I/O Terminal

Installation Manual

Revision 1.1, June 2011

AKT-SM-L15-000 Stepper Module, 24 V DC, 1.5 A
AKT-SM-L50-000 Stepper Module, 50 V DC, 5 A

Keep all manuals as a product component during the life span of the product.
Pass all manuals to future users / owners of the product.

KOLLMORGEN®

Because Motion Matters™

Record of Document Revisions

Revision	Remarks
1.0	Preliminary edition
1.1	Added the mechanical drawing to Appendix A. For more information, see “Appendix A” page 79.

IMPORTANT NOTICE

Disclaimer

The information in this manual was accurate and reliable at the time of its release. However, Kollmorgen. reserves the right to change the specifications of the product described in this manual without notice at any time.

This document contains proprietary and confidential information of Kollmorgen. The contents of the document may not be disclosed to third parties, translated, copied or duplicated in any form, in whole or in part, without the express written permission of Kollmorgen.

Registered Trademarks

Kollmorgen is a registered trademark of Danaher Corp.

June 2011

1 SAFETY PRECAUTIONS

This chapter provides safety information for the Stepper Module.

1.1 Safety Rules

The appropriate 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.

1.2 State at Delivery

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 any liability from Kollmorgen.

1.3 Personnel Qualification

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

1.4 Description of Notes and Warnings

The following notes and warnings are used in this manual. They are intended to alert the reader to the associated safety instructions.

Danger — This note is intended to highlight risks for the life or health of personnel.

Warning — This note is intended to highlight risks for equipment, materials or the environment.

Note — Indicates information that contributes to better understanding.

2 OVERVIEW

This section provides an overview of the Stepper Module.

Note: For information about configuring the Stepper Module, see the Kollmorgen Automation Suite™ IDE software and online help system.

2.1 AKT-SM-L15-000 Stepper Module

The Bus Terminal is intended for the direct connection of different small stepper motors. The slim-line PWM output stages for two motor coils are located in the Bus Terminal together with two digital inputs (24 V) for limit switches. The Bus Terminal can be adjusted to the motor and the application by changing just a few parameters. 64-fold micro-stepping ensures particularly quiet and precise motor operation. In many applications, integrated monitoring of the mechanical load makes an encoder system or limit switch unnecessary.

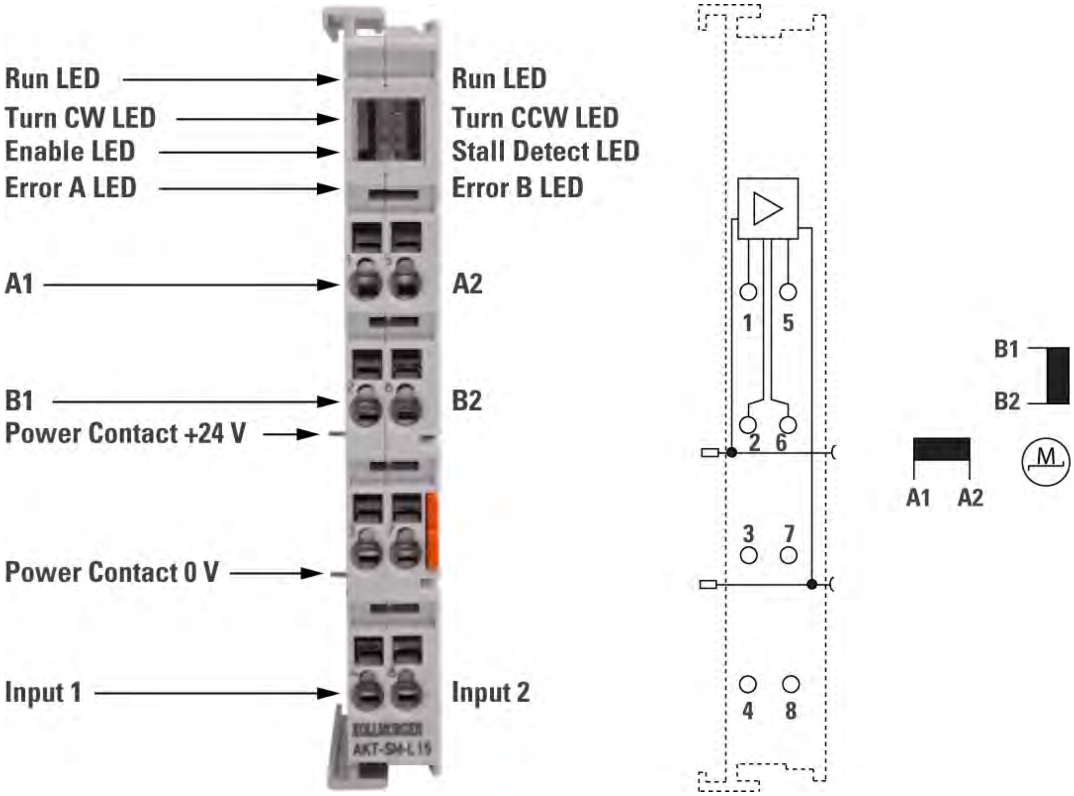


Figure 2.1 Top View, Contact Assembly, and Connection

2.1.1 Technical Data

This section provides the technical details for the Bus Terminal.

Parameters	(AKT-SM-L15-000)
Number of digital outputs	1, for one stepper motor (2 phases)
Power supply for output stages (via power contacts)	8 V_{DC} to 24 V_{DC}
Output current	2 x 1 A, 2 x 1.5 A peak current, overload and short circuit protected
Current controller frequency	ca. 25 kHz
Maximum step frequency	125 kHz
Step pattern	Full step, half step, 4 / 8 / 16 / 32 / 64-fold micro-stepping
Resolution	Max. 64-step micro-stepping
Resolution per revolution	Approx. 5000 positions at typical applications
Diagnostics LED for	Error phase A, error phase B, loss of step/stagnation, power, enable, direction of rotation, digital inputs
Number of digital inputs	2
Nominal voltage	24 V_{DC} (-15%/+20%)
"1" signal voltage	15 V ... 30 V
Input filter	0.2 ms
Input current	Typically 5 mA
Electrical isolation	500 V_{rms} (Standard-Bus/signal voltage)
Power supply for the electronics	Via the Standard-Bus
Current consumption from the Standard-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
Dimensions (W x H x D)	Approx. 15mm x 100 mm x 70 mm (width aligned: 12 mm)
Weight	Approx. 55 g
Permissible ambient temperature during operation	0°C... +55°C
Permissible ambient temperature during storage	-25°C... + 85°C

Parameters	(AKT-SM-L15-000)
Permissible relative humidity	95%, no condensation
Assembly	On 35 mm mounting rail according to EN 50022
Vibration / shock resistance	Conforms to EN 60068-2-6 / EN 60068-2-27, EN 60068-2-29
EMC resistance burst / ESD	Conforms to EN 61000-6-2 / EN 61000-6-4
Protection class	IP 20
Installation position	Variable
Approval	CE

2.1.2 LED Display

The section provides the details of the LED display.

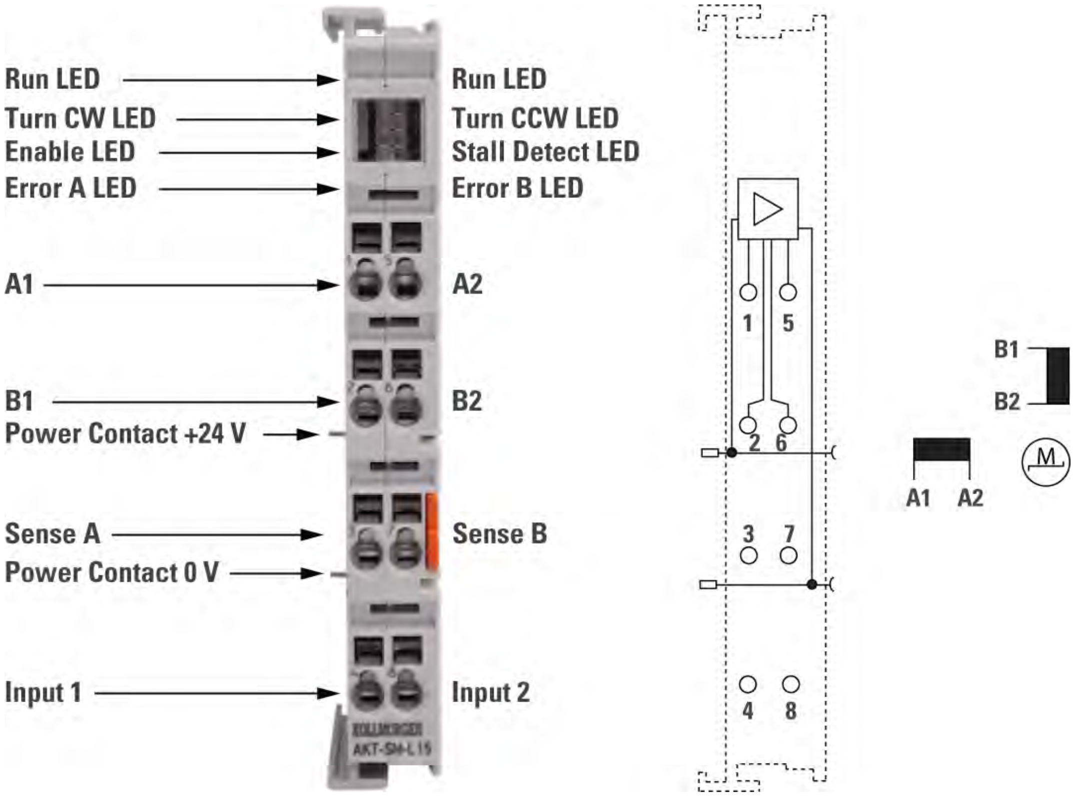


Figure 2.2 LEDs Operation Status

2.1.3 LEDs Defined

The following table describes the operation status of the LEDs for the Bus Terminal.

LED	Display	Description
Run (Green)	On	Data transmission on the Standard-Bus is active
Turn CW (Green)	On	Motor turns clockwise
Enable (Green)	Off	Motor control is disabled or device is not ready
	On	Motor control is disabled or device is ready
Error A (Red)	Flashes (1 Hz)	Configuration error
Power (Green)	Off	Supply voltage ($24V_{DC}$) not available or motor control is disabled
	On	Supply voltage ($24V_{DC}$) is available
Turn CCW (Green)	On	Motor turns counter-clockwise
Stall detect (Yellow)	On	Motor control is disabled
	Flashes (1 Hz)	The current load angle is greater than the configured angle threshold
Error B (Red)	Flashes (1 Hz)	Supply or load error

2.2 AKT-SM-L50-000 Stepper Module

The 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 (24 V) for limit switches and an incremental encoder interface (24 V), they are located in the Bus Terminal. The terminal can be adjusted to the motor and the application by changing just a few parameters. 64-fold micro-stepping ensures particularly quiet and precise motor operation. Together with a stepper motor, the Bus Terminal represents an inexpensive small servo axis.

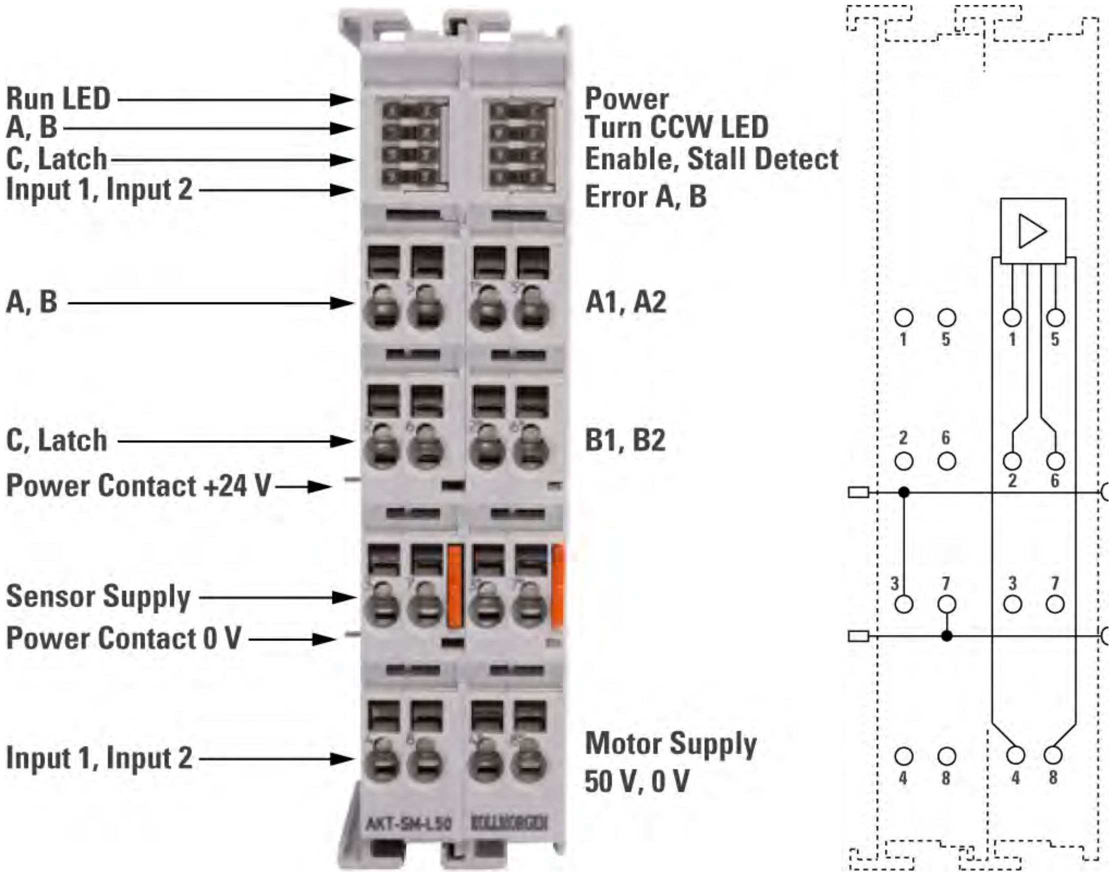


Figure 2.3 Contact Assembly Connection (Top View)

2.2.1 Technical Data

This section provides the technical details for the Bus Terminal.

