



DOCUMENTATION ISG-kernel

Commissioning manual Commissioning of axis

Short Description:
CMS-A1

© Copyright
ISG Industrielle Steuerungstechnik GmbH
STEP, Gropiusplatz 10
D-70563 Stuttgart
All rights reserved
www.isg-stuttgart.de
support@isg-stuttgart.de

Documentation version: 1.06
07/11/2024

Preface

Legal information

This documentation was produced with utmost care. The products and scope of functions described are under continuous development. We reserve the right to revise and amend the documentation at any time and without prior notice.

No claims may be made for products which have already been delivered if such claims are based on the specifications, figures and descriptions contained in this documentation.

Personnel qualifications

This description is solely intended for skilled technicians who were trained in control, automation and drive systems and who are familiar with the applicable standards, the relevant documentation and the machining application.

It is absolutely vital to refer to this documentation, the instructions below and the explanations to carry out installation and commissioning work. Skilled technicians are under the obligation to use the documentation duly published for every installation and commissioning operation.

Skilled technicians must ensure that the application or use of the products described fulfil all safety requirements including all applicable laws, regulations, provisions and standards.

Further information

Links below (DE)

<https://www.isg-stuttgart.de/produkte/softwareprodukte/isg-kernel/dokumente-und-downloads>

or (EN)

<https://www.isg-stuttgart.de/en/products/softwareproducts/isg-kernel/documents-and-downloads>

contains further information on messages generated in the NC kernel, online help, PLC libraries, tools, etc. in addition to the current documentation.

Disclaimer

It is forbidden to make any changes to the software configuration which are not contained in the options described in this documentation.

Trade marks and patents

The name ISG®, ISG kernel®, ISG virtuos®, ISG dirigent® and the associated logos are registered and licensed trade marks of ISG Industrielle Steuerungstechnik GmbH.

The use of other trade marks or logos contained in this documentation by third parties may result in a violation of the rights of the respective trade mark owners.

Copyright

© ISG Industrielle Steuerungstechnik GmbH, Stuttgart, Germany.

No parts of this document may be reproduced, transmitted or exploited in any form without prior consent. Non-compliance may result in liability for damages. All rights reserved with regard to the registration of patents, utility models or industrial designs.

General and safety instructions

Icons used and their meanings

This documentation uses the following icons next to the safety instruction and the associated text. Please read the (safety) instructions carefully and comply with them at all times.

Icons in explanatory text

- Indicates an action.
- ⇒ Indicates an action statement.



DANGER

Acute danger to life!

If you fail to comply with the safety instruction next to this icon, there is immediate danger to human life and health.



CAUTION

Personal injury and damage to machines!

If you fail to comply with the safety instruction next to this icon, it may result in personal injury or damage to machines.



Attention

Restriction or error

This icon describes restrictions or warns of errors.



Notice

Tips and other notes

This icon indicates information to assist in general understanding or to provide additional information.



Example

General example

Example that clarifies the text.



Programming Example

NC programming example

Programming example (complete NC program or program sequence) of the described function or NC command.



Release Note

Specific version information

Optional or restricted function. The availability of this function depends on the configuration and the scope of the version.

Table of contents

Preface	2
General and safety instructions	3
1 Overview	5
2 Axis operation modes	6
2.1 Overview.....	6
2.2 Drive type specific differences.....	8
2.2.1 SERCOS drive with position control in the CNC controller.....	8
2.3 Switching over axes to simulation.....	8
3 Control loop	10
3.1 Introduction.....	10
3.1.1 Drive type specific differences.....	10
3.2 Units.....	11
3.3 Overview of control loop parameters.....	11
3.4 Resolution of the position measuring system.....	12
3.5 Sign for actual and command values.....	13
3.6 Position controller gain.....	14
3.6.1 Arithmetical calculation:.....	16
3.6.1.1 Example: Lightbus drive.....	17
3.6.1.2 Example: Terminal spindle.....	18
3.6.1.3 Example: Terminal spindle with gear.....	19
3.6.1.4 Example: CNC-controlled, translatory axis with SERCOS Drive.....	20
3.6.1.5 Example: CNC-controlled, rotary axis with SERCOS drive.....	21
3.6.2 Experimental determination:.....	22
4 Drive enables/drive status	24
4.1 Drive enables.....	24
4.2 Drive status.....	25
4.3 Drive enables/drive status of drive types.....	26
4.3.1 SERCOS drives.....	26
5 Special features when using stepper motors	27
5.1 General.....	27
5.2 Possible configurations.....	27
5.2.1 Operation without encoder.....	27
5.2.2 Operation with encoder.....	28
5.3 Function restrictions.....	28
6 Appendix	29
6.1 Suggestions, corrections and the latest documentation.....	29

1 Overview

Task

This document describes how to parameterise axes in the CNC controller.

