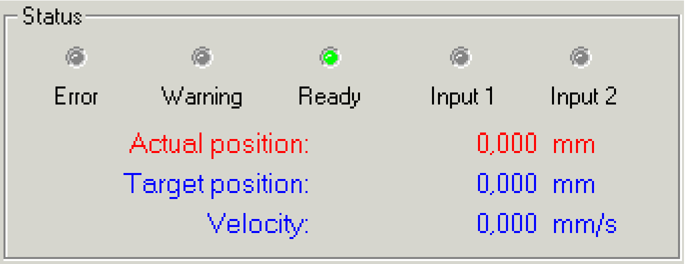


Fig. 38: *Fault reset* button in the *Commissioning* tab

Status / position

The actual position is directly coupled with the process input data DataIN [▶ 73]. The set position shows the current or last target position of the path control. The velocity is directly coupled with the process output data DataOUT [▶ 73]. The value is calculated precisely in mm or mm/s via the application parameters in the Calculations [▶ 66] tab. These values have to be specified carefully in order to ensure meaningful operation of the commissioning tool!



Actual position: 0,000 mm

Target position: 0,000 mm

Velocity: 0,000 mm/s

Fig. 39: Status and position display in the *Commissioning* tab

- **Error**
  Status bit SB.6 [▶ 74] is shown here. In the event of a fault, clicking on the fault indicator causes a message window to appear that contains plain text information about the error(s).

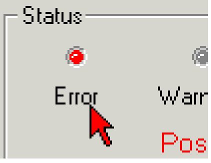


Fig. 40: *Error* status display indicator in the *Commissioning* tab

- **Warning**
  Status bit SB.5 [▶ 74] is shown here.
- **Ready**
  Status bit SB.0 [▶ 74] is shown here.
- **Input 1 / Input 2**
  The status bits SW.0 [▶ 77] and SW.1 [▶ 77] are shown here.

Speed

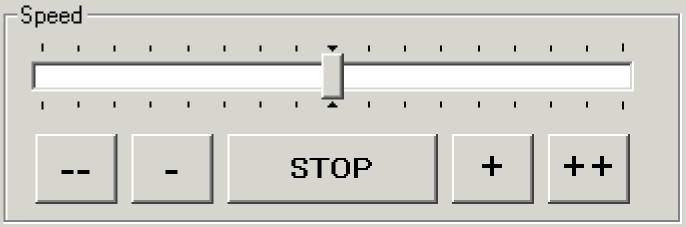


Fig. 41: Setting the velocity in the *Commissioning* tab

Via the slider you can make the motor rotate counterclockwise or clockwise. The speed increases continuously, the further the slider is moved away from the central position.
The motor can be stopped with the STOP button. The slider will then return to the central position.
The motor can be controlled manually via the --, -, + and ++ buttons. The motor rotates as long as you press and hold the button:

- - Slow counterclockwise rotation (the speed is increased every 3 seconds as long as the button is pressed)

- Very slow counterclockwise rotation

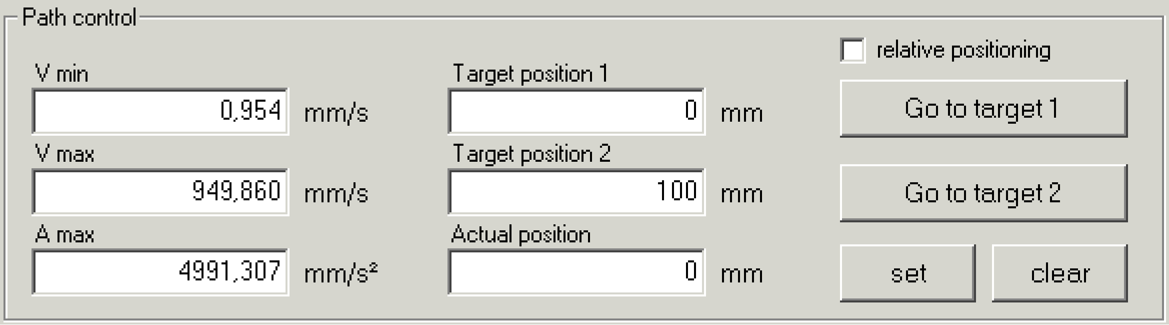
STOP The velocity is set to zero, and any active motion command is stopped

+ Very slow clockwise rotation

++ Slow clockwise rotation (the speed is increased every 3 seconds as long as the button is pressed)

Path control

The application parameters in the Calculations [▶ 66] tab are used for exact calculation in mm, mm/s or mm/s^2. These values have to be specified carefully in order to ensure meaningful operation of the commissioning tool!



Path control

V min 0,954 mm/s

V max 949,860 mm/s

A max 4991,307 mm/s²

Target position 1 0 mm

Target position 2 100 mm

Actual position 0 mm

□ relative positioning

Fig. 42: Setting the path control in the *Commissioning* tab

- **V_{min} minimum velocity (R38 [▶ 85])**
  Here you can specify the minimum value for the velocity.
  This value can be changed during operation and is entered in the RAM of the terminal.
- **V_{max} maximum velocity (R39 [▶ 85])**
  Here you can specify the maximum value for the velocity. The velocity must not exceed the value specified in the Calculations [▶ 66] tab and is therefore limited automatically!
  This value can be changed during operation and is entered in the RAM of the terminal.
- **A_{max} - maximum acceleration (R40 [▶ 85])**
  Here you can specify the maximum value for the acceleration.
  This value can be changed during operation and is entered in the RAM of the terminal.
- **Target position 1 / Target position 2**
  Here you can specify values for target positions 1 and 2. Positive or negative values are permitted.
- **Actual position**
  Here you can specify the value for the actual position. Positive or negative values are permitted.
- **Relative positioning**

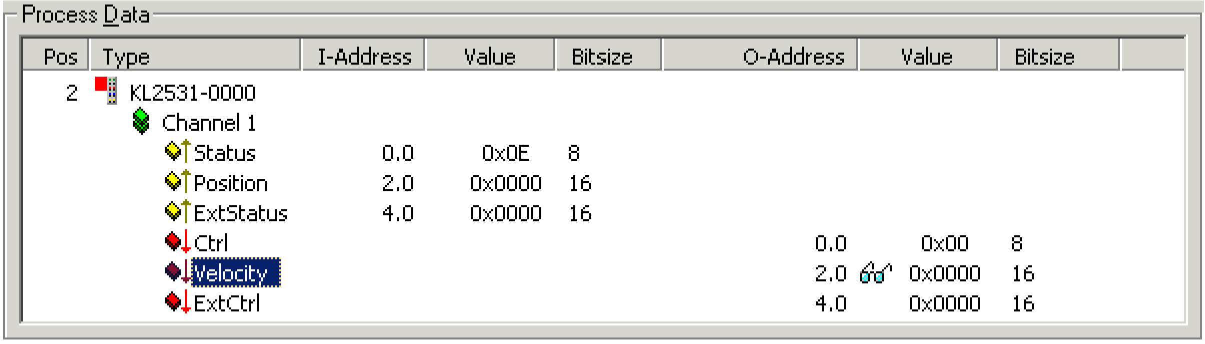
  - If the *Relative positioning* checkbox is activated, whenever a motion button is pressed the motor moves from the current position by the distance specified in the associated target position.
  - If the *Relative positioning* checkbox is not enabled, the motor travels directly to these absolute target positions when the *Start motion 1* or *Start motion 2* is pressed.