Parameters	AKT-SM-L50-000
Number of digital outputs	1, for one stepper motor (2 phases)
Power supply for output stages (via power contacts)	8 V_{DC} to 50 V_{DC}
Output current	2 x 5 A peak current
Current controller frequency	ca. 25 kHz
Maximum step frequency	125 kHz
Step pattern	Full step, half step, 4 / 8 / 16 / 32 / 64-fold micro-stepping
Resolution	Max. 64-step micro-stepping
Resolution per revolution	~5000 positions at typical applications
Number of encoder inputs	4, for one encoder system
"0" encoder voltage	-3 V ... 2 V
"1" encoder voltage	2.5 V ... 28 V
Encoder frequency	Max. 250 kHz (what is an incremental signal of 1 MHz at quadruple evaluation)
Number of digital inputs	2, for limit position
Nominal voltage	24 V_{DC} (-15%/+20%)
"0" signal voltage	-3 V ... 5 V
"1" signal voltage	15 V ... 30 V
Input filter	0.2 ms
Input current	Typically 5 mA
Diagnostics LED for	Error phase A, error phase B, loss of step/stagnation, power, enable, direction of rotation, encoder signals, digital inputs
Electrical isolation	500 V_{rms} (Standard-Bus/signal voltage)
Power supply for electronics	Via the Standard-Bus
Current consumption from the Standard-Bus	Typically 80 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
Dimensions (W x H x D)	~27mm x 100 mm x 70 mm (width aligned: 24 mm)
Weight	~100 g
Permissible ambient temperature during storage	0°C... +55°C

Parameters	AKT-SM-L50-000
Permissible relative humidity	-25°C... + 85°C
Assembly	On 35 mm mounting rail according EN 50022
Vibration / shock resistance	Conforms to EN 60068-2-6 / EN 60068-2-27, EN 60068-2-29
EMC resistance burst / ESD	Conforms to EN 61000-6-2 / EN 61000-6-4
Protection class	IP 20
Installation position	Variable
Approval	CE

2.2.2 LED Display

The section provides the details of the LED display.

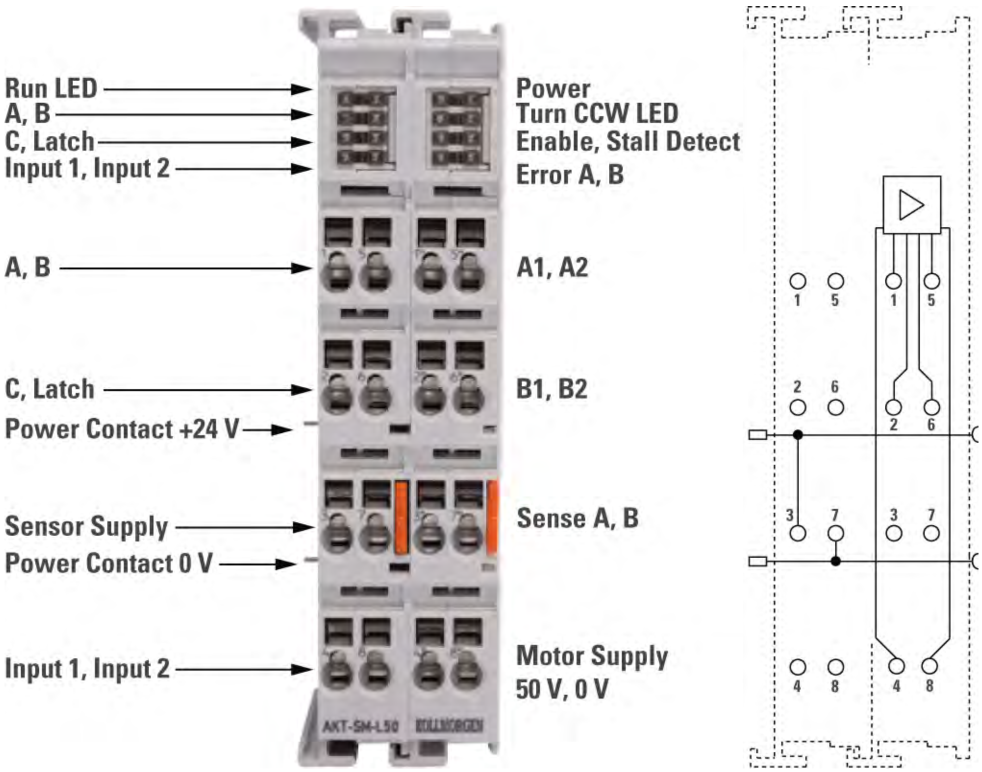


Figure 2.4 LEDs Operation Status

2.2.3 LEDs Defined (Left Prism)

The following table describes the operation status of the LEDs for the left prism.

LED	Display	Description
Run (Green)	On	Data transmission on the Standard-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 digital input E1.
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.

2.2.4 LEDs Defined (Right Prism)

The following table describes the operation status of the LEDs for the right prism.

LED	Display	Description
-	Reserved	
Turn CW (Green)	On	Motor turns clockwise
Enable (Green)	Off	Motor control is disabled or device is not ready
	On	Motor control is disabled or device is ready
Error	Flashes (1 Hz)	Configuration error
Power (Green)	Off	Supply voltage (50V_{DC}) not available or motor control is disabled
	On	Supply voltage (50V_{DC}) is available
Turn CCW (Green)	On	Motor turns counter-clockwise
Stall detect (Yellow)	On	Motor control is disabled
	Flashes (1 Hz)	The current load angle is greater than the configured load angle threshold
Error B (Red)	Flashes (1 Hz)	Supply or load error: – Supply voltage (24 V_{DC}) is too low – Open load at bridge A or B – Over current at bridge A or B

2.3 Basic Function Principles

The AKT-SM-L15-000 and AKT-SM-L50-000 Stepper Modules integrate a compact motion control solution for miniature stepper motors up to 200 W.

2.3.1 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 a minimum control 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 AKT-SM-L15-000 and AKT-SM-L50-000 Bus Terminals for the Bus Terminal system opens up new application areas. Micro-stepping 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

2.3.2 Two Stepper Motor Terminals for Optimum Performance

The AKT-SM-L15-000 and AKT-SM-L50-000 Stepper Modules differ in terms of performance.

AKT-SM-L15-000

With a size of only 12 mm, the AKT-SM-L15-000 covers the lower performance range. The supply voltage can be up to 24 VDC. The device is designed for simple integration into the 24 VDC control voltage system. With a peak current of 1.5 A per phase, a large number of small drives and axes can be supplied.

AKT-SM-L50-000

The AKT-SM-L50-000 offers higher performance comparable to that of small servo drives. With a peak current of 5 A, the AKT-SM-L50-000 can generate an impressive torque of 5 Nm in conjunction with a standard stepper motor, for example. The supply voltage of up to 50 VDC enables high speeds with good torque and therefore high mechanical output (up to about 200 W). The AKT-SM-L50-000 has an integrated incremental encoder interface for connecting all drive cables, although it is still only 24 mm wide.

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 micro stepping). 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, 5,000 positions are realistic in terms of the mechanics.

Typical stepper motor problems such as pronounced resonance are therefore a thing of the past. Micro-stepping 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. The costs and development effort are therefore 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 field busses. 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 AKT-SM-L15-000 and AKT-SM-L50-000 Stepper Modules 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.

2.3.3 Realization of More Demanding Positioning Tasks

More demanding positioning tasks can be realized via the Kollmorgen Automation Suite™ IDE software. Like other axes, the two Stepper Motor Terminals are integrated via the IDE software and can be used like standard servo axes. Special stepper motor features, such as speed reduction in the event of large following 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 field bus to another one using the IDE software.

The output stages of the stepper motor terminals have an overload protection in the form of an over-temperature 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.

Optimum adaptation to the motor and the implementation of energy-saving features require minimum programming effort. During the test phase, the IDE 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.

2.3.4 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.

Rotation 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 AKT-SM-L15-000 and AKT-SM-L50-000 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.

Rated 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 AKT-SM-L15-000 and AKT-SM-L50-000 apply a controlled current to the motor winding. If the supply voltage falls below the rated 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. 250Hz
- 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, the AKT-SM-L15-000 and AKT-SM-L50-000 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° or 0.9°, equivalent to 100, 200 or 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 micro stepping function offered by the AKT-SM-L15-000 and AKT-SM-L50-000. It enables up to 64 intermediate steps. The smaller "artificial" step angle has a further positive effect: The drive can be operated at higher speed, yet with the same precision. The maximum speed is unchanged, despite the fact that the drive operates at the limit of mechanical resolution.

2.3.5 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 micro stepping 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 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 behavior 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.

2.3.6 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) 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 phase also generate high load at the motor.
- **Mixed Decay** – If this option is enabled, the motor current is actively impressed. Deactivation via CW.9 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 satisfactory a 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 has to be activated. The user should save the adjusted current table in register page 1.

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 micro step phase through pulsing. In micro step 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 deactivated via control bit CW.9.

WARNING!!	*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
- Manual, specified via register R45

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 impressed manually onto the motor windings by setting control bits CB.3 or CW.11. This can be done on standstill and when the motor is running.

Specify Set-position (via registers)

The user can set or delete the set-position position value. Registers R2 and R3 are used as reference:

- A rising edge of CW.13 deletes the specified set-position (higher priority as CW.10 if CW.10 and CW.13 are set simultaneously)
- A rising edge of CW.10 sets the set-position to the value of register R2 and R3

Acknowledgement occurs via status bit SW.2.

Path control

For positioning taken over from a PLC, path control is the optimum solution. In this 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.

Detection of positioning error Detection

Via an external sensor (e.g. an inductive proximity switch) connected to the digital input 2, the terminal identifies the relative positioning error of the motor(only for internal positioning). Register RP0.R51 specifies the number of impulses per rotation (e.g. if a pinion is used). By setting bit CW.7 of the Control Word, the user is able to display the error into the process data and to analyze them with the PLC.

The terminal calculates a position delta by using this parameters:

1.1	IF = MS x Reg33 / Reg51	Impulse factor
1.2	ΔPos = Pos - IC x IF	relative position error

key:

IF	impulse factor	[IF] = micro steps per impulse
IC	impulse counter	counted impulses at digital input 2
MS	micro steps per full step	MS = 2^{R46}
Reg33	motor full steps	
Reg46	step size per quarter period	
Reg51	number of impulses per rotation	
Pos	present position (must value)	[Pos] = micro steps
ΔPos	relative position error	[ΔPos] = micro steps

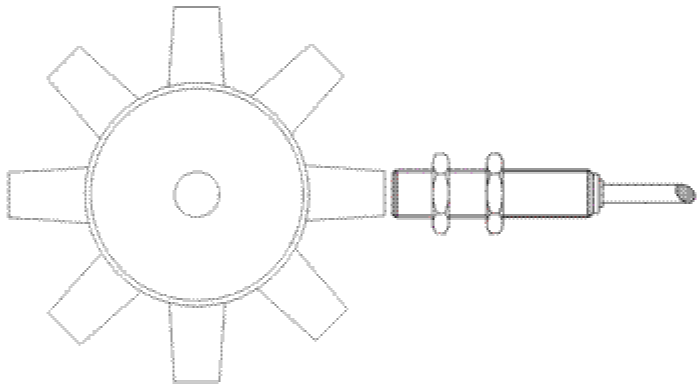


Figure 2.5: Pinion with Inductive Proximity Switch (Register R51 = 8)

The relative positioning error has to be analyzed individually by the user. It gives no direct conclusion about the real number of lost micro steps. This is caused by the fact, that the terminal doesn't calculates the beginning error (number of micro steps from the start of the motor to the first sensor impulse). Also the tolerance and the delay, with the sensor switches the 24 V can't be acquired.

From this consideration, a velocity dependence is given, that has to be regarded at the analysis. At constant velocity, the error fluctuates by a few micro steps, but is equal in median. In practice, the small fluctuation doesn't matter because, at fault:

- the motor loses many steps at one time (that means this can be detected by a jumping up error)

  or

- stops complete, what is detectable by a steady growing up positioning error.

Encode Interface

The encoder works with fourfold evaluation.

Latch functions

The internal encoder offers the option of registering one or several latch events. A latch event can be generated via the C, latch/gate, Input 1 or Input 2 input signals. By default, the terminal only stores one latch value. By setting feature bit R32.9 and parameterzing register R37, the latch array can be activated and the number of latch values can be increased so that more than one latch values can be stored.

The terminal response to the latch event is activated as follows:

- Setting control bit CW.0 activates a rising edge at the C input (highest priority if several control bits are set at the same time)
- Setting control bit CW.3 activates a rising edge at the C input (2nd highest priority)
- Setting control bit CW.4 activates a falling edge at the latch / gate input (3rd highest priority)
- Setting control bit CW.1 activates a rising edge at the digital input 1 (4th highest priority)
- Setting control bit CW.2 activates a rising edge at the digital input 2 (lowest priority)

By activating R32.8 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. The current position is deleted during the next latch event.

Once the user has enabled the function, during the next latch event, the terminal saves the current position value and indicates this by setting status bit SW.4. 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, which causes the low-order word to be shown in the DataIN process data (the terminal indicates this via status bit SW.5). The high-order word can only be read from register R5 via register communication. The following latch values can be retrieved by changing control bit CW.6. The terminal acknowledges this by changing status bit SW.6. 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: While reading the latch values, the before set enabling has to be hold up. The latch values are lost, if the enabling is taken away!

Digital Inputs

The digital inputs can be configured for brake contacts. Therefore for Input 1 bit RP0R.52.14 and for Input 2 bit RP0.R52.15 of feature register 2 has to be set to 1_{bin}. In delivery state both inputs are configured for make contacts.

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 following registers: R33, R34 (only AKT-SM-L15-000), R38, R39, R40, R42, R43 or R50 is zero), the ConfigError bit (SW.15) 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, 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 to be set!

Hardware Warnings

If the temperature inside the terminal reaches 80 °C, the bit SB.5 (Warning) of Status Byte is set. If the temperature falls under 60 °C, SB.5 is set back automatically.

Hardware Error

If one of the following errors occur, the motor is switched current-free, and bit SB.6 is set in the status byte.