- It describes the axis parameters and PLC signals required for start-up.
- It is assumed that basic machine configuration (number of axes, number of channels) is already complete and the controller starts up without any warnings or error messages.

Properties

This documentation focuses on examples for start-up and parameter definition for axes in SER-COS drives. The chapters contain tables listing other drives, possible operation modes and dependencies on drive types.

Mandatory note on references to other documents

For the sake of clarity, links to other documents and parameters are abbreviated, e.g. [PROG] for the Programming Manual or P-AXIS-00001 for an axis parameter.

For technical reasons, these links only function in the Online Help (HTML5, CHM) but not in pdf files since pdfs do not support cross-linking.

2 Axis operation modes

2.1 Overview

Axis operation mode

CNC axes can be divided into axes with drive-internal or CNC-internal position control. Both operation modes are possible for SERCOS drives. An axis operation mode can be defined in the axis parameter P-AXIS-00320 (operation_mode).



Notice

The axis parameter P-AXIS-00320 (operation_mode) is only used for SERCOS axes.

Variable name	Type	Permitted range	Axis type	Dimension
operation_mode	ISG_CHAR	see strings below	T, R	----

String	Meaning
DRIVE_DEFAULT	Default setting The operation mode used (position control in the drive, position control in the controller, velocity control, etc.) is defined automatically. It depends on the drive type and on the configured telegram.
DRIVE_POSITION_CONTROL	In this operation mode, position and velocity control take place in the drive.
CNC_POSITION_CONTROL	Position control takes place in the controller.

Default setting

If no operation mode is specified for an axis, the controller uses the default setting. The table below shows an overview of the possible operation modes depending on the drive type (default setting marked).

Drive type	Position controller in drive	Position controller in controller
Conventional drive		X (default)
SERCOS	X (default)	X
ProfiDrive MC	X (default)	X
Drive simulation		X (default)
Lightbus		X (default)
Terminal ($\pm 10V$)		X (default)
Real-time Ethernet		X (default)
CANopen	X (default)	X
Virtual		X (default)

Remark for ProfiDrive / CANopen

With ProfiDrive drives, the position lag is also calculated in the controller in "Position control in drive" mode.

With CANopen drives, the operation mode is selected depending on the bus telegram configured. Position control takes place in the drive if command positions are transferred. However, if only a velocity command value is transferred to the drive, position control takes place in the controller.

2.2 Drive type specific differences

2.2.1 SERCOS drive with position control in the CNC controller

Position control with SERCOS

With SERCOS drives, position control generally takes place in the drive. Alternatively, the drive can operate in “velocity control” mode and position control then takes place in the CNC controller.

The following requirements must be met for position control in the CNC controller:

- The “velocity control” mode is enabled in the SERCOS drive.
- The set velocity is transferred to the drive (the telegram contains S-0-0036).
- The actual drive positions are transferred to the controller (the telegram contains S-0-0051, S-0-0053 or S-0-0386).
- CNC position control (CNC_POSITON_CONTROL) is selected for the axis mode (see P-AXIS-00320).

Limitations of position control in the CNC

If position control takes place in the CNC controller, the following limitations exist compared to the “position control in drive” mode:

- No drive-controlled homing is possible. Instead, CNC-controlled homing is carried out.
- Currently, no zero pulse search is possible with CNC-controlled homing for SERCOS drives. Therefore, specify the axis parameter P-AXIS-00084 (homing_without_zero_pulse) as 1.
- No bus delays are considered to calculate the position lag (cf. P-AXIS-00191) .

2.3 Switching over axes to simulation

For test and development purposes, it is sometimes practical to move axes using the "Simulation" drive type. You can use the parameter P-AXIS-00790 to quickly switch between real drive and simulation.

This allows you to switch over axes to simulation without changing the configuration. When you switch back to the real drive type, you do not need to link the axes again.

When the parameter is activated in the default axis parameter list Activated, all axes are set to simulation. Individual axes can be excluded from this operation by deactivating the parameter in the axis parameter list.