> **WARNING**
>
> **Stepper motor returns to absolute target position!**
>
> With relative positioning you can move the stepper motor repeatedly by the distance entered under target position.
> If relative positioning is switched off and the Start motion button is pressed again, the stepper motor will return to its absolute target position!

- **Start motion 1 / Start motion 2**
  The buttons start target approach 1 or target approach 2. Target positions 1 and 2 are written into registers R2 and R3 of the terminal, and the motion command is started.
- **Set**
  This button saves the value entered in actual position as the new actual position of the terminal.
- **Delete**
  This button deletes the current position and the current latch values.

4.9 Process data

The Status byte (Status), the Control byte (Ctrl) and the process data (Data) are displayed in a tree structure under *ProcData*.



Process Data

Pos	Type	I-Address	Value	Bitsize	O-Address	Value	Bitsize
2	KL2531-0000						
	Channel 1						
	Status	0.0	0x0E	8			
	Position	2.0	0x0000	16			
	ExtStatus	4.0	0x0000	16			
	Ctrl				0.0	0x00	8
	Velocity				2.0 👓	0x0000	16
	ExtCtrl				4.0	0x0000	16

Fig. 43: Process data - Display of the process data in the tree structure

The reading glasses mark the data that are currently graphically displayed in the *History* field.

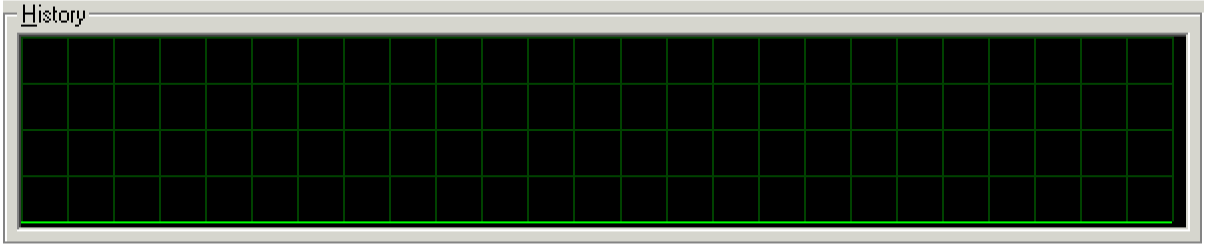
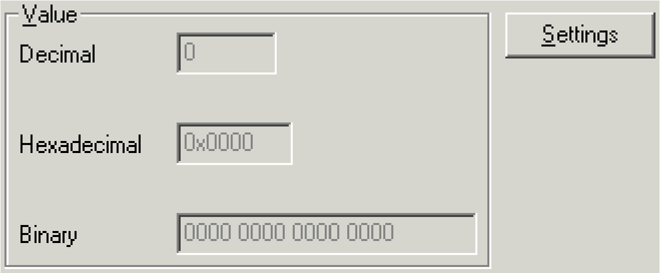


Fig. 44: Diagram - Graphic display of the data highlighted with the reading glasses in the process data

The current input values are displayed numerically in the *Value* field.



Value

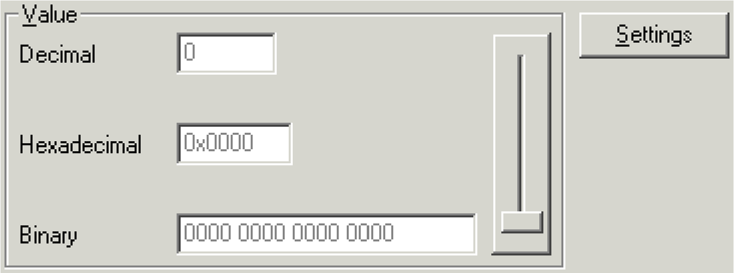
Decimal: 0

Hexadecimal: 0x0000

Binary: 0000 0000 0000 0000

Fig. 45: Value - Display of the input values

Output values can be modified through direct input or by means of the fader control.



Value

Decimal: 0

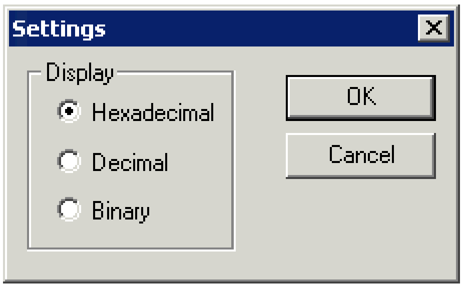
Hexadecimal: 0x0000

Binary: 0000 0000 0000 0000

Fig. 46: Value - Change in output values

⚠ **CAUTION**	**Danger for persons, the environment or devices!** Note that changing output values (forcing them) can have a direct effect on your automation application. Only modify these output values if you are certain that the state of your equipment permits it, and that there will be no risk to people or to the machine!

After pressing the *Settings* button you can set the format of the numerical display to hexadecimal, decimal or binary.



Settings

Display

● Hexadecimal

○ Decimal

○ Binary

Fig. 47: Setting the display format

5 Access from the user program

5.1 Process image

Note

No compact process image

The KL2531 and KL2541 cannot be operated with compact process image (without control and status byte), since the control and status bytes are required for process data mode of the terminals to be meaningful. Even if your Bus Coupler is set to compact process image, the KL2531 and KL2541 are represented with their complete process image!

The KL2531-0000 and KL2541-0000 are represented in the process image with at least 5 bytes of input and output data respectively. These are organized as follows:

Byte offset (without word alignment)	Byte offset (with word alignment*)	Format	Input data	Output data
0	0	Byte	SB [▸ 74]	CB [▸ 74]
1	2	Word	DataIN	DataOUT
3	4	Word	SW [▸ 77]	CW [▸ 76]

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

Legend

SB: Status byte
CB: Control byte

DataIN: Input data word (actual position)
DataOUT: Output data word (e.g. set velocity, see process data)

SW: Additional status word
CW: Additional control word

Process data

The data transferred in process data mode within the process data words DataIN and DataOUT depend on the stepper motor terminal operation mode [▸ 25].

Operation mode	DataIN	DataOUT
Velocity, direct [▸ 25]	Actual position (lower word). The full actual position (32 bit) can be read from registers R0 and R1 via register communication.	Velocity value (15 bits and the sign), see formula [▸ 25]
Velocity, with ramps [▸ 26]		Velocity value (11 bits and the sign), see formula [▸ 25]. The terminal only analyses bits 4 to 15.
Position tracking [▸ 27]		Set position (15 bits and the sign)
Travel distance control [▸ 28]		Manual operation: Velocity value (11 bits and the sign) for manual mode, see formula [▸ 25]. The terminal only analyses bits 4 to 15. In order to be able to manually specify a velocity value, no motion command must be active (CB.2 [▸ 74]=0)! Motion command: The set position (32 bits) for the motion command is entered in registers R2 and R3 via register communication. For starting the motion command DataOUT must be 0!

Note

Special types

Information on the special types KL2531-1000 and KL2541-1000 can be found in a supplementary documentation (KL2531-1000d.chm), which is available on request.

5.2 Control and status byte

Process data mode

Control byte (for process data mode)

The control byte (CB) is located in the output image [▶ 73], and is transmitted from the controller to the terminal.