- Overtemperature (OverTemperature SW.8)
- Undervoltage (UnderVoltage SW.9)
- Open load, bridge A (OpenLoad Br A, SW.10)
- Open load, bridge B (OpenLoad Br B, SW.11)
- Overcurrent bridge A (OverCurrent Br A, SW.12)
- Overcurrent bridge B (OverCurrent Br B, SW.13)

Failure of 24 V Control Voltage (NoControlPower, SW.14)
(The terminal is reinitialized automatically if the control power voltages returns)

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

2.4 Temperature Monitoring

The stepper motor terminals AKT-SM-L15-000 and AKT-SM-L50-000 have integrated temperature monitoring for housing temperature and the stepper controller chip.

Housing Temperature

If the temperature inside the terminal housing reaches 80°C, the bit SB.5 (Warning) of Status Byte is set. If the temperature falls under 60°C, SB.5 is set back automatically.

You can read the present temperature of the terminal housing from temperature register R6.

Temperature of the Stepper Controller Chip

If the temperature of the stepper controller chip exceeds 125°C,

- the terminal switches the motor off automatically. Furthermore
- bit SW.8 (OverTemperature) of status word is set,
- the LED Error B flashes,
- bit SB.6 (Error) of status byte is set
  (Bit SB.6 of status byte can be set back by setting bit CB.6 at the control byte.)

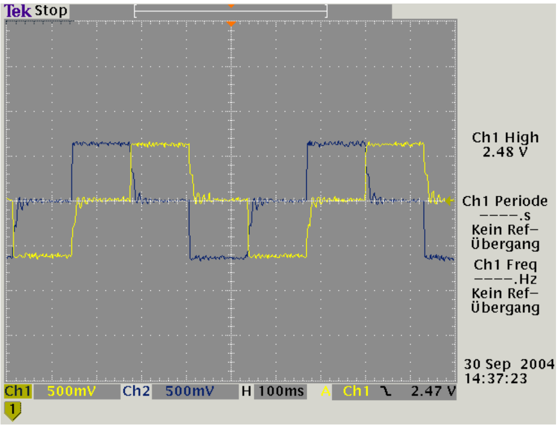
WARNING!! *Control the temperature monitoring from your user program, if you operate the stepper motor terminals at their performance limits!*

2.5 Microstepping

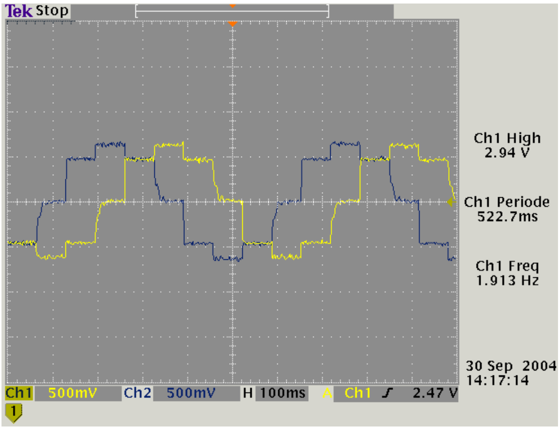
Microstepping can be specified via registers R46 and R48 to R63.

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

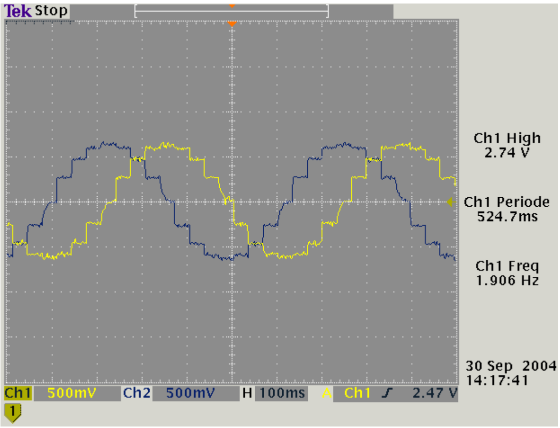
Full Step



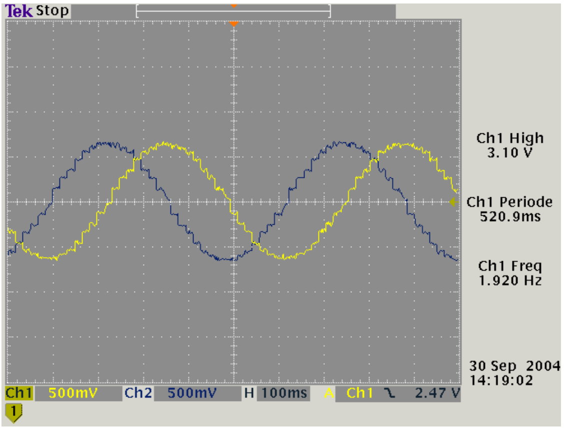
Half Step



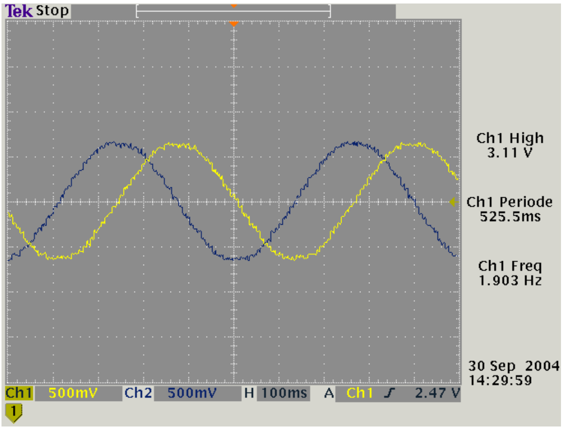
1/4 Step



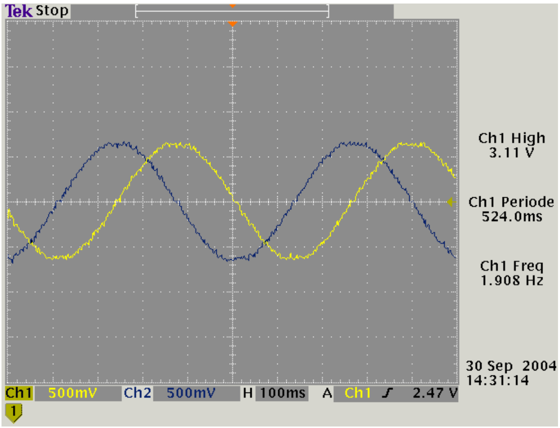
1/8 Step



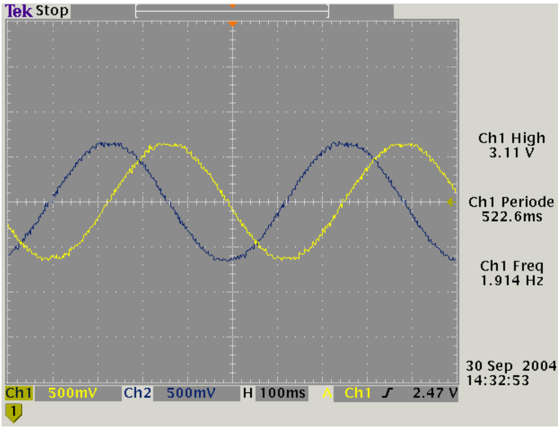
1/16 Step



1/32 Step



1/64 Step



2.6 Operating Modes

The Stepper Motor Terminals support the following operating modes:

- Velocity, direct
- Velocity, with ramps
- Position tracking
- Path control
  - Manual (velocity, with ramps)
  - Travel command (single)
  - Travel command (multi)
  - Auto-start function
  - Auto-stop function
  - Fast-stop function
  - Referencing (via digital inputs)
  - Self adjusting

The operating modes can be selected by using the register communication and feature register RP0.32.

<table>
<tr><th colspan="2">Operation Mode</th><th>For Operation With</th><th>Range of Values for DataOUT</th></tr>
<tr><td colspan="2">Velocity, direct</td><td>NC, cyclic velocity interface</td><td>-7FFF_{hex} to +7FFF_{hex}</td></tr>
<tr><td colspan="2">Velocity, with ramps</td><td>PLC</td><td>-7FFF_{hex} to -0010_{hex}
+0010_{hex} to +7FFF_{hex}</td></tr>
<tr><td colspan="2">Position tracking
(in preparation)</td><td>NC, cyclic position interface</td><td>-7FFF_{hex} to +7FFF_{hex}</td></tr>
<tr><td rowspan="2">Path control</td><td>Manual</td><td rowspan="2">PLC</td><td>-7FFF_{hex} to -0010_{hex}
+0010_{hex} to +7FFF_{hex}</td></tr>
<tr><td>Travel command</td><td>0000_{hex}</td></tr>
</table>

2.6.1 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 run-up and breaking of the motor are also controlled by the NC.

2.6.2 DataOUT

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

<table>
<tr><th>Bit</th><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><th>Contents</th><td>VZ</td><td colspan="15">Velocity value</td></tr>
</table>

Value range: $-7FFF_{hex}$ to $+7FFF_{hex}$ (-32767_{dec} to $+32767_{dec}$)

2.6.3 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 of the PLC, taking into account ramps that the terminal determines from the parameters maximum velocity (RP0.R39), maximum acceleration (RP0.R40), and minimum velocity (RP0.R38). These parameters can be deposited in the terminal registers via register.

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

Value range: $-7FFF_{hex}$ 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.

The maximum velocity specified in register RP0.R39 limits the motor speed in operating modes *velocity, with* ramps and *path control*.

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	Value in register 33	Number of full motor steps per revolution

RP0.R46	Value in register 46

R46	0	1	2	3	4	5	6
MS	1	2	4	8	16	32	64

DataOUT — Value in the output data word

Calculation of 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 = $\Delta R \times 2^{38} / f^2$ — Acceleration in MS/s^2

4.2 Reg = $\Delta R_{FS} \times MS \times 2^{38} / f^2$ — Acceleration in FS/s^2

Simplified formulas:

4.3 Reg = $\Delta R \times 1.073742 \times 10^{-3}$ — Acceleration in MS/s^2

4.4 Reg = $\Delta R_{FS} \times MS \times 1.073742 \times 10^{-3}$ — Acceleration in FS/s^2

Conversion to full steps

5.1 $\Delta R_{FS} = \Delta R / MS; \Delta R_{FS} = R / 2^{R46}$

with:

ΔR	Acceleration	$[\Delta R]$ = microsteps / s^2
ΔR_{FS}	Acceleration in full steps	$[\Delta R_{FS}]$ = 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 (R40)	$[a_{max}] = 1$

2.6.4 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 registers R48 (Kp factor) and R49 (Kv factor) are used for this purpose.

2.6.5 Path Control

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

Path control mode offers the following variants:

2.6.6 Manual

Manual mode is identical to velocity interface with ramps. The terminal automatically selects this mode if no travel command is active (CB.2 = 0).

2.6.7 Travel Command (Single)

With path control (single), the terminal position is defined by a value with a maximum distance of +/- 231 positions increments from the current position. The controller can read the current position from registers R0 and R1. While a travel command is active, the controller can already parameterize the terminal for the next travel command. The new parameters are written to the RAM section of the register (to this end, the code word register must be deleted!). The exact proceeding is described in detail as an example below.

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

R2: target position (lower value word)

R3: target position (higher value word)

RP0.R38: minimum velocity

RP0.R39: maximum velocity

RP0.R40: maximum acceleration

RP0.R58: maximum deceleration (has to be enabled by bit RP0.R52.1 of feature register 2)

Example

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

1. In the control byte, set bit CB.0 to 1 for enabling motor control.
2. Enter the set position (32 bit) in registers R2 and R3 via register communication.
3. Enter the remaining parameters in registers R37, R38, R39 and R40 (all register page 0) via register communication, if they are to be modified.
4. The process output data DataOUT now have to be zero for the travel command to be started.
5. In the control byte, set bit CB.2 to 1 for starting the travel command. The stepper motor terminal will now independently move the motor to the target position. Leave bit CB.2 set to 1 until the motor has reached the desired position. To abort the travel command delete bit CB.2.
6. During the active travel command enter the new parameters in registers R37, R38, R39 and R40 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 in the status word.
8. Deleting bit CB.2 in the control byte completes the first travel command.
9. The terminal acknowledges this by deleting bit SW.3.
10. Set bit CB.2 to 1 for starting the new travel command.
11. When the second target position is reached, the terminal reports this by setting again bit SW.3.
12. Deleting bit CB.2 in the control byte completes the second travel command.
13. The terminal acknowledges this by deleting again bit SW.3.
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.

2.6.8 Travel Command (multi)

With path control (multi) the terminal is been given up to 16 position values, 16 velocity values (from Firmware 3E) and 16 acceleration values (from Firmware 3E) stored internally in different tables. A position consists out of two registers, archived in the registers R32 to R63 of register page 2. Each of this position values is allowed to be at maximum +/- 231 positions increments away from the present position. The present position can be read by the PLC from the registers R0 and R1. The velocities are archived in the register R32 to R47 and the accelerations in the register R48 to R63 of Register-Page 3.

During an active travel command the PLC is already able to parameterize the terminal for the next following travel command (except R39 and R40, these register will be overwritten before the next travel command starts). In doing so the new parameters are written to the RAM-Area of the terminal (therefore the code word register has do be deleted!).

A multiple travel command is started, by writing register R7

with command 0x0500 (here after every travel command the internal position index is incremented, until position 15 is reached - this means the table is worked out forwards) or

with command 0x0510 (here after every travel command the internal position index is decremented, until position 0 is reached - this means the table is worked out backwards).

The start position is set by adding an offset to the command code 0x0500 or 0x0510 (e.g. 0x0504 -> start position 4 forward, 0x0518 -> start position 8 backward). A multiple travel command can be stopped by command 0x0540. The exact proceeding is described in detail as an example below.

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

- RP2.R32: target position 0 (lower value word)
- RP2.R33: target position 0 (higher value word)
- RP2.R34: target position 1 (lower value word)
- RP2.R35: target position 1 (higher value word)
- etc.

Up to 16 possible velocities are specified in the following registers:

- RP3.R32: velocity 0
- RP3.R33: velocity 1
- etc.

Up to 16 possible accelerations are specified in the following registers:

- RP3.R32: acceleration 0
- RP3.R33: acceleration 1
- etc.