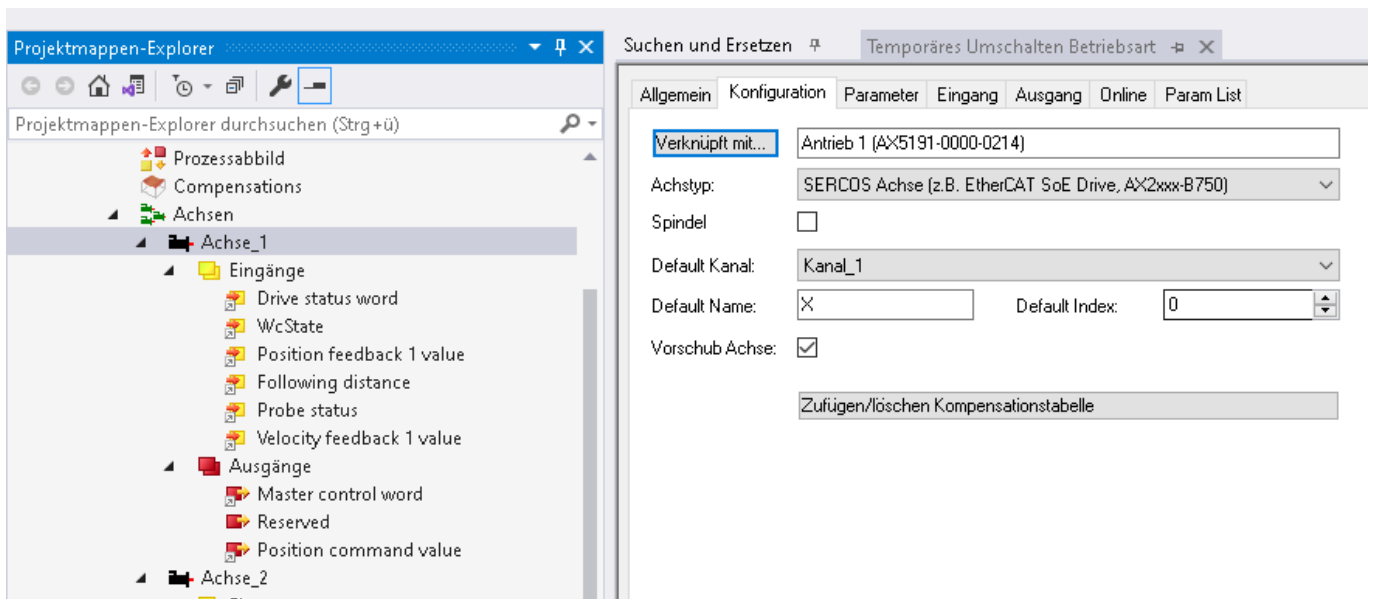


Fig. 1: Axis_1 is linked to SECOS drive

If you do not use P-AXIS-00790, you must switch the axis type to "Simulation" for each individual axis when you switch to simulation. This results in losing all links to process data, see figure below. When switching back to the real drive type, you must re-establish the link to the correct drive for each individual axis

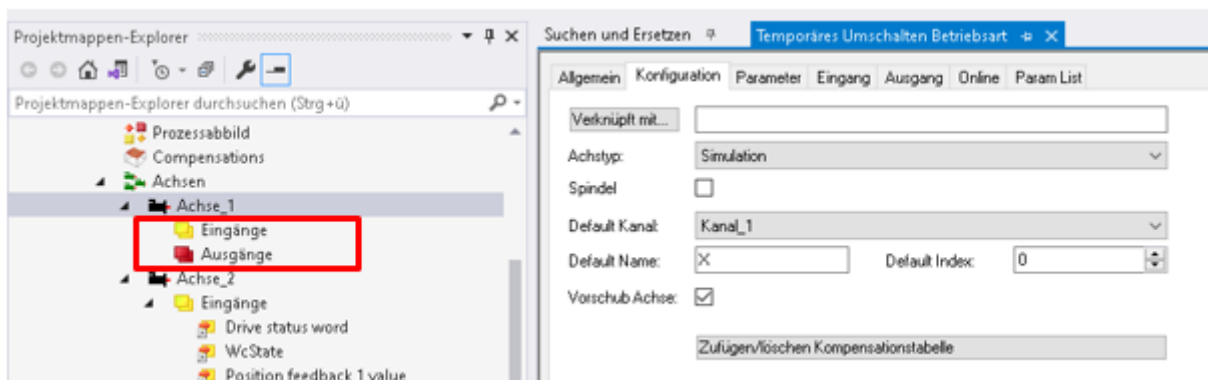


Fig. 2: Axis_1 is set to simulation

The status of the link for the axis is retained when you use P-AXIS-00790. As a result, the above-mentioned work steps are not required. All axes can also be switched to simulation simultaneously using the default axis parameter list.



Notice

This functionality is available as of V2.11.2055 ; V2.11.2827 ; V3.1.3079.39 ; V3.1.3107.28.

3 Control loop

3.1 Introduction

3.1.1 Drive type specific differences

Depending on the drive type, the actual position controller for the axes is implemented in the CNC or in the drive.

With an intelligent drive, the drive assumes position control. The axis parameters for the position control loop are carried out by corresponding drive parameters. In this case, the CNC position control loop is disabled. The CNC assumes monitoring functions only (e.g. position lag monitoring or collision monitoring) and supplies command value cyclically to the drive.

On the other hand, if the CNC position controller is used (e.g. Terminal drive type), a manipulated variable is output cyclically to the drive. This manipulated variable contains the appropriate normalisations based on the set axis parameters.