Bit	CB.7	CB.6	CB.5	CB.4	CB.3	CB.2	CB.1	CB.0
Name	RegAccess	Reset	AutoStop	AutoStart	Reduced Current	Start	PLC-Stop	Enable

Legend

Bit	Name	Description	
CB.7	RegAccess	0_{bin}	Register communication off (process data mode)
CB.6	Reset	1_{bin}	all errors that may have occurred are reset by setting this bit (rising edge)
CB.5	AutoStop[1]	1_{bin}	The automatic stop function is enabled
CB.4	AutoStart[2]	1_{bin}	The automatic start function is enabled
CB.3	Reduced Current (identical to bit CW.11 [▶ 76])	1_{bin}	Reduced coil current (when the motor is at standstill) is active (specified via R45 [▶ 85])
CB.2	Start[2]	0_{bin}	Motion command is stopped
		1_{bin}	Motion command is started
CB.1	PLC-Stop[2]	1_{bin}	The motor is stopped with the emergency acceleration (R50 [▶ 86]) (rising edge)
CB.0	Enable	0_{bin}	Disable motor control
		1_{bin}	Enable motor control

[1]) Available in all operating modes except *position tracking*.
[2]) Only available in the *path control* mode.

Status byte (for process data mode)

The status byte (SB) is located in the input image [▶ 73], and is transmitted from terminal to the controller.

Bit	SB.7	SB.6	SB.5	SB.4	SB.3	SB.2	SB.1	SB.0
Name	RegAccess	Error	Warning	Idle	LoadAngle			Ready

Legend

Bit	Name	Description	
SB.7	RegAccess	0_{bin}	Acknowledgement for process data mode
SB.6	Error	0_{bin}	No error
		1_{bin}	An error has occurred (SW.13-SW.9 [▶ 77])
SB.5	Warning	1_{bin}	The internal terminal temperature is above 80 °C, or the load angle is greater than register R47 [▶ 85]
SB.4	Idle	0_{bin}	The terminal is executing an instruction (the motor is running or a motion command is active)
		1_{bin}	The terminal is without instruction (Idle)
SB.3 to SB.1	LoadAngle	Load angle (current mechanical load at the motor, depends on the maximum coil current and the current velocity)	
SB.0	Ready	0_{bin}	Motor control is disabled or an error has occurred (SB.6=1)
		1_{bin}	Motor control is enabled and no error has occurred (acknowledgement for enable, SB.6=0)

Register communication

Control byte (for register communication)

The control byte (CB) is located in the output image [▶ 73], and is transmitted from the controller to the terminal.

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

Legend

Bit	Name	Description	
CB.7	RegAccess	1_{bin}	Register communication switched on
CB.6	R/W	0_{bin}	Read access
		1_{bin}	Write access
CB.5 to CB.0	Reg. no.	• Register number: Enter the number of the register [▶ 78] that you • with input data word DataIN [▶ 73] • with output data word DataOut [▶ 73]	

Status byte (for register communication)

The status byte (SB) is located in the input image [▶ 73], and is transmitted from terminal to the controller.

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

Legend

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

5.3 Control and status word

Control word

The control word (CW) is located in the output image [▶ 73], and is transmitted from the controller to the terminal.

Bit	CW.15	CW.14	CW.13	CW.12	CW.11	CW.10	CW.9	CW.8
Name	-	Set Position in ProcData	Clear Position	Reduced Current (drive)	Reduced Current	SetPos	disMixed Decay	GetEncoder Position

Bit	CW.7	CW.6	CW.5	CW.4	CW.3	CW.2	CW.1	CW.0
Name	GetPosition Error	ToggleLatch Data	GetLatch Data	enLatch FallEdge	enLatch RiseEdge	enLatch Input2	enLatch Input1	enLatchC

Legend

Bit	Name	Description	
CW.15	-	0_{bin}	reserved
CW.14	Set position in ProcData	1_{bin}	Switches the process data from set velocity to set position (see function Specifying the set position [▶ 19])
CW.13	ClearPosition	1_{bin}	Position value is deleted (rising edge) ATTENTION: higher priority than CW.10
CW.12	Reduced current (drive)	1_{bin}	Reduced coil current (specified through R45 [▶ 85]) is active, with rotating motor
CW.11	ReducedCurrent (same function as bit CB.3 [▶ 74])	1_{bin}	Reduced coil current (when the motor is at standstill) is active (specified via R45 [▶ 85])
CW.10	SetPos	1_{bin}	Position value is set with registers R2 [▶ 81] and R3 [▶ 81] (rising edge) NOTE: only active if CW.13=0
CW.9	disMixedDecay	0_{bin}	Motor control via 2 pulsed transistors per half-wave
		1_{bin}	Motor control via 1 pulsed and one switched-through transistor per half-wave
CW.8	GetEncoder position[1, 2])	1_{bin}	Show the current encoder position in the process data
CW.7	GetPosition error[1]	0_{bin}	Show the current position in byte2,3 of the input process data
		1_{bin}	Show the relative position error [▶ 20] in byte2.3 of the input process data
CW.6	ToggleLatch data	0_{bin}	If this bit is changed, the saved latch values - up to 20 - (R37 [▶ 85]) can be retrieved if the latch array (R32.9 [▶ 84]) is activated and CW.5 is set
		1_{bin}	
CW.5	GetLatchData	0_{bin}	Show the current position in byte2,3 of the input process data
		1_{bin}	Show the current latch value [▶ 21] in byte2,3 of the input process data
CW.4	enLatchFallEdge (external)	1_{bin}	External latch event is enabled (for latch inputs with falling edge, the terminal saves the current position) ATTENTION: higher priority than CW.1 and CW.2!
CW.3	enLatchRiseEdge (external)	1_{bin}	External latch event [▶ 21] is enabled (for latch inputs with rising edge, the terminal saves the current position) ATTENTION: higher priority than CW.1, CW.2 and CW.24!
CW.2	enLatchInput2	1_{bin}	External latch event is enabled (the terminal saves the current position when a rising edge of digital input E2 is encountered)
CW.1	enLatchInput1	1_{bin}	External latch event is enabled (the terminal saves the current position when a rising edge of digital input E1 is encountered) ATTENTION: higher priority than CW.2!
CW.0	enLatchC (internal)	1_{bin}	Internal latch event is enabled (for C inputs with rising edge, the terminal saves the current position) ATTENTION: higher priority than CW.1, CW.2, CW.3 and CW.4!

[1]) Only with internal positioning
[2]) Only for KL2541

Status word

The status word (SW) is located in the input image [▶ 73], and is transmitted from terminal to the controller.

Bit	SW.15	SW.14	SW.13	SW.12	SW.11	SW.10	SW.9	SW.8
Name	Config Error	NoControl Power	OverCurrent Br B	OverCurrent Br A	OpenLoad Br B	OpenLoad Br A	Under Voltage	Over Temperature

Bit	SW.7	SW.6	SW.5	SW.4	SW.3	SW.2	SW.1	SW.0
Name	Encoder Disabled	LatchData Toggled	LatchData In-Byte2,3	LatchValid	Target Reached	SetPos Ready	Input E2	Input E1