For useful and effective traveling operation with multiple positioning the following registers can be re-parameterized:

- RP0.R38: minimum velocity
- RP0.R58: maximum deceleration (has to be enabled by bit RP0.R52.1 of feature register 2)

Example

In the following example a multiple travel command is started at index 5, then tree positions are approached (position 5, 6 and 7). Afterwards the internal position index is set to 3 and two positions are approached (position 3 and 2). Then the multiple travel command is stopped.

1. Enlist the target positions via register communication (max. 16 positions per 32 bit) into register RP2.R32 to RP2.R63.
2. Set bit CB.0 of control byte to 1bin, to enable the motor control.
3. Enlist the further parameters via register communication into register RP0.R37, RP0.R38, RP0.R39, RP0.R40 und RP0.R58, if they should be re-parameterized.
4. Now, the Process output data DataOUT has to be zero, so that the travel command can be started.
5. Write the start command 0x0505 (0x0500 + Offset 5) via register communication into Register R7, to set the internal position index to position 5 as first target position.
6. Set bit CB.2 of control byte to 1bin, to start the first travel command.
7. Now, the stepper motor terminal drives the motor autonomously to target position. Hold bit CB.2 on 1bin until the motor has reached the desired position. If the travel command should be aborted in between, set bit CB.2 back to 0bin.
8. During the active travel command, enlist the new parameters via register communication into register RP0.R37, RP0.R38, RP0.R39, RP0.R40 und RP0.R58, if they should be re-parameterized.
9. The Terminal reports by setting bit SW.3 of status word to 1bin the achievement of the target position.
10. Set bit CB.2 of control byte back to 0bin, to finalize the first travel command.
11. The terminal acknowledges this by setting bit SW.3 back to 0bin.
    On this, the position index is automatically incremented about one position (to position 6).
12. Set bit CB.2 of control byte to 1bin, to start the second travel command.
13. Set bit CB.2 of control byte back to 0bin, to finalize the second travel command, after the terminal has acknowledged the achievement of position 6 by setting bit SW.3 back to 0bin.
14. Set bit CB.2 of control byte to 1bin, to start the third travel command.
15. Set bit CB.2 of control byte back to 0bin, to finalize the third travel command, after the terminal has acknowledged the achievement of position 7 by setting bit SW.3 back to 0bin.

16. Write the start command 0x0513 (0x0510 + Offset 3) via register communication into Register R7, to set the internal position index to position 3 as next target position.
17. Set bit CB.2 of control byte to 1bin, to start the forth travel command.
18. Set bit CB.2 of control byte back to 0bin, to finalize the fourth travel command, after the terminal has acknowledged the achievement of position 3 by setting bit SW.3 back to 0bin.

    On this, the position index is automatically decremented about one position (to position 2).
19. Set bit CB.2 of control byte to 1bin, to start the fifth travel command.
20. Set bit CB.2 of control byte back to 0bin, to finalize the fifth travel command, after the terminal has acknowledged the achievement of position 2 by setting bit SW.3 back to 0bin.

Write the stop command 0x0540 via register communication into Register R7, to stop the multiple travel command.

2.6.9 Auto-Start Function

The auto-start function can be activated by setting of bit RP0.R32.0 in the feature register and is available in all operation modes.

This function gives the possibility to the user, to make a fast and simple positioning independent of the PLC.

For this, the digital input 1 is used as a preselection for the position set point value. The user has to write the set position into register R2 and R3, like he has to at a standard travel command. Afterwards the enabling does take place by setting of bit CB.4 in the control byte. According to the status of the digital input 1, the stepper motor runs to the following positions, if:

- the digital input 1 = has high potential (24 V), to the position given in R2 and R3
- the digital input 1 = has low potential (0 V), back to position 0

The digital input can change its status during the ride. The terminal recognizes this change immediately, stops the motor and drives into the other direction. The terminal reports the arrival at the target position by setting bit SW.3 of the status word.

DANGER!! The stepper motor is, if standing between position 0 and the upper target position (given by R2 and R3), started immediately, when bit CB.4 of control byte is activated.
Before activating, it is recommended, first to delete the present position and to set the input 1 to low potential.

Ensure that your system state permits this and that hazards for persons or machinery have been ruled out!

2.6.10 Auto-stop Function

The auto-stop function can be activated by setting of bit RP0.R32.1 in the feature register.

This function gives the possibility to the user, to stop the motor automatically (e.g. by 2 limit switches). By setting of bit CB.5 in control byte this function is enabled. If the motor runs against one of the limit switches, creating a rising edge to input 1 or input 2, it is stopped with emergency deceleration (configurable by register RP0.R50). The inputs 1 and 2 can be configured for break contacts via bit R52.14 and R52.15 (this means LOW is the active state). Thereby the motor is also switched off at wire breakage.

2.6.11 Fast-stop Function

The PLC stop function is comparable to the auto-stop function, but this function can be used by the PLC. It doesn't have to be activated explicit in the feature register. By setting of bit CB.1 in feature register the function is enabled and the motor is stopped with emergency deceleration immediately. This can be useful e.g. for an active travel command, that is configured with low deceleration.

2.6.12 Referencing

The stepper motor terminal provides the opportunity of a simple referencing (homing). Therefore the digital inputs are used as cam guide. If only one reference input should be used, the second input can be deactivated via bit RP0.52.4 or RP0.52.5. The exact proceeding is described in detail as an example below.

For useful referencing the following registers can be re-parameterized:

- RP0.R40: maximum acceleration
- RP0.R50: emergency acceleration
- RP0.R53: reference velocity backwards $v_{ref,b}$ (backwards means falling position values)
- RP0.R54: reference velocity forwards $v_{ref,f}$ (forwards means rising position values)
- RP0.R55: reference position (lower value word)
- RP0.R56: reference position (higher value word)

Example

A referencing is proceeded in the following example.

1. Enlist the parameters via register communication into register RP0.R40, RP0.R50, RP0.R53, RP0.R54, RP0.R55 and RP0.R56.
2. Set bit CB.0 of control byte to 1bin, to enable the motor control.
3. Write the command 0x0520 via register communication into Register R7, to prepare the reference journey.
4. Now, the process output data DataOUT has to be zero, so that the referencing can be started.
5. At the beginning of the reference journey both digital inputs are not allowed to be driven, because they are disjunction-linked to each other!
6. Set bit CB.2 of control byte to 1bin, to start the referencing. Now, the stepper motor terminal drives the motor with speed vref,b in negative direction towards the limit switch (in case of bit RP0.R52.0 is set with speed vref,f in positive direction).
7. If the terminal detects the arriving at a limit switch by a positive edge at one of the two inputs, it stops the motor with emergency acceleration.
8. Afterwards the stepper motor terminal drives the motor with speed vref,b in contrary direction.
9. If a falling edge appears at the driven digital input (the motor has driven away from the limit switch), the terminal stops the motor with emergency deceleration.
10. Now, the Terminal reports the successfully finalized referencing by setting bit SW.3 of status word. Simultaneously the reference position configured in register RP0.R55 and RP0.R56 is set as actual position.
11. Set bit CB.2 of control byte back to 0bin, to finalize the referencing.

The terminal acknowledges this by setting bit SW.3 back to 0bin.

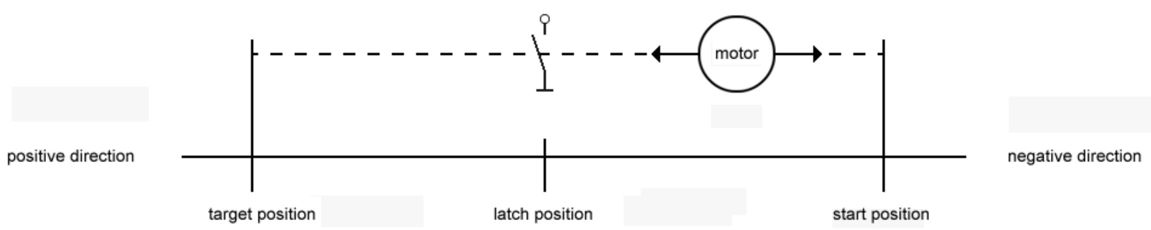
2.6.13 Self Adjusting

The stepper motor terminal provides the opportunity of an easy self adjusting "on the fly". A linear axis offers a useful stage for this.

A switch, mounted at the stretch of way is used as triggering sensor. If it is passed over, it causes a latch event in the terminal (any latch input can be used).

At the first time the latch value is taken as reference position. If the point is passed over again and a latch value is stored, the terminal corrects the difference between the reference position and the value latched at running operation during the next standstill.

The exact proceeding is described in detail as an example below.



Example

A self adjusting is proceeded in the following example.

1. Set bit CB.0 of control byte to 1bin, to enable the motor control.
2. Write the command 0x0530 via register communication into Register R7, to activate the self adjusting.
3. Activate a latch event by using bit CW.0 to CW.4 of control word, to latch the reference position.
4. Write a velocity value to process data word DataOUT, to drive the motor to reference position.
5. The terminal sets bit SW.4 of status word, after the reference position is passed over.
6. Set the velocity value in process data word DataOUT to zero and delete the latch enabling (the reference position is not stored until now).
7. Write a contrary velocity value to process data word DataOUT, to drive the motor to start position again.
8. Arrived at the starting position, activate at control word again the latch event that was referenced with before.
9. Write a velocity value to process data word DataOUT, to drive the motor over the latch point again.
10. Set the velocity value in process data word DataOUT to zero and delete the latch enabling (the present position is corrected now).
11. Write the stop command 0x0540 via register communication into Register R7, to deactivate the self adjusting.

3 MOUNTING AND WIRING

This section provides mounting and wiring information for the Bus Terminal.

Note: For information about configuring the Stepper Module, see the Kollmorgen Automation Suite™ IDE software and online help system.

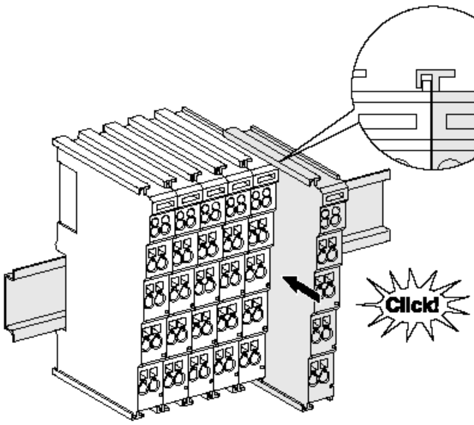
3.1 Installation of Bus Terminals on Mounting Rails

DANGER!! Bring the bus terminal system into a safe, powered down state before starting installation, disassembly or wiring of the Bus Terminals!

3.1.1 Assembly

The Bus Coupler and Bus Terminals are attached to commercially available 35 mm mounting rails (DIN rails according to EN 50022) 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.



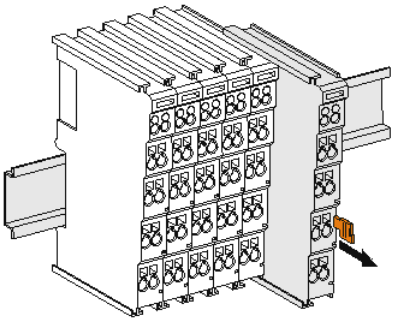
If the Terminals are clipped onto the mounting rail first and then pushed together without tongue and groove, the connection will not be operational! When correctly assembled, no significant gap should be visible between the housings.

Note: During the installation of the Bus Terminals, the locking mechanism of the terminals must not come into conflict with the fixing bolts of the mounting rail.

3.1.2 Disassembly

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

1. Carefully pull the orange-colored lug approximately 1 cm out of the disassembled terminal, until it protrudes loosely. The lock with the mounting rail is now released for this terminal, and the terminal can be pulled from the mounting rail without excessive force.



2. Grasp the released terminal with thumb and index finger simultaneous at the upper and lower grooved housing surfaces and pull the terminal away from the mounting rail.

3.1.3 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 Standard Bus/Performance 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.

Note: 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 interrupt the power contacts and thus represent the start of a new supply rail.

3.1.4 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.

WARNING!! 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).

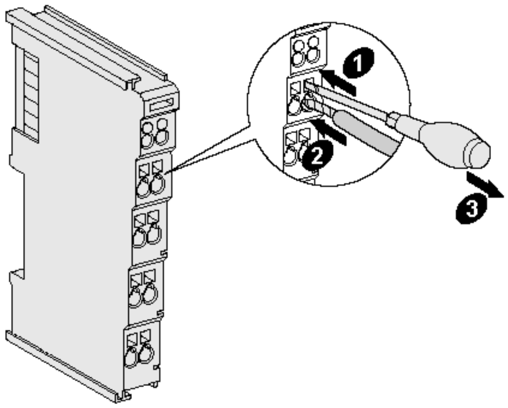
For insulation testing, disconnect the PE supply line at the Bus Coupler or the Power Feed Terminal! In order to decouple further feed points for testing, these Power Feed Terminals can be released and pulled at least 10 mm from the group of terminals.

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

3.1.5 Wiring

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

1. Open a spring-loaded terminal by slightly pushing with a screwdriver or a rod into the square opening above the terminal.



2. The wire can now be inserted into the round terminal opening without any force.
3. The terminal closes automatically when the pressure is released, holding the wire securely and permanently.

Wire Size Width	Wire Stripping Length
0,08 ... 2,5 mm^2	8 mm

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

3.2 AKT-SM-L15-000 Connections

The section describes the connections for the Stepper Module.