The table below shows the effect of the above parameters with different drive types:

	Position controller..				
	..in the drive	..in the controller			
Drive type	SERCOS	SERCOS	ProfiDrive MC	Lightbus	Terminal (±10V)
Parameter					
Measuring system resolution wegauf_z, wegauf_n		X	X	X	X
Gain factor of P position control loop Kv		X	X	X	X
Adaptation to D/A multi_gain_z, multi_gain_n		X		X	X
Sign vz_stellgr, vz_istw		X	X	X	X



Attention

Before starting up the axes, check the electrical safety devices of the axes, especially the emergency stop devices and hardware limit switches. They must be operational.

3.2 Units

The following units are generally used in the axis parameter lists:

Size	Unit
Position linear	0.1 μm
Position rotary	$0.1 \times 10^{-3} \text{ }^\circ$
Velocity	1 $\mu\text{m} / \text{s}$ or $10^{-3} \text{ }^\circ/\text{s}$
Acceleration	1 mm/s^2
Time	μs
Position controller gain (Kv)	$0.01 * 1/\text{s}$

3.3 Overview of control loop parameters

The graphic below shows the position control loop used and the contact points of the drive parameters that are explained in the sections below.

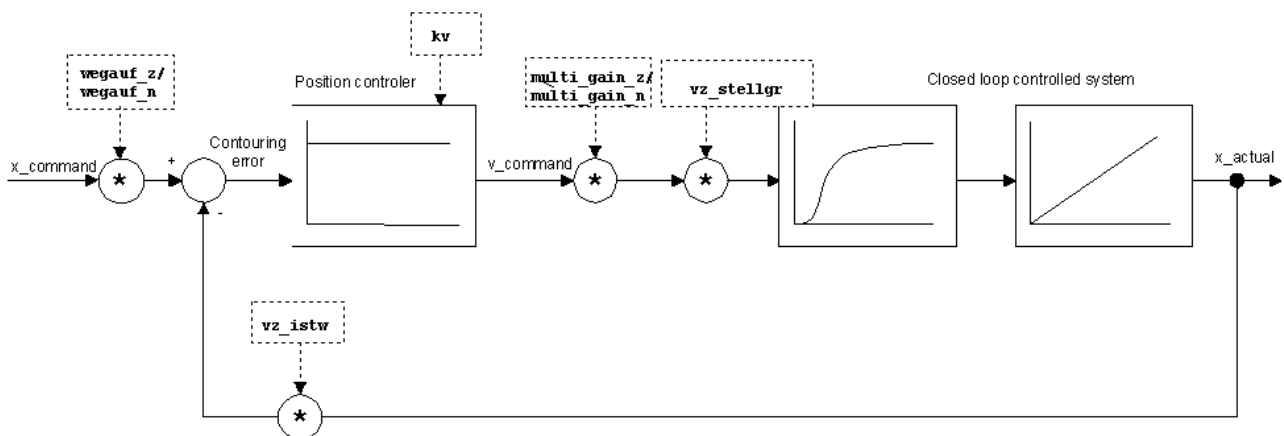


Fig. 3: Overview of the position control loop for conventional drives.

3.4 Resolution of the position measuring system

Configuration of the conversion factor

To convert the increments supplied by the position measuring system to the metric system (the unit used internally), the conversion factor must be entered in the two entries `getriebe[x].wegaufz` and `getriebe[x].wegaufn`. If gears are present, they must also be taken into consideration. The resolution is specified as a fraction and the dimension is increments/0.1 μm and Increments per 0.0001°.



Example

Example 1

For an incremental encoder with 4096 increments and a spindle pitch of 5 mm, the resolution is 4096 increments per 5 mm. Therefore, enter the following data in the axis parameter list:

```
getriebe[0].wegaufz 4096
getriebe[0].wegaufn 50000
```



Example

Example 2

In addition, if a gear is used with a gear ratio of $i = 10:1$, the distance per motor revolution is 5 mm / i . This results in the following for `wegaufz` and `wegaufn`:

```
getriebe[0].wegaufz 4096
getriebe[0].wegaufn 5000
```

Variable name	Type	permitted range	Dimension
wegaufz P-AXIS-00234	SGN32	MAX(SGN32) \rightarrow wegaufz \rightarrow 1	Incr.

Variable name	Type	permitted range	Dimension
wegaufn P-AXIS-00233	SGN32	MAX(SGN32) \rightarrow wegaufn \rightarrow 1	0.1 μm or 0.0001°



Notice

Each of these entries is not directly adopted when the axis parameter list is updated. A controller restart is required.



Notice

For SERCOS drives with position control in the controller (CNC_POSITION_CONTROL mode), the path resolution of the measuring system is taken into consideration in the feedrate constant S-0-0123, i.e. the value 1 results for the quotient P-AXIS-00234/ P-AXIS-00233.