Legend

Bit	**Name**	**Description**		
SW.15	ConfigError	1_{bin}	Configuration error (LED *Error A* flashes). Cause: • Value for motor full steps is 0 (RP0.R33 [▶ 84]), only KL2541 • Value for encoder increments is 0 (RP0.R34 [▶ 84]), only KL2541 • Value for min. velocity v_{min} is 0 (RP0.R38 [▶ 85]) • Value for max. velocity v_{max} is 0 (RP0.R39 [▶ 85]) • Value for max. acceleration a_{max} is too small (RP0.R40 [▶ 85]) • Value for coil current I_S at $a > a_{th}$ is 0 (RP0.R42 [▶ 85]) • Value for coil current I_S at $a \leq a_{th}$ is 0 (RP0.R43 [▶ 85]) • Value for max. emergency acceleration a_e is 0 (RP0.R50 [▶ 86]) The fault indication will automatically be deleted when correct data are written to the incorrectly configured registers.	
SW.14	NoControlPower	1_{bin}	Outage of the 24 V control voltage (the LED *Error B* flashes, and the terminal switches the motor off automatically). The terminal is automatically re-Initialized once the voltage has been re-established.	ATTENTION: the user must acknowledge these error messages by setting bit CB.6 [▶ 74] of control byte!
SW.13	OverCurrent Br B	1_{bin}	Overcurrent, bridge B (*error B* LED flashes, and the terminal switches the motor off automatically).	
SW.12	OverCurrent Br A	1_{bin}	Overcurrent, bridge A (*error B* LED flashes, and the terminal switches the motor off automatically).	
SW.11	OpenLoad Br B	1_{bin}	Open load, bridge B (*error B* LED flashes, and the terminal switches the motor off automatically).	
SW.10	OpenLoad Br A	1_{bin}	Open load, bridge B (*error B* LED flashes, and the terminal switches the motor off automatically).	
SW.9	UnderVoltage	1_{bin}	Supply voltage too low (*error B* LED flashes, and the terminal switches the motor off automatically).	
SW.8	OverTemperature	1_{bin}	Overtemperature (see SB.5 [▶ 74]) - the stepper controller chip temperature is above 125 °C (the LED *Error B* flashes, and the terminal switches the motor off automatically).	
SW.7	EncoderDisabled	0_{bin}	The encoder is enabled	
		1_{bin}	the encoder is locked (feature bit R32.12 [▶ 84]=1 and latch input=0 or R32.13 [▶ 84]=1 and latch input=1)	
SW.6	LatchDataToggled	0_{bin}	A new latch value is entered in byte2,3 of the input process data if bit CW.6 [▶ 76] was modified and SW.6 has taken on the new state (acknowledgement for ToggleLatchData)	
		1_{bin}		
SW.5	LatchDataIn-Byte2,3	0_{bin}	The current position is shown in byte2,3 of the input process data	
		1_{bin}	The last latch value is shown in byte2,3 of the input process data (acknowledgement for GetLatchData)	
SW.4	LatchValid	1_{bin}	a latch event has occurred (at CW.0 [▶ 76]=1, CW.3 [▶ 76]=1 or CW.4 [▶ 76]=1); if the latch array is enabled (R32.9 [▶ 84]) this bit set once the number of latch values specified in R37 [▶ 85] has been stored	
SW.3	TargetReached[3]	1_{bin}	Target position is reached	
SW.2	SetPosReady	1_{bin}	the current position was deleted (for CW.13 [▶ 76]=1) or set (for CW.10 [▶ 76]=1) (acknowledgement for ClearPos and SetPos)	
SW.1	Input 2	Status of digital input 2. The evaluation of digital input 2 can be inverted with bit RP0.R52.15 [▶ 86] of feature register 2.		
SW.0	Input 1	Status of digital input 1. The evaluation of digital input 2 can be inverted with bit RP0.R52.14 [▶ 86] of feature register 1.		

[3]) only in *path control* mode

5.4 Register overview

The registers are used for the paramétrisation of the stepper motor terminals. They can be read or written by means of register communication.

Registers 0 to 31 always have the same meaning. The content of registers 32 to 63 is specified via the register page selection register (R4).

Register no.	Comment		Default value		R/W	Memory
R0 [▶ 81]	Actual position (low-order word)		-	-	R	RAM
R1 [▶ 81]	Actual position (high-order word)		-	-	R	RAM
R2 [▶ 81]	Setting of the target position or position (low-order word)		-	-	R/W	RAM
R3 [▶ 81]	Setting of the target position or position (high-order word)		-	-	R/W	RAM
R4 [▶ 81]	Register page selection register (range: 0 to 2)		0x0000	0_{dec}	R/W	RAM
R5 [▶ 81]	Latch value (high-order word)		-	-	R	RAM
R6 [▶ 82]	Temperature register		e.g. 0x003C	e.g. 60_{dec}	R	RAM
R7 [▶ 82]	Command register		0x0000	0_{dec}	R/W	RAM
R8 [▶ 83]	Terminal type	KL2531:	0x09E3	2531_{dec}	R	ROM
		KL2541:	0x09ED	2541_{dec}		
R9 [▶ 83]	Firmware version		e.g. 0x3141	e.g. $1A_{ASCII}$	R	ROM
R10	Multiplex shift register		0x0128	296_{dec}	R	ROM
R11	Signal channels		0x0128	296_{dec}	R	ROM
R12	Minimum data length		0x2828	10260_{dec}	R	ROM
R13	Data structure		0x0004	4_{dec}	R	ROM
R14	reserved		-	-	-	-
R15	Alignment register		0x7F80	32640	R/W	RAM
R16 [▶ 83]	Hardware version number		e.g. 0x0000	e.g. 0_{dec}	R/W	EEPROM
R17	reserved		-	-	-	-
...	...		...	...	...	...
R30	reserved		-	-	-	-
R31 [▶ 83]	Code word register		0x0000	0_{dec}	R/W	RAM
R32 to R63	Register for showing register page 0 [▶ 79], register page 1 [▶ 80], register page 2 [▶ 80] or register page 3 [▶ 80]					

Register page 0 (further parameters)

Register no.	Comment	Default value		R/W	Memory
R32 [▶ 84]	Feature register 1	0x0000	0_{dec}	R/W	EEPROM
R33 [▶ 84]	Full motor steps	0x00C8	200_{dec}	R/W	EEPROM
R34 [▶ 84]	Encoder increments (KL2541 only)	0x0FA0	4000_{dec}	R/W	EEPROM
R35 [▶ 84]	Maximum coil current A	0x0064	100_{dec}	R/W	EEPROM
R36 [▶ 84]	Maximum coil current B	0x0064	100_{dec}	R/W	EEPROM
R37 [▶ 85]	Number of latch values	0x0014	20_{dec}	R/W	RAM / EEPROM
R38 [▶ 85]	Min. velocity v_{min}	0x000A	10_{dec}	R/W	RAM / EEPROM
R39 [▶ 85]	Max. velocity v_{max}	0x07FF	2047_{dec}	R/W	RAM / EEPROM
R40 [▶ 85]	max. acceleration a_{acc}	0x07FF	2047_{dec}	R/W	RAM / EEPROM
R41 [▶ 85]	Acceleration threshold a_{th}	0x03FF	1023_{dec}	R/W	EEPROM
R42 [▶ 85]	Coil current I_S, $a > a_{th}$	0x0064	100_{dec}	R/W	EEPROM
R43 [▶ 85]	Coil current I_S, $a \leq a_{th}$	0x0064	100_{dec}	R/W	EEPROM
R44 [▶ 85]	Coil current I_S, $v = 0$ (automatic)	0x0032	50_{dec}	R/W	EEPROM
R45 [▶ 85]	Coil current I_S (manual)	0x0032	50_{dec}	R/W	EEPROM
R46 [▶ 85]	Step size per quarter period	0x0006	6_{dec}	R/W	EEPROM
R47 [▶ 85]	Load angle threshold	0x0007	7_{dec}	R/W	RAM / EEPROM
R48 [▶ 85]	Kp factor	0x0000	0_{dec}	R/W	RAM / EEPROM
R49 [▶ 85]	Kv factor	0x0000	0_{dec}	R/W	RAM / EEPROM
R50 [▶ 86]	Emergency acceleration a_e	0x07FF	2047_{dec}	R/W	RAM / EEPROM
R51 [▶ 86]	Number of pulses per revolution	0x0001	1_{dec}	R/W	EEPROM
R52 [▶ 86]	Feature register 2	0x0000	0_{dec}	R/W	EEPROM
R53 [▶ 86]	Referencing velocity backward $v_{ref,b}$	0x01F4	500_{dec}	R/W	EEPROM
R54 [▶ 86]	Referencing velocity forward $v_{ref,f}$	0x0032	50_{dec}	R/W	EEPROM
R55 [▶ 86]	Referencing position (low-order word)	0x0000	0_{dec}	R/W	EEPROM
R56 [▶ 87]	Referencing position (high-order word)	0x0000	0_{dec}	R/W	EEPROM
R57 [▶ 87]	Encoder tolerance (KL2541 only)	0x0000	0_{dec}	R/W	EEPROM
R58 [▶ 87]	max. deceleration a_{dec}	0x07FF	2047_{dec}	R/W	RAM / EEPROM
R59	reserved	-	-	-	-
...	...	...	...	...	...
R61	reserved	-	-	-	-
R62 [▶ 87]	Setting the target position or position (low-order word) for extended auto-start	-	-	R/W	RAM / EEPROM
R63 [▶ 87]	Setting the target position or position (high-order word) for extended auto-start	-	-	R/W	RAM / EEPROM