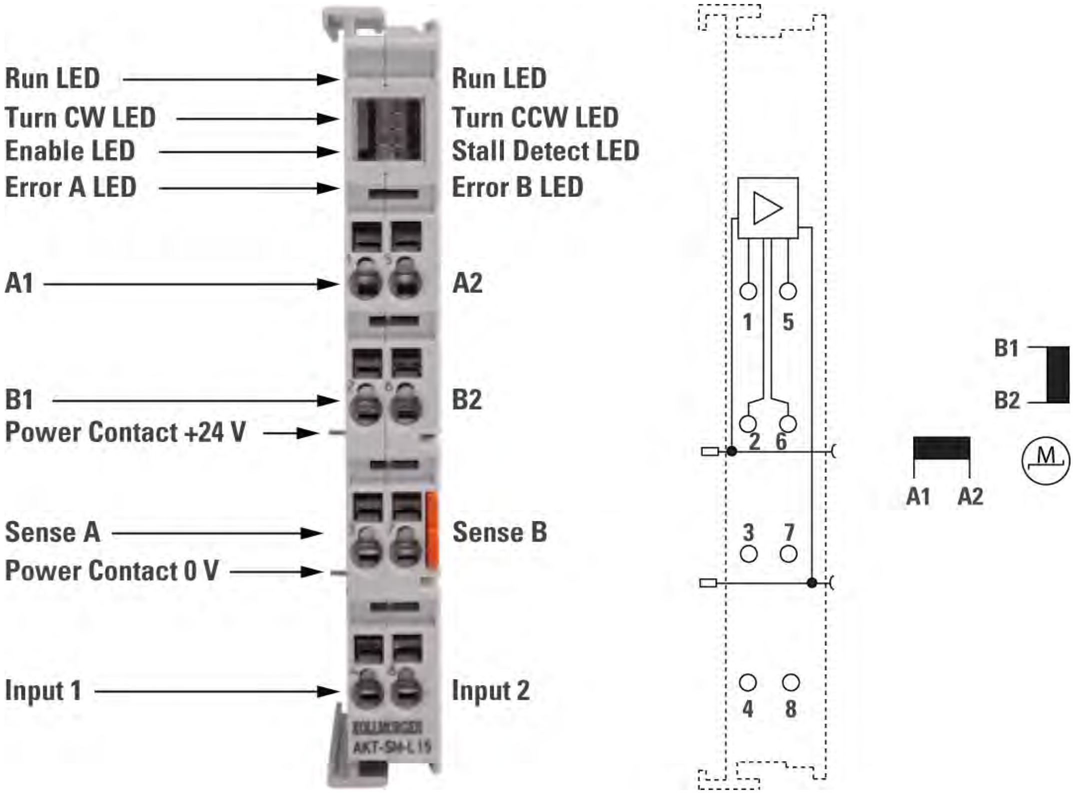


Figure 4.1 Terminal Connections

3.2.1 Terminal Connection Descriptions

The following table provides the terminal descriptions for the Stepper Module.

Terminal	Number	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})
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})

ATTENTION!! Please note the connection examples in the following section.

3.3 AKT-SM-L15-000 Connection Examples

DANGER!! *Bring the bus terminal system into a safe, powered down state before starting installation, disassembly or wiring of the Bus Terminals!*

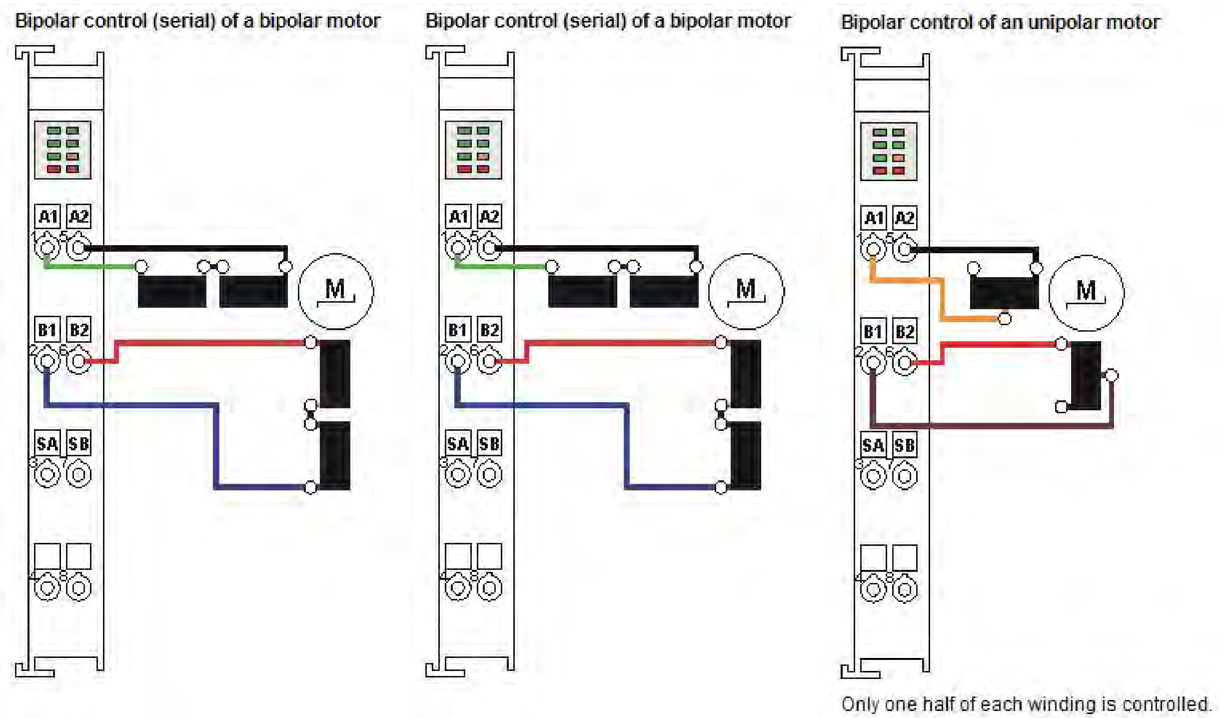
ATTENTION!! *Connect the windings of a motor phase only to terminal points of the same output stage of the stepper motor terminal, e.g.:*

- *one phase to the terminal points A1 and A2,*
- *the other phase to the terminal 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!

3.3.1 Connection Modes

The AKT-SM-L15-000 Stepper Module has bipolar output stages and is able to control bipolar and unipolar motors.



3.4 AKT-SM-L50-000 Connections

The section describes the connections for the Stepper Module.

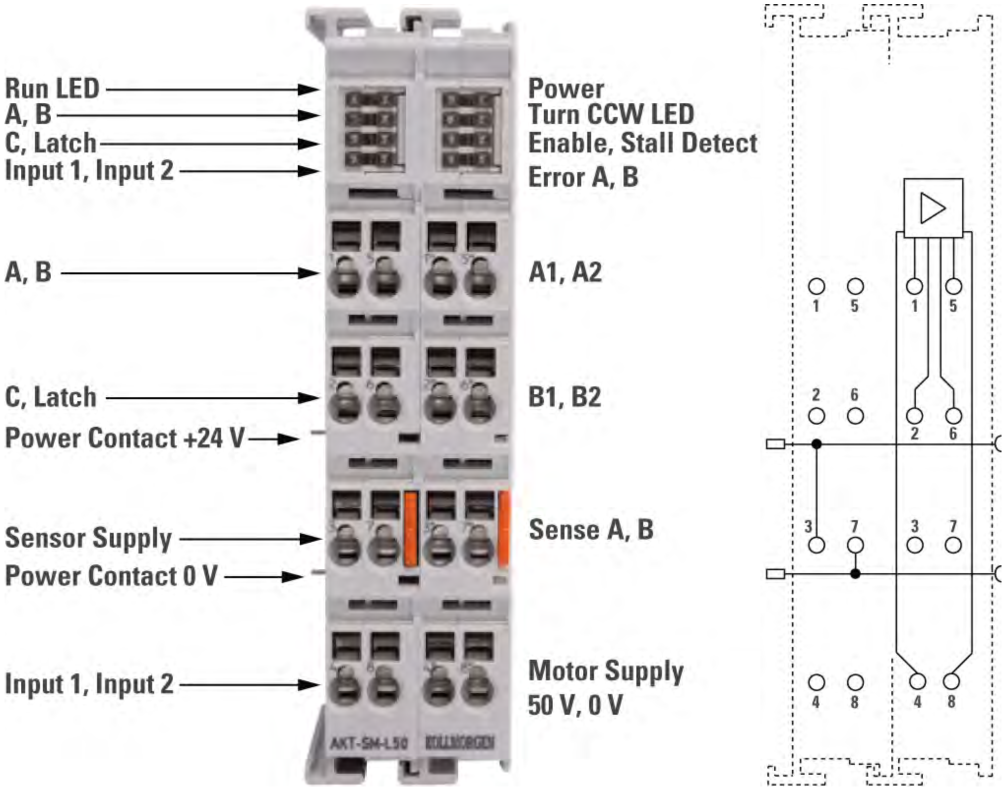


Figure 3.1 Terminal Connections

3.4.1 Terminal Connection Descriptions (Left)

The following table provides the terminal descriptions for the Stepper Module.

Terminal	Number	Connection
A	1	Encoder input A
C	2	Encoder input C
Encoder supply +24 V	3	Encoder supply (from positive power contact)
Input 1	4	Digital input 1 (24 V_{DC})
B	5	Encoder input B
Latch/Gate	6	Latch input
Encoder supply 0 V	7	Encoder supply (from negative power contact)
Input 2	8	Digital input 2 (24 V_{DC})

3.4.2 Terminal Connection Descriptions (Right)

The following table provides the terminal descriptions for the Stepper Module.

Terminal	Number	Connection
A1	1’	Motor winding A
B1	2’	Motor winding B
Sense A	3’	Motor winding A
Motor supply	4’	Feed 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’	Feed for output stages (0 V_{DC})

ATTENTION!! Please note the connection examples in the following section.

3.4.3 Power Contacts

The Power Contacts (+24 V_{DC}) supply the following loads:

- Incremental Encoder (terminal point 3 and 7)
- Digital inputs (terminal point 4 and 8)
- Output drivers of stepper motor terminal

ATTENTION!! The voltage for the power contacts has to be activated already when the Standard-Bus voltage is switched on, so that internal circuits (output drivers) can be initialized.

If this is not possible because of application conditions (e. g. supply is routed via a circuit with emergency stop switch), you have to do a software reset for the terminal after booting the system.

DANGER!! If the Standard-Bus voltage (5V) fails, the motor controller is not set back! This means, if the motor is in motion, it is not stopped!

3.5 AKT-SM-L50-000 Connection Examples

DANGER!! *Bring the bus terminal system into a safe, powered down state before starting installation, disassembly or wiring of the Bus Terminals!*

ATTENTION!! *Connect the windings of a motor phase only to terminal points of the same output stage of the stepper motor terminal, e.g.:*

- *one phase to the terminal points A1 and A2,*
- *the other phase to the terminal 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!

3.5.1 Connection Modes

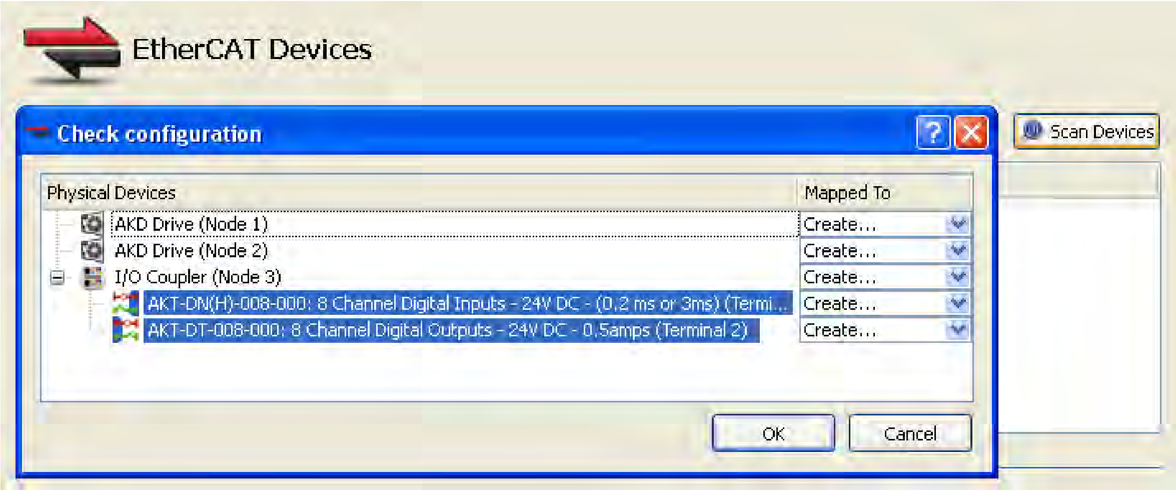
The AKT-SM-L50-000 Stepper Module has bipolar output stages and is able to control bipolar and unipolar motors.

4 AUTOMATIC CONFIGURATION

This chapter describes the basics of automatic configuration within the KAS Integrated Development Environment (IDE).