3.5 Sign for actual and command values

Inversion

To adapt the manipulated variable and the position sensor signals to the hardware used, it is sometimes necessary to invert the signs for the manipulated variable and the position actual value.



Example

The axis is programmed in the positive direction. However, the mechanical system actually moves in the negative direction.

In this case, use the two entries `lr_hw[].vz_stellgr` and `lr_hw[].vz_istw`.

The manipulated variable output is inverted by setting `vz_stellgr` to 1; when `vz_istw` is set to 1, an increase in the position of the position sensor results in a reduction of the axis position used in the CNC kernel.

```
lr_hw[0].vz_stellgr  0
```

```
lr_hw[0].vz_istw    0
```

Variable name	Type	Permitted range	Dimension
<code>vz_stellgr</code>	BOOLEAN	0, 1	----

Variable name	Type	Permitted range	Dimension
<code>vz_istw</code>	BOOLEAN	0, 1	----



Notice

Each of these entries is not directly adopted when the axis parameter list is updated. A controller restart is required.

3.6 Position controller gain

Gain

The CNC position controller uses a P controller for position control. The controller gain is defined by the parameter `getriebe[x].kv` of dimension 0.01/s.

In a stationary case, the following applies when a control loop is set correctly:

$$s = \frac{v}{kv}$$

Where

s: position lag

v: axis velocity

kv: position controller gain

To set a position controller gain of 3/s, the following entry is required in the axis parameter list:

`getriebe[].kv 300`

Variable name	Type	permitted range	Dimension
kv	UNS32 at the MC_ReadAxisError function block at its AxisErrorID output	$0 < kv \rightarrow \text{MAX}(\text{UNS32})$	0.01/s

Adapt the manipulated variable to D/A converter format

The two entries P-AXIS-00129 (`getriebe[x].multi_gain_z`) and P-AXIS-00128 (`getriebe[x].multi_gain_n`) are used to adapt the manipulated variable calculated from the position lag to the D/A converter format.

The fraction P-AXIS-00129 / P-AXIS-00128 results in the conversion factor of the manipulated variable into the D/A converter format and is also referred to as the multi-gain factor.

There are 2 possibilities to calculate P-AXIS-00129 and P-AXIS-00128:

1. Arithmetical calculation [► 16]:

Specify the digital value at the D/A converter input at which the axis moves at a velocity of [1m/min] or [1000 °/min]. To calculate `multi_gain_z/multi_gain_n` arithmetically, gear ratios, spindle pitch and the scaling factors acting on the velocity command value on the drive side must be known.

2. Experimental determination [► 22]:

Derived from position lag and axis velocity.



Attention

The entries P-AXIS-00129 and P-AXIS-00128 are provided solely to adapt the format of the manipulated variable. To optimise the control loop, use the position controller gain factor P-AXIS-00099 (getriebe[].kv).

3.6.1 Arithmetical calculation:

The multi-gain factor is defined as the value of the manipulated variable at which the axis moves at a velocity of 1 m/min or 1000 °/min.

This section provides examples for calculating the multi-gain factor arithmetically for a Lightbus drive and for a Terminal drive. The gain factors of all transmission elements between the controller velocity command value output through to the moved axis are required to calculate the multi-gain factor arithmetically. They include:

- Scaling factors acting on the velocity command value
- Gear ratios
- Spindle pitches etc.

The procedure is always identical:

1. Calculate the conversion factor f between motor revolutions per minute and the command value transferred in the cyclic telegram.
2. Calculate the motor speed n_{1m}/min required to move the axis at a velocity of 1 m/min or 1000 °/min.
3. The ratio between P-AXIS-00129 and P-AXIS-00128 is then equal to the product $f * n_{1m}/\text{min}$.



Notice

For a SERCOS drive with position control in the controller, the scaling factor (P-AXIS-00129/ P-AXIS-00128) is the velocity command value S-0-0036 that should be output to the drive so that the axis moves at a velocity of 1 m/min (translatory axis) or 1000 °/min (rotary axis).

The scaling factor is dependent on:

- the feed rate constant S-0-0123
- the weighting factor S-0-0045 and the weighting exponent S-0-0046 of the velocity data in the drive
- the gear ratios S-0-0121 and S-0-0122

3.6.1.1 Example: Lightbus drive

With a Lightbus drive, the parameter VUNIT is assigned the value 1 in the default factory parameter definition. This means that the specified velocity for the drive takes place in revolutions per minute (rpm).

Step 1 of 3:

According to the manufacturer's documentation, the conversion factor between revolutions per minute and the command value transferred in the cyclic telegram is:

$$f = \frac{1048576 * 32}{4000 * 60} \approx 139,8101 \frac{DA_Digits}{U/min}$$

The value 140 (rounded) must be output to rotate the motor at a speed of one revolution per minute.