Register page 1 (user current table)

Register no.	Comment	Default value		R/W	Memory
R32 [▶ 88]	Current value 4 to 1	0x1100	4352_{dec}	R/W	EEPROM
R33	Current value 8 to 5	0x3221	12833_{dec}	R/W	EEPROM
R34	Current value 12 to 9	0x4433	17459_{dec}	R/W	EEPROM
R35	Current value 16 to 13	0x5554	21844_{dec}	R/W	EEPROM
R36	Current value 20 to 17	0x7666	30310_{dec}	R/W	EEPROM
R37	Current value 24 to 21	0x8877	34935_{dec}	R/W	EEPROM
R38	Current value 28 to 25	0x9998	39320_{dec}	R/W	EEPROM
R39	Current value 32 to 29	0xAAAA	43690_{dec}	R/W	EEPROM
R40	Current value 36 to 33	0xBBBB	48059_{dec}	R/W	EEPROM
R41	Current value 40 to 37	0xCCCC	52428_{dec}	R/W	EEPROM
R42	Current value 44 to 41	0xDDDC	56796_{dec}	R/W	EEPROM
R43	Current value 48 to 45	0xEEDD	61149_{dec}	R/W	EEPROM
R44	Current value 52 to 49	0xEEEE	61166_{dec}	R/W	EEPROM
R45	Current value 56 to 53	0xFFEE	65518_{dec}	R/W	EEPROM
R46	Current value 60 to 57	0xFFFF	65535_{dec}	R/W	EEPROM
R47 [▶ 88]	Current value 64 to 61	0xFFFF	65535_{dec}	R/W	EEPROM
R48	reserved	-	-	-	-
...	...	...	...	...	...
R63	reserved	-	-	-	-

Register page 2 (user position table)

Register no.	Comment	Default value		R/W	Memory
R32 [▶ 89]	Target position 0 (low-order word) (starting value for register 2)	0x0000	0_{dec}	R/W	EEPROM
R33	Target position 0 (high-order word) (starting value for register 3)	0x0000	0_{dec}	R/W	EEPROM
R34	Target position 1 (low-order word)	0x0000	0_{dec}	R/W	EEPROM
R35	Target position 1 (high-order word)	0x0000	0_{dec}	R/W	EEPROM
...	...	...	...	...	...
...	...	...	...	...	...
R62	Target position 15 (low-order word)	0x0000	0_{dec}	R/W	EEPROM
R63 [▶ 90]	Target position 15 (high-order word)	0x0000	0_{dec}	R/W	EEPROM

Register page 3 (user velocity and acceleration table)

Register no.	Comment	Default value		R/W	Memory
R32 [▶ 90]	Velocity 0 (starting value for register 39)	0x07FF	2047_{dec}	R/W	EEPROM
R33	Velocity 1	0x07FF	2047_{dec}	R/W	EEPROM
...	...	...	...	...	EEPROM
R47	Velocity 15	0x07FF	2047_{dec}	R/W	EEPROM
R48	Acceleration 0 (starting value for register 40)	0x07FF	2047_{dec}	R/W	EEPROM
R49	Acceleration 1	0x07FF	2047_{dec}	R/W	EEPROM
...	...	...	...	...	EEPROM
R63 [▶ 91]	Acceleration 15	0x07FF	2047_{dec}	R/W	EEPROM

5.5 Register description

All registers can be read or written via register communication. They are used for parameterizing the terminals.

Registers 0 to 31 always have the same meaning.

The content of registers 32 to 63 is specified via the register page selection register (R4 [▶ 81]).

- 0x0000: Register page 0 [▶ 84] - further terminal parameters (default)
- 0x0001: Register page 1 [▶ 88] - current values of the user current table
- 0x0002: Register page 2 [▶ 89] - user position table

Note

In running operation write to the RAM section!

During service operation, registers R37, R38, R39, R40, R47, R48, R49, R50 and R58 should only be written via the RAM section. To this end, the code word in register R31 [▶ 83] must not be set (if the code word is set, data are written to the EEPROM section of the terminal)! The EEPROM section should only be used for initialization after a terminal reset (caused by the software or power-off).
Frequent modifications of the EEPROM register could destroy the memory cells in the long term. Although modern EEPROM blocks are designed for many writing cycles, their service life can be reduced by multiple positioning with different velocities and accelerations.
If a motion command is started with the maximum velocity v_1, for example, a new max. velocity v_2 can be entered in register R39 [▶ 85] and a new target position in registers R2 [▶ 81] and R3 [▶ 81] during the motion. Once the target position is reached and control bit CB.2 [▶ 74] and the process output data are both zero, the terminal calculates the new parameters for the next motion command.

5.5.1 Register description R0 - R31

R0: Actual position (low-order word)
Contains the low-order word of the actual position.
If the latch function is enabled (RP0.R52.2 [▶ 86] = 1_{bin}), this register must be read first. In doing so, registers R0 and R1 are cached. Both registers are enabled again during subsequent read of register R1.

R1: Actual position (high-order word)
Contains the high-order word of the actual position.

R2: Setting of the target position or position (low-order word)
Here you can specify the low-order word for the required position (maximum: 65535).

R3: Setting of the target position or position (high-order word)
Here you can specify the high-order word for the required position (maximum: 65535).

R4: Register page selection register
This register is used for specifying which register page is shown in registers R32 to R63 of the KL2531/KL2541:

- 0x0000: Register page 0 [▶ 84] - further terminal parameters (default)
- 0x0001: Register page 1 [▶ 88] - current values of the user current table
- 0x0002: Register page 2 [▶ 89] - user position table

R5: Latch value (high-order word)
Contains the high-order word of the latch value.
The low-order word of the latch value can only be read in process data mode by setting CW.5 [▶ 76].

R6: Temperature register

The internal temperature of the terminal can be read, in °C, through register R6. The terminal sets bit SB.5 [▸ 74] as a warning if the temperature exceeds the threshold of 80 °C. SB5 is automatically reset when the temperatures drops below 60 °C.

R7: Command register

- **Standard commands**
  To execute a standard command, the user code word 0x1235 must be entered in register R31 [▸ 83].