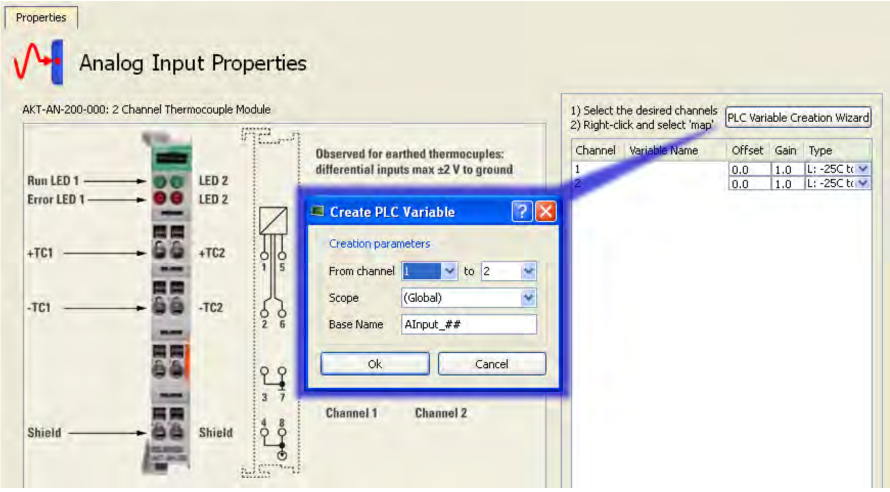
4.1 Scan Device

For ease-of-use the KAS IDE Scan Device feature provides automatic integration of I/O devices. This allows you to automatically locate and add I/O terminals to the application project:



4.2 Setting I/O Values

After the I/O slice is mapped it can be selected in the application project and the offset and gain values can be set. Additionally, the IDE allows you to map the I/O points to variables in your application:

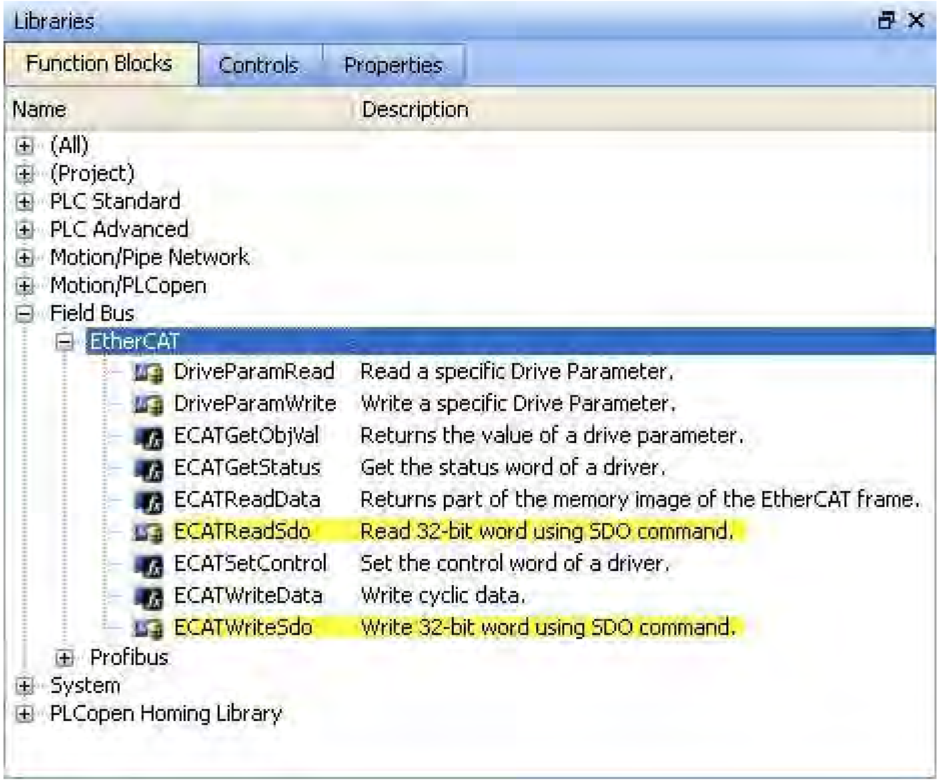


For more detailed information on these procedures refer to the section “EtherCAT Scan Device” in the KAS IDE online help.

5 MANUAL CONFIGURATION

Kollmorgen strongly recommends automatic configuration using the KAS IDE over manual configuration. For automatic configuration refer to chapter 5. Manual configuration is for advanced procedures only. The following sections provide information on:

1. Advanced configuration settings that can be made on Registers within this I/O module using EtherCAT Read and Write SDO function blocks in the application project code. These function blocks are located as follows:



2. Manually mapping this I/O block into a KAS project. Manual mapping requires an additional configuration tool. Contact Kollmorgen for more information.

5.1 Process Image

The AKT-SM-L15-000 and AKT-SM-L50-000 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 AKT-SM-L15-000 and AKT-SM-L50-000 are represented with their complete process image.

Byte Offset (without word alignment)	Byte Offset (with word alignment*)	Format	Input Data	Output Data
0	0	Byte	SB	CB
1	2	Word	DataIN	DataOUT
3	4	Word	SW	CW

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

Key

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

5.1.1 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.

<table>
<tr><th>Operation Mode</th><th>DataIN</th><th>DataOUT</th></tr>
<tr><td>Velocity, direct</td><td rowspan="4">Actual position (lower word).
The full actual position (32 bit) can be read from registers R0 and R1 via register communication.</td><td>Velocity value (15 bits and the sign), see formula.</td></tr>
<tr><td>Velocity with ramps</td><td>Velocity value (11 bits and the sign), see formula. The terminal only analyses bits 4 to 15.</td></tr>
<tr><td>Position tracking</td><td>Set position (15 bits and the sign)</td></tr>
<tr><td>Path control</td><td>Manual operation:
Velocity value (11 bits and the sign) for manual mode, see formula. The terminal only analyses bits 4 to 15. In order to be able to manually specify a velocity value, no travel command must be active (CB.2 = 0)!

Travel command:
The set position (32 bits) for the travel command is entered in registers R2 and R3 via register communication. For starting the travel command DataOUT must be 0!</td></tr>
</table>

5.2 Control and Status Byte

This section describes the control and status bytes for the I/O Terminals.

5.2.1 Process Data Mode

Control byte (for process data mode)

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

Bit	CB.7	CB.6	CB.5	CB.4	CB.3	CB.2	CB.3	CB.0
Name	RegAccess	Reset	AutoStop	AutoStart	Reduced Current	Start	FastStop	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 active
CB.4	AutoStart[2]	1_{bin}	The automatic Start Function is active
CB.3	Reduced Current (identical to bit CW.11)	1_{bin}	Reduced coil current (when the motor is at standstill) is active (specified via R45)
CB.2	Start[2]	0_{bin}	Travel command is stopped
		1_{bin}	Travel command is started
CB.1	FastStop[2]	1_{bin}	The motor is stopped with emergency acceleration (R50) (rising edge)
CB.0	Enable	0_{bin}	Disable motor control
		1_{bin}	Disable motor control

1. In all operation modes but *Position tracking* available.
2. Only in operation mode *Path control* available.

Status byte (for process data mode)

The status byte (SB) is located in the input image, 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)
SB.5	Warning	1_{bin}	Overtemperature was detected (SW.8)
SB.4	IDIE	0_{bin}	Terminal executes a job currently (motor is running or travel command is active)
		1_{bin}	Terminal has no job (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)

5.2.2 Register Communication

Control byte (for register communication)

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

Bit	CB.7	CB.6	CB.5	CB.4	CB.3	CB.2	CB.3	CB.0
Name	RegAccess	R/W	Register number					

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	Register number	Register number: Enter the number of the register that you: - Want to read with input data word DataIn or - Want to write with output data word DataOut.	

Status byte (for register communication)

The status byte (SB) is located in the input image, 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	Register number					

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	Register number	Number of the register that was read or written.	

5.3 Control and Status Word

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

5.3.1 Control Word

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	Toggle LatchData	Get LatchData	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 process data from *set-velocity* to *set-position* (see function specify set-position)
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 via R45) is active when the motor is running

Bit	Name	Description	
CW.11	ReducedCurrent (same function as bit CB.3)	1_{bin}	Reduced coil current (specified via R45) is active when the motor is at standstill
CW.10	SetPos	1_{bin}	Position value is set with registers R2 and R3 (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 1 switched-through transistor per half-wave
CW.8	GetEncoder Position[1, 2]	1_{bin}	Display the present encoder position into the process image
CW.7	GetPositionError[1]	0_{bin}	Display the present position in Byte 2 and 3 of input process data
		1_{bin}	Display the relative position error in Byte 2 and 3 of input process data
CW.6	ToggleLatchData	0_{bin} 1_{bin}	If this bit is changed, the saved latch values - up to 20 - (R37) can be retrieved if the latch array (R32.9) is activated and CW.5 is set
CW.5	GetLatchData	0_{bin}	Show the current position in byte2,3 of the input process data
		1_{bin}	Show the current latch value 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 is enabled (for latch inputs with rising edge, the terminal saves the current position) ATTENTION: Higher priority than CW.1, CW.2 and CW.4!
CW.2	enLatchInput2	1_{bin}	External latch event is enabled (at rising edge of digital input 1, the terminal stores the present position)
CW.1	enLatchInput1	1_{bin}	External latch event is enabled (at rising edge of digital input 2, the terminal stores the present position) 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 at internal positioning
2. Only for AKT-SM-L50-000

5.3.2 Status Word

The status word (SW) is located in the input image, 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 InByte2,3	LatchValid	Target Reached	SetPos Ready	Input 2	Input 1

Legend

Bit	Name	Description	
SW.15	ConfigError	1_{bin}	Configuration error (the error A LED is set). Cause: • Value for full motor steps is 0 (R33), only AKT-SM-L50-000 • Value for encoder increments is 0 (R34), only AKT-SM-L50-000 • Value for min. velocity v_{min} is 0 (R38) • Value for max. velocity v_{max} is 0 (R39) • Value for max. acceleration a_{max} is 0 (R40) • Value for coil current IS for $a > a_{th}$ is 0 (R42) • Value for coil current IS for $a \leq a_{th}$ is 0 (R43) The error display is deleted automatically, if the wrong configured registers are written with correct data.
SW.14	NoControl Power	0_{bin}	Breakdown of control power voltage (error B LED flashes, and the terminal switches the motor off automatically). The terminal is reinitialized automatically if the control power voltages returns. **ATTENTION:** the acknowledgement of this error message has to be done by the user, by setting bit CB.6 of the control byte!
SW.13	OverCurrent Br B	1_{bin}	Overcurrent, bridge B (*error B* LED flashes, and the terminal switches the motor off automatically) **ATTENTION:** the acknowledgement of this error message has to be done by the user, by setting bit CB.6 of the control byte!

Bit	Name	Description	
SW.12	OverCurrent Br A	1_{bin}	Overcurrent, bridge A (*error B* LED flashes, and the terminal switches the motor off automatically) **ATTENTION:** the acknowledgement of this error message has to be done by the user, by setting bit CB.6 of the control byte!
SW.11	OpenLoad Br B	1_{bin}	Open load, bridge B (*error B* LED flashes, and the terminal switches the motor off automatically) **ATTENTION:** the acknowledgement of this error message has to be done by the user, by setting bit CB.6 of the control byte!
SW.10	OpenLoad Br A	1_{bin}	Open load, bridge A (*error B* LED flashes, and the terminal switches the motor off automatically) **ATTENTION:** the acknowledgement of this error message has to be done by the user, by setting bit CB.6 of the control byte!
SW.9	UnderVoltage	1_{bin}	Supply voltage too low (*error B* LED flashes, and the terminal switches the motor off automatically) **ATTENTION:** the acknowledgement of this error message has to be done by the user, by setting bit CB.6 of the control byte!
SW.8	OverTemperature	1_{bin}	Overtemperature (see SB.5) - the temperature of stepper-controller-chip is higher than 125 °C (*error B* LED flashes, and the terminal switches the motor off automatically). **ATTENTION:** the acknowledgement of this error message has to be done by the user, by setting bit CB.6 of the control byte!
SW.7	EncoderDisabled	0_{bin}	The encoder is enabled
		1_{bin}	The encoder is disabled (feature bit R32.12 = 1 and latch input = 0 or R32.13 = 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 was modified and SW.6 has taken on the new state (acknowledgement for ToggleLatchData)
		1_{bin}	
SW.5	LatchDataInByte2,3	0_{bin}	The current position is shown in byte 2, 3 of the input process data
		1_{bin}	The last latch value is shown in byte 2, 3 of the input process data (acknowledgement for GetLatchData)
SW.4	LatchValid	1_{bin}	A latch event has occurred (for CW. 0 = 1, CW.3 = 1 or CW.4 = 1); if the latch array is activated (R32.9), this bit is set once the number of latch values specified in R37 has been stored
SW.3	TargetReached[3]	1_{bin}	Target position is reached

Bit	Name	Description	
SW.2	SetPosReady	1_{bin}	The current position was deleted (for CW.13 = 1) or set (for CW.10 = 1) (acknowledgement for ClearPos and SetPos)
SW.1	Input 2	State of digital input 2. Evaluation of digital input 2 can be reversed with bit RP0.R52.15 of feature register 2.	
SW.0	Input 1	State of digital input 1. Evaluation of digital input 1 can be reversed with bit RP0.R52.14 of feature register 2.	

3. Only in operation mode *Path.*

5.3.3 Register Overview

The registers are used for the parameterization 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 Number	Comment	Default Value		R/W	Memory
R0	Actual position (low-order word)	-	-	R	RAM
R1	Actual position (high-order word)	-	-	R	RAM
R2	Set target position or position (low-order word)	-	-	R/W	RAM
R3	Set target position or position (high-order word)	-	-	R/W	RAM
R4	Register page selection register (range of values: 0 to 2)	0x0000	0_{dec}	R/W	RAM
R5	Latch value (high-order word)	-	-	R	RAM
R6	Temperature register	e.g. 0x003C	e.g. 60_{dec}	R	RAM
R7	Command register	0x0000	0_{dec}	R/W	RAM
R8	Terminal type	0x09E3	2531_{dec}	R	ROM
		0x09ED	2541_{dec}		
R9	Firmware revision level	e.g. 0x3141	e.g. $1A_{ASCII}$	R	ROM

Register Number	Comment	Default Value		R/W	Memory
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	Hardware version number	e.g. 0x0000	e.g. 0_{dec}	R/W	EEPROM
R17	Reserved	-	-	-	-
…	…	…	…	…	…
R30	Reserved	-	-	-	-
R31	Code word register	0x0000	0_{dec}	R/W	RAM
R32 to R63	Register for showing register page 0, register page 1, register page 2 or register page 3				

Register page 0 (further parameters)

Register Number	Comment	Default Value		R/W	Memory
R32	Feature register 1	0x0000	0_{dec}	R/W	EEPROM
R33	Full motor steps	0x00C8	200_{dec}	R/W	EEPROM
R34	Encoder increments (AKT-SM-L50-000 only)	0x0FA0	4000_{dec}	R/W	EEPROM
R35	Maximum coil current A	0x0064	100_{dec}	R/W	EEPROM
R36	Maximum coil current B	0x0064	100_{dec}	R/W	EEPROM
R37	Number of latch values	0x0014	20_{dec}	R/W	RAM / EEPROM
R38	Min. velocity v_{min}	0x000A	10_{dec}	R/W	RAM / EEPROM
R39	Max. velocity v_{max}	0x07FF	2047_{dec}	R/W	RAM / EEPROM