Step 2 of 3:

To reach a speed of 1 m/min, the following motor speed is required at a spindle pitch of 5 mm and a gear ratio of $i = 1:1$

$$n_{1m/min} = \frac{1 \text{ m/min}}{s} = \frac{1000 \text{ mm/min}}{5 \text{ mm/rev}} = 200 \text{ rev/min}$$

Step 3 of 3:

At a velocity scaling factor f , the output velocity command value for 1 m/min is:

$$\begin{aligned} DA_{out1m/min} &= f * n_{1m/min} = 139,8101 \frac{DA_Digits}{rev/min} * 200 \text{ rev/min} \\ &= 27962,02 \text{ DA_Digits} \end{aligned}$$

This results in the following value:

$$multi_gain = \frac{P - AXIS - 00129}{P - AXIS - 00128} = 27962,02$$

Alternatively, $multi_gain$ can be calculated using the following equation:

$$multi_gain = \frac{P - AXIS - 00129}{P - AXIS - 00128} = \frac{10^7}{I_{rev}} * \frac{P - AXIS - 00034}{P - AXIS - 00033} * f$$

where I_u is the number of encoder increments per motor revolution. For a Lightbus drive, this results in 65536 increments per revolution.

A spindle pitch of 5 mm/revolution and 65536 increments per revolution result in $wegauf_z=65536$ and $wegauf_n=50000$.

$$multi_gain = \frac{P - AXIS - 00129}{P - AXIS - 00128} = \frac{10^7}{65536} * \frac{65536}{50000} * \frac{1048576 * 32}{4000 * 60}$$

The possible values for P-AXIS-00129 and P-AXIS-00128 would then be:

P-AXIS-00129	27962	279620	2796202	...
P-AXIS-00128	1	10	100	...

3.6.1.2 Example: Terminal spindle

A spindle is activated by a D/A converter terminal KL4032; a voltage of 10 V at the input of the drive controller corresponds to a speed of 9000 rpm; no gear is present. With the D/A converter terminal KL4032, a digital output value of 32767 (0x7FFF) is equivalent to an output voltage of 10 V with default scaling.

Step 1 of 3:

The conversion factor between rpm and the command value transferred in the cyclic telegram is then:

$$f = \frac{32767 * \frac{DA_Digits}{10V}}{9000 * \frac{rev}{min}} \approx 3,6408 \frac{DA_Digits}{rev/min}$$

A value of 3.64 would therefore be output to the D/A converter in order to rotate the spindle at a speed of 1 rpm.

Step 2 of 3:

To reach a speed of 1000 °/min, the following motor speed is required:

$$n_{1m}/min = \frac{1000 \text{ }^\circ/min}{360 \text{ }^\circ/rev} = 2,7778 \text{ rev}/min$$

Step 3 of 3:

At a velocity scaling factor f, the output velocity command value for 1000 °/min is:

$$DA_{out \text{ } 1m/min} = f * n_{1m}/min = 3,6408 \frac{DA_Digits}{rev/min} * 2,7778 \frac{rev}{min} = 10,1134 DA_{Digits}$$

The value of multi_gain is therefore:

$$multi_gain = \frac{P - AXIS - 00129}{P - AXIS - 00128} = 10,1134$$

The possible values for P-AXIS-00129 and P-AXIS-00128 would then be:

P-AXIS-00129	10	101	1011	...
P-AXIS-00128	1	10	100	...

3.6.1.3 Example: Terminal spindle with gear

A spindle is activated by a D/A converter terminal KL4032; a voltage of 10 V at the input of the drive controller corresponds to a speed of 9000 rpm; the motor is fitted with a gear which has a gear ratio of $i=5$ (5 motor revolutions correspond to one revolution at the gear output). With the D/A converter terminal KL4032, a digital output value of 32767 (0x7FFF) is equivalent to an output voltage of 10 V with default scaling.

Step 1 of 3:

As in the previous example, the conversion factor between motor revolutions per minute and the command value transferred in the cyclic telegram is:

$$f = \frac{32767 * \frac{DA_Digits}{10V}}{9000 * \frac{rev}{min}} \approx 3,6408 \frac{DA_Digits}{\frac{rev}{min}}$$

A value of 3.6408 would therefore be output to the D/A converter in order to rotate the spindle motor at a speed of 1 rpm.

Step 2 of 3:

Now the use of a gear becomes noticeable.

To reach a speed of 1000 °/min, the following motor speed is required:

$$n_{1m/min} = \frac{1000 \text{ °/min}}{360 \text{ °/rev}} * i = 13,8889 \text{ rev/min}$$

Step 3 of 3:

At a velocity scaling factor f , the output velocity command value for 1000 °/min is:

$$\begin{aligned} DA_out_{1m/min} &= f * n_{1m/min} = 3,6408 \frac{DA_Digits}{\frac{rev}{min}} * 13,8889 \frac{rev}{min} \\ &= 50,5668 DA_Digits \end{aligned}$$

The value of `multi_gain` is therefore:

$$multi_gain = \frac{P - AXIS - 00129}{P - AXIS - 00128} = 50,5668$$

The possible values for P-AXIS-00129 and P-AXIS-00128 would then be:

P-AXIS-00129	50	506	5057	...
P-AXIS-00128	1	10	100	...