Command 0x7000: Restore Factory Settings
Entering 0x7000 in register R7 restores the delivery state for the following registers:

Register page 0 (further parameters):
R32 [▸ 84]: 0
R33 [▸ 84]: 200
R34 [▸ 84]: 4000
R35 [▸ 84]: 100
R36 [▸ 84]: 100
R37 [▸ 85]: 20
R38 [▸ 85]: 10
R39 [▸ 85]: 2047
R40 [▸ 85]: 2047
R41 [▸ 85]: 1023
R42 [▸ 85]: 100
R43 [▸ 85]: 100
R44 [▸ 85]: 50
R45 [▸ 85]: 50
R46 [▸ 85]: 6
R47 [▸ 85]: 7
R48 [▸ 85]: 8192
R49 [▸ 85]: 8192

R50 [▸ 86]: 2047
R51 [▸ 86]: 1
R52 [▸ 86]: 0
R53 [▸ 86]: 500
R54 [▸ 86]: 50
R55 [▸ 86]: 0
R56 [▸ 87]: 0
R57 [▸ 87]: 0
R58 [▸ 87]: 2047

Register page 1 (current table):
R32 [▸ 88]: 0x1100
R33 [▸ 88]: 0x3221
R34 [▸ 88]: 0x4433
R35 [▸ 88]: 0x5554
R36 [▸ 88]: 0x7666
R37 [▸ 88]: 0x8877
R38 [▸ 88]: 0x9998
R39 [▸ 88]: 0xAAAA
R40 [▸ 88]: 0xBBBB
R41 [▸ 88]: 0xCCCC
R42 [▸ 88]: 0xDDDC
R43 [▸ 88]: 0xEEDD
R44 [▸ 88]: 0xEEEE
R45 [▸ 88]: 0xFFEE
R46 [▸ 88]: 0xFFFF
R47 [▸ 88]: 0xFFFF

Register page 2 (position table):
R32-R63 [▸ 89]: 0

Register page 3 (velocity and acceleration table):
R32-R63 [▸ 90]: 2047

Command 0x8000: Software Reset
Entering 0x8000 in register R7 initiates a full software reset for the terminal. All internal variables (position, latch value, error, etc.) are deleted or set to a new value, which is calculated from the stored EEPROM values. The internal circuits (DAC, output driver) are reinitialized with a software reset.

⚠ WARNING

The motor is not-braked during a software reset!

During a software reset, the motor is switched current-free, i.e. the rotor axis can rotate freely. Ensure that your system state permits this and that hazards for persons or machinery have been ruled out!

- **Special commands**
  To execute a special command, no code word must be entered in register R31 [▶ 83].

  Command 0x0500: start multiple motion command [▶ 30] **'forward'**
  The entry 0x0500 starts a multiple motion command. Up to 16 positions can be approached, which are specified through a position table. The start position is specified through an offset, which is added to the command code (e.g. use the command code 0x0504 for the start position). The position table is processed in forward direction.

  Command 0x0510: start multiple motion command 'backward'
  The entry 0x0500 starts a multiple motion command. Up to 16 positions can be approached, which are specified through a position table. The start position is specified through an offset, which is added to the command code (e.g. use the command code 0x0518 for the start position). The position table is processed in backward direction.

  Command 0x0520: start referencing [▶ 33]
  The entry 0x0520 starts automatic referencing (homing) of the stepper motor. The digital inputs serve as limit switches.

  Command 0x0530: start self-adjustment [▶ 34]
  The entry 0x0530 starts automatic self-adjustment of the position.

  Command 0x0540: stopping or resetting special functions
  The entry 0x0540 stops the currently active special command.

R8: Terminal type
The terminal identifier is contained in register R8:
KL2531: 0x09E3 (2531$_{dec}$)
KL2541: 0x09ED (2541$_{dec}$)

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.

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 written into the RAM registers and in the EEPROM registers and are therefore retained if the terminal is restarted.
- Registers R37, R38, R39, R40, R47, R48, R49, R50 and R58 can be written without user code. The new values are only stored in the RAM.

The code word is reset if the terminal is restarted.

5.5.2 Register page 0 (user parameters)

R32: Feature register 1
The feature register specifies the terminal's configuration.

Bit	R32.15	R32.14	R32.13	R32.12	R32.11	R32.10	R32.9	R32.8
Name	enCounterMode	disEncFltr	enPosGateLock	enNegGateLock	FeedbackType	enSignalDisplay	enMultiLatchValues	enLatchEventClrPos

Bit	R32.7	R32.6	R32.5	R32.4	R32.3	R32.2	R32.1	R32.0
Name	enUserCurTable	invPosFeedback	RampType	OperationMode		disWdTimer	enAutoStop	enAutoStart

Legend

Bit	Name	Description			default
R32.15	enCounterMode	0_{bin}	Encoder mode active	KL2541 only The KL2531 does not analyse these bits!	0_{bin}
		1_{bin}	Counter mode active		
R32.14	disEncFltr	1_{bin}	Input filter for encoder signals disabled		0_{bin}
R32.13	enPosGateLock	1_{bin}	A positive level at the latch/gate input locks the counter.		0_{bin}
R32.12	enNegGateLock	1_{bin}	A negative level at the latch/gate input locks the counter.		0_{bin}
R32.11	FeedbackType	0_{bin}	Use encoder for position feedback		0_{bin}
		1_{bin}	Use internal counter for position feedback		
R32.10	enSignalDisplay	1_{bin}	The C and gate signals are shown in the status word		0_{bin}
R32.9	enMultiLatchValues	0_{bin}	Only one latch value is saved.		0_{bin}
		1_{bin}	Several latch values are saved. The number is specified in register R37 [▶ 85].		
R32.8	enLatchEventClrPos	1_{bin}	Latch event deletes the current position (with the associated control bits)		0_{bin}
R32.7	enUserCurTable	0_{bin}	Internal current table (sine) active		0_{bin}
		1_{bin}	User current table active		
R32.6	invPosFeedback	1_{bin}	Inverts the position feedback. This option is only intended for the *path control* mode, if an external encoder is installed opposite the stepper motor (e.g. the encoder shows a negative rotation when the motor rotates in positive direction). If this function is enabled with internal position feedback or if the position detection of the external encoder is incorrect, the path control operates incorrectly!		0_{bin}
R32.5	RampType	0_{bin}	Linear ramps		0_{bin}
		1_{bin}	Exponential ramp during deceleration (only for path control)		
R32.4 + R32.3	OperationMode	00_{bin}	Velocity, direct		00_{bin}
		01_{bin}	Velocity, with ramps		
		10_{bin}	Position tracking		
		11_{bin}	Travel distance control		
R32.2	disWdTimer	0_{bin}	Watchdog timer is active (the watchdog is triggered if no process data are received for 100 ms)		0_{bin}
		1_{bin}	Watchdog timer disabled		
R32.1	enAutoStop	1_{bin}	The auto-stop function is enabled		0_{bin}
R32.0	enAutoStart	1_{bin}	The auto-start function is enabled		0_{bin}

R33: Full motor steps
Enter the number of full steps that the connected stepper motor requires for a complete turn (default: 200).

R34: Encoder increments (KL2541 only)
Enter the number of increments issued by the encoder connected to the KL2541 during a complete turn (default: 4000).

R35: Maximum coil current A
This register specifies the current limit for winding A (0 to 100 % of rated terminal current):
KL2531-0000: 100 % = 1.5 A
KL2541-0000: 100 % = 5.0 A

R36: Maximum coil current B
This register specifies the current limit for winding B (0 to 100 % of rated terminal current):
KL2531-0000: 100 % = 1.5 A
KL2541-0000: 100 % = 5.0 A

R37: Number of latch values

This register specifies the number of stored latch values (default: 20, max.: 20).