Register Number	Comment	Default Value		R/W	Memory
R40	Max. acceleration a_{max}	0x07FF	2047_{dec}	R/W	RAM / EEPROM
R41	Acceleration threshold a_{th}	0x03FF	1023_{dec}	R/W	EEPROM
R42	Coil current I_S, $a > a_{th}$	0x0064	100_{dec}	R/W	EEPROM
R43	Coil current I_S, $a \leq a_{th}$	0x0064	100_{dec}	R/W	EEPROM
R44	Coil current I_S, $v = 0$ (automatic)	0x0032	50_{dec}	R/W	EEPROM
R45	Coil current I_S, (manual)	0x0032	50_{dec}	R/W	EEPROM
R46	Step size per quarter period	0x0006	6_{dec}	R/W	EEPROM
R47	Load angle threshold	0x0007	0_{dec}	R/W	RAM / EEPROM
R48	Kp factor	0x0000	0_{dec}	R/W	RAM / EEPROM
R49	Kv factor	0x0000	0_{dec}	R/W	RAM / EEPROM
R50	Emergency acceleration a_e	0x07FF	2047_{dec}	R/W	RAM / EEPROM
R51	Number of impulses per revolution	0x0001	1_{dec}	R/W	REEPROM
R52	Feature-Register 2	0x0000	0_{dec}	R/W	REEPROM
R53	Referencing speed backward $v_{ref,b}$	0x01F4	500_{dec}	R/W	REEPROM
R54	Referencing speed forward $v_{ref,f}$	0x0032	50_{dec}	R/W	EEPROM
R55	Referencing position (lower value word)	0x0000	0_{dec}	R/W	EEPROM
R56	Referencing position (higher value word)	0x0000	0_{dec}	R/W	EEPROM
R57	Encoder tolerance (only AKT-AM-L50-000)	0x0000	0_{dec}	R/W	EEPROM
R58	max. deceleration a_{dec}	0x07FF	2047_{dec}	R/W	RAM
R59	Reserved	-	-	-	-
…	…	…	…	…	…
R63	Reserved	-	-	-	-

Register page 1 (user current table)

Register Number	Comment	Default Value		R/W	Memory
R32	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	Current value 64 to 61	0xFFFF	65535_{dec}	R/W	EEPROM
R48	Reserved	-	-	-	-
…	…	…	…	…	…
R63	Reserved	-	-	-	-

Register-Page 2 (user positions table)

Register Number	Comment	Default Value		R/W	Memory
R32	Target position 0 (lower-value word) (start value for register 2)	0x0000	0_{dec}	R/W	EEPROM
R33	Target position 0 (higher-value word) (start value for register 3)	0x0000	0_{dec}	R/W	EEPROM
R34	Target position 1 (lower-value word)	0x0000	0_{dec}	R/W	EEPROM

Register Number	Comment	Default Value		R/W	Memory
R35	Target position 1 (higher-value word)	0x0000	0_{dec}	R/W	EEPROM
…	…	…	…	…	…
…	…	…	…	…	…
R62	Target position 15 (lower-value word)	0x0000	0_{dec}	R/W	EEPROM
R63	Target position 15 (higher-value word)	0x0000	0_{dec}	R/W	EEPROM

Register-Page 3 User speeds- and acceleration table

Register Number	Comment	Default Value		R/W	Memory
R32	Velocity 0 (start value for register 39)	0x07FF	2047_{dec}	R/W	EEPROM
R33	Velocity 1	0x07FF	2047_{dec}	R/W	EEPROM
…	…	…	…	…	…
R47	Velocity 15	0x07FF	2047_{dec}	R/W	EEPROM
R48	Acceleration 0 (start value for register 40)	0x07FF	2047_{dec}	R/W	EEPROM
R49	Acceleration 1	0x07FF	2047_{dec}	R/W	EEPROM
…	…	…	…	…	EEPROM
R63	Acceleration 15	0x07FF	2047_{dec}	R/W	EEPROM

5.3.4 Register Description

All registers can be read or written via register communication. They are used for parameterzing 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).

Note:

In running operation, the register R37, R38, R39, R40, R47, R48, R49 and R50 should only be written to via the RAM area. Therefore the code word mustn't be set in register R31 (if the code word is set, it is written into the EEPROM area)! The EEPROM area should only be used for initialization after a terminal reset (caused by software or power off).

Frequent changes of the EEPROM-Register can damage the memory cells. Today's EEPROM chips are designed for many writing cycles, but by multiple positioning with different speeds and accelerations, this number can be reached.

If, for example, a travel command is started with the maximum velocity v1, a new maximum velocity v2 may be entered in register R39 and a new target position in registers R2 and R3 during the final phase of the motion. Once the target position is reached and control bit CB.2 and the process output data are both zero, the terminal calculates the new parameters for the next travel command.

R0: Actual Position (low-order word)

Contains the low-order word of the actual position.

At activated latch function (RP0.R52.2 = 1bin), this register has to be read first. At this register R0 und R1 are buffered. At the following reading of register R1, both registers are unblocked again.

R1: Actual position (high-order word)

Contains the high-order word of the actual position.

R2: Set target position or position (low-order word)

You can specify the low-order word of the required position here.

R3: Set target position or position (high-order word)

You can specify the high-order word of the required position here.

R4: Register page selection register

This register is used for specifying which register page is shown in registers R32 to R63 of the AKT-SM-L15-000 and AKT-SM-L50-000:

- 0x0000: Register page 0 - further terminal parameters (default)
- 0x0001: Register page 1 - current values of the user current table
- 0x0002: Register-Page 2 - user positioning 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.

R6: Temperature register

The internal temperature of the terminal can be read out from register R6. The terminal sets the bit SB.5 as a warning, if the temperature oversteps the threshold of 80 °C. If the temperature falls under 60 °C, SB.5 is set back automatically.

R7: Command register

Standard commands

For a standard command to be executed, it is first necessary for the user code word, 0x1235, to be entered into register R31.

Command 0x7000: Restore Factory Settings

Entering 0x7000 in register R7 restores the factory settings for the following registers:

Register page 0 (further parameters):
R32: 0
R33: 200
R34: 4000
R35: 100
R36: 100
R37: 20
R38: 10
R39: 0x07FF
R40: 0x07FF
R41: 0x03FF
R42: 100
R43: 100
R44: 50
R45: 50
R46: 6
R47: 7
R48: 0
R49: 0
R50: 2047
R51: 1
R52: 0
R53: 500
R54: 50
R55: 0
R56: 0
R57: 0
R58: 2047

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

Register page 2 (position table):
R32-R63: 0

Register page 3 (velocity and acceleration table):
R32-R63: 2047

Command 0x8000: Software Reset

Entering 0x8000 in register R7 initiates a full software reset for the terminal. All internal variables (position, latch values, errors, etc.) are deleted or set to new values calculated from stored EEPROM values. The internal circuitries (DAC, output drivers) are newly initialized by a software reset.

DANGER!! 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

For a special command to be executed, it is not necessary to enter any code word into register R31.

Command 0x0500: Start multi travel command 'forward'

Entering 0x0500 a multi travel command is started forwards. Up to 16 positions specified by a positioning table can be approached. The start position is specified by an offset added to the command code (e.g. for start position 4 use command code 0x0504). The positioning table is traversed forwards.

Command 0x0510: Start multi travel command 'backward'

Entering 0x0510 a multi travel command is started backwards. Up to 16 positions specified by a positioning table can be approached. The start position is specified by an offset added to the command code (e.g. for start position 8 use command code 0x0518). The positioning table is traversed forwards.

Command 0x0520: Start referencing

Entering 0x0520 an automatically referencing (homing) of the stepper motor is started. For this, the digital inputs act as limit switches.

Command 0x0530: Start self adjusting

Entering 0x0530 an automatically self adjusting of position is started.

Command 0x0540: Stop special commands or put back

Entering 0x0540 the presently activated special command is stopped.

R8: Terminal type

The description of the terminal is contained in register R8:

AKT-SM-L15-000: 0x09E3 (2531_{dec})

AKT-SM-L50-000: 0x09ED (2541_{dec})

R9: Firmware revision level

Register R9 contains the ASCII coding of the terminal's firmware revision level, e.g. 0x3141 = '1A'. The '0x31' corresponds here to the ASCII character '1', while the '0x41' represents the ASCII character 'A'.

This value can not be changed.

R16: Hardware version number

Register R16 contains the hardware version of the terminal.

R31: Code word register

- If you write into the user registers without first entering the user code word (0x1235) into the code word register, the terminal will not accept the supplied data.
- If you write values into the user registers and have previously entered the user code word (0x1235) in the code word register, these values are stored in the RAM registers and in the SEEPROM registers and are therefore retained if the terminal is restarted.
- The register R37, R38, R39, R40, R47, R48, R49, R50 and R58 can also be written to without code word. At this, the new values are only stored to the RAM.

The code word is reset if the terminal is restarted.

5.3.5 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	enNegGate Lock	Feedback Type	enSignal Display	enMulti Latch Values	enLatch EventClrPos

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

Legend

Bit	Name	Description		Default
R32.15	enCounterMode (AKT-SM-L50-00 only)	0_{bin}	Encoder mode active	0_{bin}
		1_{bin}	Counter mode active	0_{bin}
R32.14	disEncFltr (AKT-SM-L50-00 only)	1_{bin}	Input filter for encoder signals deactivated	0_{bin}
R32.13	enPosGateLock (AKT-SM-L50-00 only)	1_{bin}	A positive level at the latch/gate input locks the counter.	0_{bin}
R32.12	enNegGateLock (AKT-SM-L50-00 only)	1_{bin}	A negative level at the latch/gate input locks the counter.	0_{bin}
R32.11	FeedbackType (AKT-SM-L50-00 only)	0_{bin}	Use encoder for position feedback	0_{bin}
		1_{bin}	Use internal counter for position feedback	
R32.10	enSignalDisplay (AKT-SM-L50-00 only)	1_{bin}	The C and gate signals are shown in the status word	0_{bin}
R32.9	enMultiLatchValues (AKT-SM-L50-00 only)	0_{bin}	Only one latch value is saved.	0_{bin}
		1_{bin}	Several latch values are saved. The number is specified in register R37.	
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	

Bit	Name	Description		Default
R32.6	invPosFeedback	1_{bin}	Inverts the position feedback. This option is only intended for the operation mode path control, if an external encoder and the stepper motor are mounted fronting to each other (the encoder for example shows negative rotation at positive rotation of the motor). If this option is activated with internal position feedback or the position detection of the external encoder is wrong, the path control works incorrect.	0_{bin}
R32.5	RampType	0_{bin}	Linear ramps	0_{bin}
		1_{bin}	Exponential ramps (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}	Path 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 deactivated	
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 (AKT-SM-L50-000)

Enter the number of increments issued by the encoder connected to the AKT-SM-L50-000 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):

AKT-SM-L15-000: 100% = 1.5 A

AKT-SM-L50-000: 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):

AKT-SM-L15-000: 100% = 1.5 A

AKT-SM-L50-000: 100% = 5.0 A

R37: Number of latch values

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

R38: Min. velocity vmin

This register specifies the maximum cutoff velocity for the operating modes velocity, with ramps and path control (default: 10dec, maximum: 2047dec).

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 vmax

This register specifies the maximum velocity for the operating modes velocity, with ramps and path control (default: 0x07FF, maximum: 0x07FF).

R40: Max. acceleration amax

This register specifies the maximum acceleration for the operating modes velocity, with ramps and path control (default: 0x07FF, maximum: 0x07FF).

R41: Acceleration threshold ath

This register specifies the acceleration threshold for the operating modes velocity, with ramps and path control (default: 0x03FF, maximum: 0x07FF).

R42: Coil current, a > ath (in %)

This register specifies the coil current between 0 and 100 % of the set terminal coil current when a > ath (default: 100%).

R43: Coil current, a ≤ ath (in %)

This register specifies the coil current between 0 and 100 % of the set terminal coil current when a ≤ ath (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 or CW.11 is set, the coil currents for stand-still and running 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!

R46: Step size

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

Register Value	0x0006	0x0005	0x0004	0x0003	0x0002	0x0001	0x0001
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 from which the Stall Detect LED starts flashing and the status bit SB.5 is set. (default: 7, maximum: 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 NC cyclic position interface in position tracking mode (in preparation) (default: 8192, maximum: 65535).

R49: Kv factor

This register specifies the Kv factor for the NC cyclic position interface in position tracking mode (in preparation) (Default: 8192, maximum: 65535).

R50: Emergency acceleration ae

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

R51: Number of impulses per revolution

This register specifies the number of impulses per revolution for the position error detection (default: 1).

R52: Feature-Register 2

The feature register 2 specifies more of the terminal's configuration.

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	-	-	disReference Input2	disReference Input1	ShowIdle	LatchRegister Position	enDecelaration Ramp	revHoming Direction

Legend

Bit	**Name**	**Description**		**Default**
R52.15	revDigInput2	0_{bin}	Evaluate digital Input 2 as a make contact: HIGH-level = active state (SW.1=1_{bin}).	0_{bin}
		1_{bin}	Evaluate digital Input 2 as a brake contact: LOW-level = active state (SW.1 = 1_{bin}).	
R52.14	revDigInput1	0_{bin}	Evaluate digital Input 1 as a make contact: HIGH-level = active state (SW.0 = 1_{bin}).	0_{bin}
		1_{bin}	Evaluate digital Input 1 as a brake contact: LOW-level = active state (SW.0 = 1_{bin}).	
R52.13	enManEnc Tolerance (AKT-SM-L50-000 only)	0_{bin}	The tolerance of the target position is given from the quotient of register R34 / R33 (position feedback with encoder).	0_{bin}
		1_{bin}	The tolerance of the target position is given from register R57.	
R52.12	enClrProcess Data	1_{bin}	The input process data is set to zero in case of watchdog error	0_{bin}
R52.6 - R52.11	-	Reserved		
R52.5	disReferenceInput2	1_{bin}	Disable digital input 2 as reference input.	0_{bin}

Bit	Name	Description		Default
R52.4	disReferenceInput1	1_{bin}	Disable digital input 1 as reference input.	0_{bin}
R52.3	ShowIdle	1_{bin}	Idle bit SB.4 of status byte activated.	0_{bin}
R52.2	LatchRegisterPosition	1_{bin}	Latching of positions register R0 and R1 activated.	0_{bin}
R52.1	enDecelaration Ramp	0_{bin}	The acceleration and the deceleration ramp is given by register R40.	0_{bin}
		1_{bin}	The deceleration ramp is given by register R58	
R52.0	revHoming Direction	0_{bin}	Drive with negative velocity towards the limit switch ($V_{ref,f}$ < 0), Drive with positive velocity away from the limit switch ($V_{ref,b}$ > 0)	0_{bin}
		1_{bin}	Drive with positive velocity towards the limit switch ($V_{ref,f}$ > 0), Drive with negative velocity away from the limit switch ($V_{ref,b}$ < 0)	0_{bin}

R53: Referencing speed backward vref,b

This register specifies the speed for reference function (homing), which the terminal drives with towards the limit switch (default: 500, maximum: 2047).