3.6.1.4 Example: CNC-controlled, translatory axis with SERCOS Drive

The linear axis moves by 1 mm per motor revolution. The position data in the drive are weighted for translatory movements:

Weighting factor: S-0-0077 = 1

Weighting exponent: S-0-0078 = -7

The path resolution of the measuring system is therefore:

$$10^{-7} m = 0,1 \mu m$$

A path motion of 1 mm per motor revolution results in a feedrate constant of:

$$S-0-0123 = \frac{1 \text{ mm}}{0,1 \mu m} = 10000$$

A translatory weighting is also used for velocity data:

Weighting factor: S-0-045 = 1

Weighting exponent: S-0-046 = -4

The weighting factor for velocity is therefore:

$$10^{-4} \frac{m}{min}$$

Manipulated variable scaling is therefore:

$$\frac{P - \text{AXIS} - 00129}{P - \text{AXIS} - 00128} = \frac{\text{multi_gain_z}}{\text{multi_gain_n}} = \frac{1 \frac{m}{min}}{10^{-4} \frac{m}{min}} = \frac{10000}{1}$$

Excerpt from the axis parameter list

```

getriebe[0].wegaufz      1
getriebe[0].wegaufn     1
getriebe[0].multi_gain_z 10000
getriebe[0].multi_gain_n 1
  
```

3.6.1.5 Example: CNC-controlled, rotary axis with SERCOS drive

The encoder supplies 65536 increments per revolution (S-0-0116). The rotation resolution in the drive is therefore specified by S-0-0079 = 65536.

The resolution of the position measuring system is therefore:

$$\begin{aligned} \frac{P - \text{AXIS} - 00234}{P - \text{AXIS} - 00233} &= \frac{\text{wegauf}_z}{\text{wegauf}_n} = \frac{65536 \text{ Increments}}{360 * 10^{-4} \text{ degree}} \\ &= \frac{65536 \text{ Increments}}{3600000 * 0,0001^\circ} \end{aligned}$$

The drive uses a rotary weighting for speeds:

Weighting factor: S-0-045 = 1

Weighting exponent: S-0-046 = -4

The weighting factor for the speeds is:

$$10^{-4} \frac{\text{rev}}{\text{min}} = 10^{-4} \frac{360^\circ}{\text{min}}$$

The scaling required for the manipulated variable is therefore:

$$\frac{P - \text{AXIS} - 00129}{P - \text{AXIS} - 00128} = \frac{\text{multi_gain}_z}{\text{multi_gain}_n} = \frac{1000 \frac{^\circ}{\text{min}}}{10^{-4} \frac{360^\circ}{\text{min}}} = \frac{10000000}{360}$$

Excerpt from the axis parameter list

```
getriebe[0].wegaufz      65536
getriebe[0].wegaufn     3600000
getriebe[0].multi_gain_z 10000000
getriebe[0].multi_gain_n 360
```

3.6.2 Experimental determination:

Alternatively, the ratio `multi_gain_z/multi_gain_n` can be determined experimentally. The actual gain acting in the position control loop can be determined from the position lag and axis velocity when the axis is moving at constant speed according to the equation:

$$k = \frac{v}{s}$$

The gain determined experimentally is compared with the `kv` factor (P-AXIS-00099) set in the axis parameter list:

$$f = \frac{P - AXIS - 00099 / 100}{k}$$

Then the new multi-gain factor is calculated starting from the currently parameterised values of `multi_gain_z` and `multi_gain_n`:

$$multi_gain_{new} = \frac{P - AXIS - 00129_{new}}{P - AXIS - 00128_{new}} = \frac{P - AXIS - 00129_{cur}}{P - AXIS - 00128_{cur}} * f$$

After completing a parameter change, the parameters must be checked again using equation 1. To further optimise the position control loop, the parameter P-AXIS-00099 (`getriebe[.kv]`) is used exclusively.



Attention

If factor `f` is greater than 1, the total gain in the position control loop increases when the `multi_gain` factor is changed. This may cause the axes to oscillate. In such cases, set the position controller gain P-AXIS-00099 to the value `P-AXIS-00099 / f` in order to maintain total gain in the position control loop constant.



Example

Position lag and gain in the position control loop:

position lag $s = 1,9$ mm, axis velocity $v = 200$ mm/min, P-AXIS-00099 = $400 * 1/s$, multi_gain_z = multi_gain_n = 1.