R38: Min. velocity v_{min}

This register specifies the maximum cutoff velocity for the operating modes velocity, with ramps [▶ 26] and path control [▶ 28] (default: 10, max.: 2047).
The maximum cutoff velocity is the maximum velocity from which the stepper motor can stop directly without step errors. It depends on the holding torque of the motor and mass rotating with its shaft.

R39: Max. velocity v_{max}

This register specifies the maximum velocity for the operating modes velocity, with ramps [▶ 26] and path control [▶ 28] (default: 2047, max.: 2047).

R40: max. acceleration a_{acc}

This register specifies the maximum acceleration for the operating modes velocity, with ramps [▶ 26] and path control [▶ 28] (default: 2047, max.: 2047).

R41: Acceleration threshold a_{th}

This register specifies the acceleration threshold for the operating modes velocity, with ramps [▶ 26] and path control [▶ 28] (default: 2047, max.: 2047).

R42: Coil current, $a > a_{th}$ (in %)

This register specifies the coil current between 0 and 100 % of the set terminal coil current when $a > a_{th}$ (default: 100 %).

R43: Coil current, $a \leq a_{th}$ (in %)

This register specifies the coil current between 0 and 100 % of the set terminal coil current when $a \leq a_{th}$ (default: 100 %).

R44: Coil current, v = 0 (automatic) (in %)

This register specifies the automatic holding current (default: 50 %).
When the motor is at standstill, the coil currents are automatically reduced to this value, in order to prevent unnecessary heating of the motor. This value refers to the set coil current of the terminal.

R45: Coil current (manual) (in %)

This register specifies the manual holding current (default: 50 %).

If control bit CB.3 [▶ 74] or CW.11 [▶ 76] is set, the coil currents can be manually reduced to this value while the motor is at standstill or in motion, in order to avoid unnecessary heating of the motor. This value refers to the set coil current of the terminal. The automatically and manually reduced coil current values are not added! The manual value has a higher priority than the automatic value!

R46: Step size

This register specifies the number of steps for a quarter period.

Register value	0x0006	0x0005	0x0004	0x0003	0x0002	0x0001	0x0000
Steps/quarter period	64	32	16	8	4	2	1
Step size	1/64	1/32	1/16	1/8	1/4	1/2 (half step)	1/1 (full step)

R47: Load angle threshold

This register specifies the load angle threshold, at which the *Stall Detect* LED starts flashing and the status bit SB.5 [▶ 74] is set (default: 7, max.: 7).
The load angle is a measure for the motor load. It is shown with values between 0 and 7, with 7 representing the maximum load. This value is strongly dependent on the set coil current and the current velocity. The most reliable information about the motor load can be gleaned at medium velocities.

R48: Kp factor

This register specifies the Kp factor for the TwinCAT NC cyclic position interface in *position tracking* mode (in preparation) (default: 8192, max.: 65535).

R49: Kv factor

This register specifies the Kv factor for the TwinCAT NC cyclic position interface in *position tracking* mode (in preparation) (default: 8192, max.: 65535).

R50: Emergency acceleration a $_e$

This register specifies the emergency acceleration for path control mode (default: 2047, max.: 2047).

R51: Number of pulses per revolution

This register specifies the number of pulses per revolution for position error detection [▸ 20] (default: 1).

R52: Feature register 2

The feature register 2 specifies further terminal configurations.

Bit	R52.15	R52.14	R52.13	R52.12	R52.11	R52.10	R52.9	R52.8
Name	revDigInput2	revDigInput1	enManEnc Tolerance	enClrProcess Data	-	-	-	-

Bit	R52.7	R52.6	R52.5	R52.4	R52.3	R52.2	R52.1	R52.0
Name	-	-	disReferenceInput2	disReference Input1	ShowIdle	LatchRegister Position	enDeceleration Ramp	revHoming Direction

Legend

Bit	Name	Description			default
R52.15	revDigInput2	0_{bin}	Evaluate digital input 2 as N/O contact: HIGH level = active state (SW.1 [▸ 77]=1_{bin})		0_{bin}
		1_{bin}	Evaluate digital input 2 as N/C contact: LOW level = active state (SW.1 [▸ 77]=1_{bin})		
R52.14	revDigInput1	0_{bin}	Evaluate digital input 1 as N/O contact: HIGH level = active state (SW.0 [▸ 77]=1_{bin})		0_{bin}
		1_{bin}	Evaluate digital input 1 as N/C contact: LOW level = active state (SW.0 [▸ 77]=1_{bin})		
R52.13	enManEnc tolerance (KL2541 only)	0_{bin}	The tolerance of the target position is formed from the quotient of registers R34 [▸ 84] / R33 [▸ 84] (position feedback with encoder).	The KL2531 does not analyse this bit!	0_{bin}
		1_{bin}	The tolerance of the target position is specified in register R57 [▸ 87]		
R52.12	enClrProcess data	1_{bin}	In the event of a watchdog error the input process data are set to zero.		0_{bin}
R52.7 - R52.11	-	reserved			
R52.6	enExtAutostart	0_{bin}	First target position = 0 Second target position = register R2/R3		0_{bin}
		1_{bin}	First target position = register R62/R63 Second target position = register R2/R3		
R52.5	disReferenceInput2	1_{bin}	Digital input 2 disabled as reference input [▸ 33]		0_{bin}
R52.4	disReferenceInput1	1_{bin}	Digital input 1 disabled as reference input [▸ 33]		0_{bin}
R52.3	ShowIdle	1_{bin}	Idle bit of status byte SB.4 [▸ 74] enabled		0_{bin}
R52.2	LatchRegisterPosition	1_{bin}	Latching of position registers R0 [▸ 81] and R1 [▸ 81] enabled		0_{bin}
R52.1	enDecelaration ramp	0_{bin}	The acceleration and deceleration ramps are specified in register R40 [▸ 85].		0_{bin}
		1_{bin}	The acceleration ramp is specified in register R40 [▸ 85]. The deceleration ramp is specified in register R58 [▸ 87].		
R52.0	revHoming direction	0_{bin}	approach the limit switch with negative velocity ($V_{ref,f}$ < 0), move away from the limit switch with positive velocity ($V_{ref,b}$ > 0)		0_{bin}
		1_{bin}	approach the limit switch with positive velocity ($V_{ref,f}$ > 0), move away from the limit switch with negative velocity ($V_{ref,b}$ < 0)		

R53: Referencing velocity backward v $_{ref,b}$

For the referencing function (homing) this register specifies the velocity with which the terminal approaches the limit switch (default: 500, max.: 2047).
In this case backward means that the terminal controls the motor with falling position values.

R54: Referencing velocity forward v $_{ref,f}$

For the referencing function (homing) this register specifies the velocity with which the terminal moves away from the limit switch (default: 50, max.: 2047).
In this case forward means that the terminal controls the motor with rising position values.

R55: Referencing position (low-order word)

Here you can specify the low-order word for the referencing position (default: 0, max.: 65535).

R56: Referencing position (high-order word)
Here you can specify the high-order word for the referencing position (default: 0, max.: 65535).

R57: Encoder tolerance (KL2541 only)
For position feedback via encoder, this register specifies the tolerance of the motion for the target approach, at which bit SW.3 [▸ 77] is set (default: 0, max.: 65535).

R58: max. deceleration a $_{dec}$
Here you can specify the maximum deceleration for the operating modes velocity, with ramps [▸ 26] and path control [▸ 28] (default: 0, max.: 2047).