Backward means, the terminal drives the motor with falling position values.

R54: Referencing speed forward vref,f

This register specifies the speed for reference function (homing), which the terminal drives with away from the limit switch (default: 50, maximum: 2047).

Forward means, the terminal drives the motor with rising position values.

R55: Referencing position (lower value word)

Here you can specify the lower value word of the referencing position (default: 0, maximum: 65535).

R56: Referencing position (higher value word)

Here you can specify the higher value word of the referencing position (default: 0, maximum: 65535).

R57: Encoder tolerance (only AKT-SM-L50-000)

This register specifies (at position feedback via encoder) the tolerance of the target journey, the position has to lie within, where bit SW.3 is set (default: 0, maximum: 65535).

R58: max. deceleration adec

This register specifies the maximum deceleration for the operation modes Speed with, Ramps and Path Control (default: 0, maximum: 2047).

5.3.6 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).

R34: Current values 12 to 9

This register specifies current values 12, 11, 10 and 9 (structure as current values 4 to 1).

R35: Current values 16 to 13

This register specifies current values 16, 15, 14 and 13 (structure as current values 4 to 1).

R36: Current values 20 to 17

This register specifies current values 20, 19, 18 and 17 (structure as current values 4 to 1).

R37: Current values 24 to 21

This register specifies current values 24, 23, 22 and 21 (structure as current values 4 to 1).

R38: Current values 28 to 25

This register specifies current values 28, 27, 26 and 25 (structure as current values 4 to 1).

R39: Current values 32 to 29

This register specifies current values 32, 31, 30 and 29 (structure as current values 4 to 1).

R40: Current values 36 to 33

This register specifies current values 36, 35, 34 and 33 (structure as current values 4 to 1).

R41: Current values 40 to 37

This register specifies current values 40, 39, 38 and 37 (structure as current values 4 to 1).

R42: Current values 44 to 41

This register specifies current values 44, 43, 42 and 41 (structure as current values 4 to 1).

R43: Current values 48 to 45

This register specifies current values 48, 47, 46 and 45 (structure as current values 4 to 1).

R44: Current values 52 to 49

This register specifies current values 52, 51, 50 and 49 (structure as current values 4 to 1).

R45: Current values 56 to 53

This register specifies current values 56, 55, 54 and 53 (structure as current values 4 to 1).

R46: Current values 60 to 57

This register specifies current values 60, 59, 58 and 57 (structure as current values 4 to 1).

R47: Current values 64 to 61

This register specifies current values 64, 63, 62 and 61 (structure as current values 4 to 1).

5.3.7 Register Page 2 (User Positioning Table)

Register page 2 specifies the target positions for the user positioning table.

R32: Target position 0 (lower value word - start value for R2)

This register specifies the lower value word of target position 0.

R33: Target position 0 (higher value word - start value for R3)

This register specifies the higher value word of target position 0.

R34: Target position 1 (lower value word)

This register specifies the lower value word of target position 1.

R35: Target position 1 (higher value word)

This register specifies the higher value word of target position 1.

R36: Target position 2 (lower value word)

This register specifies the lower value word of target position 2.

R37: Target position 2 (higher value word)

This register specifies the higher value word of target position 2.

R38: Target position 3 (lower value word)

This register specifies the lower value word of target position 3.

R39: Target position 3 (higher value word)

This register specifies the higher value word of target position 3.

R40: Target position 4 (lower value word)

This register specifies the lower value word of target position 4.

R41: Target position 4 (higher value word)

This register specifies the higher value word of target position 4.

R42: Target position 5 (lower value word)

This register specifies the lower value word of target position 5.

R43: Target position 5 (higher value word)

This register specifies the higher value word of target position 5.

R44: Target position 6 (lower value word)

This register specifies the lower value word of target position 6.

R45: Target position 6 (higher value word)

This register specifies the higher value word of target position 6.

R46: Target position 7 (lower value word)

This register specifies the lower value word of target position 7.

R47: Target position 7 (higher value word)

This register specifies the higher value word of target position 7.

R48: Target position 8 (lower value word)

This register specifies the lower value word of target position 8.

R49: Target position 8 (higher value word)

This register specifies the higher value word of target position 8.

R50: Target position 9 (lower value word)

This register specifies the lower value word of target position 9.

R51: Target position 9 (higher value word)

This register specifies the higher value word of target position 9.

R52: Target position 10 (lower value word)

This register specifies the lower value word of target position 10.

R53: Target position 10 (higher value word)

This register specifies the higher value word of target position 10.

R54: Target position 11 (lower value word)

This register specifies the lower value word of target position 11.

R55: Target position 11 (higher value word)

This register specifies the higher value word of target position 11.

R56: Target position 12 (lower value word)

This register specifies the lower value word of target position 12.

R57: Target position 12 (higher value word)

This register specifies the higher value word of target position 12.

R58: Target position 13 (lower value word)

This register specifies the lower value word of target position 13.

R59: Target position 13 (higher value word)

This register specifies the higher value word of target position 13.

R60: Target position 14 (lower value word)

This register specifies the lower value word of target position 14.

R61: Target position 14 (higher value word)

This register specifies the higher value word of target position 14.

R62: Target position 15 (lower value word)

This register specifies the lower value word of target position 15.

R63: Target position 15 (higher value word)

This register specifies the higher value word of target position 15.

5.3.8 Register Page 3 (User Velocity and Acceleration Table)

Register page 3 specifies the velocities and accelerations for the user positioning table.

R32: Velocity 0 (start value for R39)

This register specifies the velocity 0.

R33: Velocity 1

This register specifies the velocity 1.

R34: Velocity 2

This register specifies the velocity 2.

R35: Velocity 3

This register specifies the velocity 3.

R36: Velocity 4

This register specifies the velocity 4.

R37: Velocity 5

This register specifies the velocity 5.

R38: Velocity 6

This register specifies the velocity 6.

R39: Velocity 7

This register specifies the velocity 7.

R40: Velocity 8

This register specifies the velocity 8.

R41: Velocity 9

This register specifies the velocity 9.

R42: Velocity 10

This register specifies the velocity 10.

R43: Velocity 11

This register specifies the velocity 11.

R44: Velocity 12

This register specifies the velocity 12.

R45: Velocity 13

This register specifies the velocity 13.

R46: Velocity 14

This register specifies the velocity 14.

R47: Velocity 15

This register specifies the velocity 15.

R48: Acceleration 0 (start value for R40)

This register specifies the acceleration 0.

R49: Acceleration 1

This register specifies the acceleration 1.

R50: Acceleration 2

This register specifies the acceleration 2.

R51: Acceleration 3

This register specifies the acceleration 3.

R52: Acceleration 4

This register specifies the acceleration 4.

R53: Acceleration 5

This register specifies the acceleration 5.

R54: Acceleration 6

This register specifies the acceleration 6.

R55: Acceleration 7

This register specifies the acceleration 7.

R56: Acceleration 8

This register specifies the acceleration 8.

R57: Acceleration 9

This register specifies the acceleration 9.

R58: Acceleration 10

This register specifies the acceleration 10.

R59: Acceleration 11

This register specifies the acceleration 11.

R60: Acceleration 12

This register specifies the acceleration 12.

R61: Acceleration 13

This register specifies the acceleration 13.

R62: Acceleration 14

This register specifies the acceleration 14.

R63: Acceleration 15

This register specifies the acceleration 15.

5.3.9 Examples of Register Communication

In the examples, the numbering of the bytes is according to the description without Word-Alignment.

Example 1: Reading the Firmware Issue Status from Register 9 of a Terminal

Output Data

Byte 0: Control Byte	Byte 1: DataOUT1, High Byte	Byte 2: DataOUT1, Low Byte
0x89 (1000 1001_{bin})	0xXX	0xXX

Explanation:

- Bit 0.7 set indicates register communication active.
- Bit 0.6 not set indicates reading the register.
- Bit 0.5 to Bit 0.0 indicates with 00 1001bin the register number 9.

- The output data word (Byte 1 and Byte 2) has no function at the reading access. If you want to change a register, you have to write the desired value into the output data 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 in the Status Byte, as an acknowledgement.
- The terminal returns the Firmware Issue Status 0x3341 in ASCII code, in the input data word (Byte 1 and Byte 2). This has to be interpreted as ASCII code:
  - ASCII code 0x33 stands for the cipher 3
  - ASCII code 0x41 stands for the letter A. Therefore the firmware version is 3A.

Example 2: Writing to an user register

Note: At normal operation all user registers other than register 31are write protected. In order to deactivate write protection, you have to write the password (0x1235) into register 31. Write protection is activated again by writing any value other than 0x1235

Note that some of the settings that can be made in registers only become active after the next power restart (power-off/power-on) of the terminal.

I. Writing the code word (0x1235) to 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 indicates: register communication active.
- Bit 0.6 set indicates: writing to the register.
- Bit 0.5 to Bit 0.0 indicates with 01 1111bin the register number 31.
- The output data word (Byte 1 and Byte 2) contains the code word 0x1235) to deactivate the 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:

- In the Status Byte, the terminal returns a value, that differs only at bit 0.6 from the value of the of the Control Byte.
- The input data word (Byte 1 and Byte 2) has no function after the writing access. Values that might be shown are not valid!

II. Reading Register 31 (verifying 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 indicates register communication active.
- Bit 0.6 not set indicates reading the register.
- Bit 0.5 to Bit 0.0 indicates with 01 1111bin the register number 31.
- The output data word (Byte 1 and Byte 2) has no function at the reading 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 in the Status Byte, as an acknowledgement.
- The terminal returns the current value of the code word register in the input data word (Byte 1 and Byte 2).

III. Writing into Register 32 (changing the content of the feature register)

Output Data

Byte 0: Control Byte	Byte 1: DataOUT1, High Byte	Byte 2: DataOUT1, Low Byte
0xE0 (1110 0000_{bin})	0x00	0x02

Explanation:

- Bit 0.7 set indicates register communication active.
- Bit 0.6 set indicates: writing to the register
- Bit 0.5 to Bit 0.0 indicates with 10 0000_{bin} the register number 32.
- The output data word (Byte 1 and Byte 2) contains the new value for the feature register.

ATTENTION!!

The given value 0x0002 is only an example!
The bits of the feature register change the properties of the terminal und and have different meanings, depending on the terminal type. Please check the description of the feature register of your terminal type (chapter *register description*) about the meanings of the bits in detail, before changing the values!

Input Data (answer of the bus terminal)

Byte 0: Status Byte	Byte 1: DataIN1, High Byte	Byte 2: DataIN1, Low Byte
0xA0 (1010 0000_{bin})	0xXX	0xXX

Explanation:

- In the Status Byte, the terminal returns a value, that differs only at bit 0.6 from the value of the of the Control Byte.
- The input data word (Byte 1 and Byte 2) has no function after the writing access. Values that might be shown are not valid!

IV. Reading Register 32 (verifying the 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 indicates register communication active.
- Bit 0.6 not set indicates reading the register.
- Bit 0.5 to Bit 0.0 indicates with 10 0000bin the register number 32.
- The output data word (Byte 1 and Byte 2) has no function at the reading 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 in the Status Byte, as an acknowledgement.
- The terminal returns the current value of the feature register in the input data word (Byte 1 and Byte 2).

V. Writing to Register 31 (setting the code word back)

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 indicates register communication active.
- Bit 0.6 set indicates: writing to the register.
- Bit 0.5 to Bit 0.0 indicates with 01 1111_{bin} the register number 31.
- The output data word (Byte 1 und Byte 2) contains 0x0000 to activate the write protection again.

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

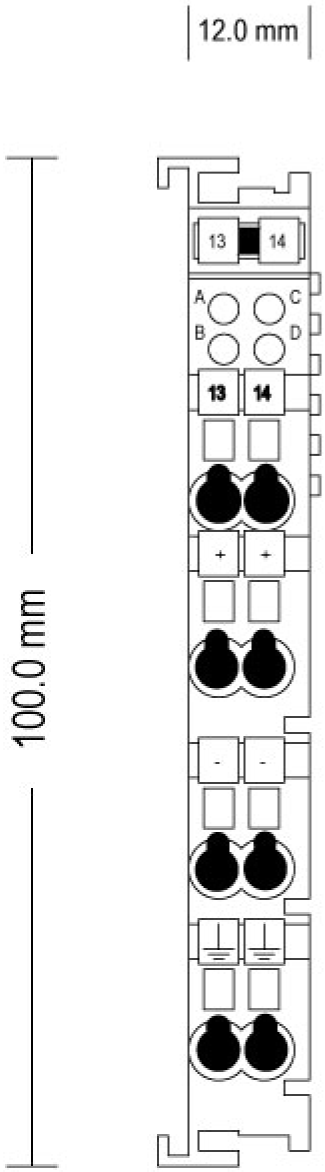
Explanation:

- In the Status Byte, the terminal returns a value, that differs only at bit 0.6 from the value of the of the Control Byte.
- The input data word (Byte 1 and Byte 2) has no function after the writing access. Values that might be shown are not valid!

APPENDIX A

This section provides the mechanical drawing of the I/O terminal.

A.2 I/O 12mm Mechanical Drawing



About Kollmorgen

Kollmorgen is a leading provider of motion systems and components for machine builders. Through world-class knowledge in motion, industry-leading quality and deep expertise in linking and integrating standard and custom products, Kollmorgen delivers breakthrough solutions that are unmatched in performance, reliability and ease-of-use, giving machine builders an irrefutable marketplace advantage.

For assistance with your application needs, contact us at: 540-633-3545, contactus@kollmorgen.com or visit www.kollmorgen.com

North America

Kollmorgen

203A West Rock Road
Radford, VA 24141 USA
Phone: 1-540-633-3545
Fax: 1-540-639-4162

Europe

Kollmorgen

Wacholderstraße 40 – 42
40489 Düsseldorf Germany
Phone: + 49 (0) 203-9979-235
Fax: + 49 (0) 203-9979-3314

KOLLMORGEN®

Because Motion Matters™