R62: Setting the target position or position (low-order word) for extended auto-start
Here you can specify the low-order word for the required position (maximum: 65535).
Without code word this value is written into the RAM, with code word written into the EEPROM.

R63: Setting the target position or position (high-order word) for extended auto-start
Here you can specify the high-order word for the required position (maximum: 65535).
Without code word this value is written into the RAM, with code word written into the EEPROM.

5.5.3 Register page 1 (user current table)

Register page 1 specifies the current values for the user current table.

R32: Current value 4 to 1
This register specifies current values 4, 3, 2 and 1.

Structure

Bit	15 to 12	11 to 8	7 to 4	3 to 0
Contents	Current value 4	Current value 3	Current value 2	Current value 1

R33: Current values 8 to 5
This register specifies current values 8, 7, 6 and 5 (structure as current values 4 to 1 [▸ 88]).

R34: Current values 12 to 9
This register specifies current values 12, 11, 10 and 9 (structure as current values 4 to 1 [▸ 88]).

R35: Current values 16 to 13
This register specifies current values 16, 15, 14 and 13 (structure as current values 4 to 1 [▸ 88]).

R36: Current values 20 to 17
This register specifies current values 20, 19, 18 and 17 (structure as current values 4 to 1 [▸ 88]).

R37: Current values 24 to 21
This register specifies current values 24, 23, 22 and 21 (structure as current values 4 to 1 [▸ 88]).

R38: Current values 28 to 25
This register specifies current values 28, 27, 26 and 25 (structure as current values 4 to 1 [▸ 88]).

R39: Current values 32 to 29
This register specifies current values 32, 31, 30 and 29 (structure as current values 4 to 1 [▸ 88]).

R40: Current values 36 to 33
This register specifies current values 36, 35, 34 and 33 (structure as current values 4 to 1 [▸ 88]).

R41: Current values 40 to 37
This register specifies current values 40, 39, 38 and 37 (structure as current values 4 to 1 [▸ 88]).

R42: Current values 44 to 41
This register specifies current values 44, 43, 42 and 41 (structure as current values 4 to 1 [▸ 88]).

R43: Current values 48 to 45
This register specifies current values 48, 47, 46 and 45 (structure as current values 4 to 1 [▸ 88]).

R44: Current values 52 to 49
This register specifies current values 52, 51, 50 and 49 (structure as current values 4 to 1 [▸ 88]).

R45: Current values 56 to 53
This register specifies current values 56, 55, 54 and 53 (structure as current values 4 to 1 [▸ 88]).

R46: Current values 60 to 57
This register specifies current values 60, 59, 58 and 57 (structure as current values 4 to 1 [▸ 88]).

R47: Current values 64 to 61
This register specifies current values 64, 63, 62 and 61 (structure as current values 4 to 1 [▸ 88]).

5.5.4 Register page 2 (user position table)

Register page 2 specifies the target position for the user position table.

R32: Target position 0 (low-order word - starting value for register 2)
This register specifies the low-order word for target position 0.

R33: Target position 0 (high-order word - starting value for register 3)
This register specifies the high-order word for target position 0.

R34: Target position 1 (low-order word)
This register specifies the low-order word for target position 1.

R35: Target position 1 (high-order word)
This register specifies the high-order word for target position 1.

R36: Target position 2 (low-order word)
This register specifies the low-order word for target position 2.

R37: Target position 2 (high-order word)
This register specifies the high-order word for target position 2.

R38: Target position 3 (low-order word)
This register specifies the low-order word for target position 3.

R39: Target position 3 (high-order word)
This register specifies the high-order word for target position 3.

R40: Target position 4 (low-order word)
This register specifies the low-order word for target position 4.

R41: Target position 4 (high-order word)
This register specifies the high-order word for target position 4.

R42: Target position 5 (low-order word)
This register specifies the low-order word for target position 5.

R43: Target position 5 (high-order word)
This register specifies the high-order word for target position 5.

R44: Target position 6 (low-order word)
This register specifies the low-order word for target position 6.

R45: Target position 6 (high-order word)
This register specifies the high-order word for target position 6.

R46: Target position 7 (low-order word)
This register specifies the low-order word for target position 7.

R47: Target position 7 (high-order word)
This register specifies the high-order word for target position 7.

R48: Target position 8 (low-order word)
This register specifies the low-order word for target position 8.

R49: Target position 8 (high-order word)
This register specifies the high-order word for target position 8.

R50: Target position 9 (low-order word)
This register specifies the low-order word for target position 9.

R51: Target position 9 (high-order word)
This register specifies the high-order word for target position 9.

R52: Target position 10 (low-order word)
This register specifies the low-order word for target position 10.

R53: Target position 10 (high-order word)
This register specifies the high-order word for target position 10.

R54: Target position 11 (low-order word)
This register specifies the low-order word for target position 11.

R55: Target position 11 (high-order word)
This register specifies the high-order word for target position 11.

R56: Target position 12 (low-order word)
This register specifies the low-order word for target position 12.

R57: Target position 12 (high-order word)
This register specifies the high-order word for target position 12.

R58: Target position 13 (low-order word)
This register specifies the low-order word for target position 13.

R59: Target position 13 (high-order word)
This register specifies the high-order word for target position 13.

R60: Target position 14 (low-order word)
This register specifies the low-order word for target position 14.

R61: Target position 14 (high-order word)
This register specifies the high-order word for target position 14.

R62: Target position 15 (low-order word)
This register specifies the low-order word for target position 15.

R63: Target position 15 (high-order word)
This register specifies the high-order word for target position 15.

5.5.5 Register page 3 (user velocity and acceleration table)

Register page 3 specifies the velocities and accelerations for the corresponding target positions in the user position table.

R32: Velocity 0 (starting value for register 39)
This register specifies velocity 0.

R33: Velocity 1
This register specifies velocity 1.

R34: Velocity 2
This register specifies velocity 2.

R35: Velocity 3
This register specifies velocity 3.

R36: Velocity 4
This register specifies velocity 4.

R37: Velocity 5
This register specifies velocity 5.

R38: Velocity 6
This register specifies velocity 6.

R39: Velocity 7
This register specifies velocity 7.

R40: Velocity 8
This register specifies velocity 8.

R41: Velocity 9
This register specifies velocity 9.

R42: Velocity 10
This register specifies velocity 10.

R43: Velocity 11
This register specifies velocity 11.

R44: Velocity 12
This register specifies velocity 12.

R45: Velocity 13
This register specifies velocity 13.

R46: Velocity 14
This register specifies velocity 14.

R47: Velocity 15
This register specifies velocity 15.

R48: Acceleration 0
This register specifies acceleration 0.

R49: Acceleration 1
This register specifies acceleration 1.

R50: Acceleration 2
This register specifies acceleration 2.

R51: Acceleration 3
This register specifies acceleration 3.

R52: Acceleration 4
This register specifies acceleration 4.

R53: Acceleration 5
This register specifies acceleration 5.

R54: Acceleration 6
This register specifies acceleration 6.

R55: Acceleration 7
This register specifies acceleration 7.

R56: Acceleration 8
This register specifies acceleration 8.

R57: Acceleration 9
This register specifies acceleration 9.

R58: Acceleration 10
This register specifies acceleration 10.

R59: Acceleration 11
This register specifies acceleration 11.

R60: Acceleration 12
This register specifies acceleration 12.

R61: Acceleration 13
This register specifies acceleration 13.

R62: Acceleration 14
This register specifies acceleration 14.

R63: Acceleration 15
This register specifies acceleration 15.

5.6 Examples of Register Communication

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

5.6.1 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.

5.6.2 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