Actual gain in the position control loop:

$$k = v/s = 200 \text{ mm} / (60 \text{ s} * 1.9 \text{ mm})$$

$$k = 1.75$$

The ratio between parameterised gain and actual effective gain results from equation 1:

$$f = \frac{P - AXIS - 00099 / 100}{k} \approx 2,286$$

The multi-gain factor is therefore.

$$multi_gain_{new} = \frac{P - AXIS - 00129_{new}}{P - AXIS - 00128_{new}} = \frac{1}{1} * 2,286 = 2,286$$

Possible entries in the axis parameter list would then be:

P-AXIS-00129: getriebe[].multi_gain_z 2286

P-AXIS-00128: getriebe[].multi_gain_n 1000

Since these entries increase total gain in the position control loop by the factor f , P-AXIS-00099 should also be multiplied by the factor $1/f$ to maintain total gain in the control loop constant:

$$P - AXIS - 00099_{new} = \frac{P - AXIS - 00099_{old}}{f} = \frac{400}{2,286} \approx 175$$

After checking the parameters calculated based on equation 1, the position control loop can be further adjusted by changing P-AXIS-00099.

4 Drive enables/drive status

4.1 Drive enables

To move a drive, the PLC must set enable bits on the HLI. The PLC can set the required enables by activating the associated control units. For more information on the HLI (High Level Interface) and how to handle control units, please refer to the [HLI] documentation.

The HLI enable bits are normally transferred to the drive amplifier together with the cyclic drive data and act directly in the drive. In addition, plausibility checks are carried out in the NC kernel using the drive enables.

For example, if the required enables are not set, an error message is output if an attempt is made to move an axis with the NC.

Enables act on individual axes. For more information, please refer to the HLI documentation in the section Status information of an axis drive.

The control units for drive enables are:

- **MCControlBoolUnit_DriveOn:** Drive on (abbr.: DriveOn).
- **MCControlBoolUnit_TorquePermission:** Activate the drive-internal position or speed controller. The drive generates a torque and goes to control (abbr.: TorquePermission).
- **MCControlBoolUnit_ReleaseFeedhold:** Reset drive-internal feedhold (abbr. ReleaseFeedhold).

Since not all drive types in the above list support the functions described, the table below provides an overview of the enables required for each of the drive types. Each of the drive types is then described in detail in the following sections.

An X in the table indicates that the drive supports the function and the related enable must then be set on the HLI.

Control unit	SERCOS drive	PROFI-DRIVE drive	Terminal drive	Lightbus drive	RT Ethernet drive	CANopen drive
DriveOn	X	X				X
TorquePermission	X	X	X	X	X	X
Release Feedhold	X					X

4.2 Drive status

Digital drives supply information on the internal drive status in the cyclic telegram. This information is evaluated internally in the NC kernel and sent to the HLI for the PLC.

For CNC versions up to Version V2.11.20xx

The state signals can be read out in section

`pAC[achs_index]^ .addr^ .StateLR_Data`

(achs_index starts at 1).

The following signals are applied there:

- **X_ReadyPowerOn**: The drive is ready for power connection
- **X_ReadyControlLoopOn**: The drive goes to control. Once the corresponding drive enables [▶ 24] are set, the drive obeys the positions specified by the NC kernel.

For CNC versions as of V2.11.2800

The state signals can be read out in the section

`gpAx[achs_index].lr_state.StateLR_Data`

(achs_index starts at 0).

The following signals are applied there:

- **ready_for_power_on_r**: The drive is ready for power connection
- **ready_for_control_loop_on_r**: The drive goes to control. Once the corresponding drive enables [▶ 24] are set, the drive obeys the positions specified by the NC kernel.

Depending on the drive type, not all signals are supported on the HLI. The table below contains an overview of the state signals supported by each drive type:

State signal	SERCOS drive	PROFIDRIVE drive	Terminal drive	Lightbus drive	RT Ethernet drive	CANopen drive
ReadyPowerOn	X	X		X	X	X
ReadyControl-LoopOn	X	X				X

4.3 Drive enables/drive status of drive types

4.3.1 SERCOS drives

The following enable and state signals are supported on SERCOS drives.

- **MCControlBoolUnit_DriveOn:**
Enable power connection in the drive (abbr.: DriveOn). This enable signal is entered directly in Bit 15 of the SERCOS control word.
- **MCControlBoolUnit_TorquePermission:**
Activate drive-internal position or speed controller. The drive goes to control (abbr.: TorquePermission).
- **MCControlBoolUnit_ReleaseFeedhold:**
Reset drive-internal feedhold (abbr: ReleaseFeedhold).
- **ready_for_power_on_r** (for CNC versions as of V2.11.2800)
or **X_ReadyPowerOn** (for CNC versions up to V2.11.20xx):
The drive is ready for power connection
- **ready_for_control_loop_on_r** (for CNC versions as of V2.11.2800)
or **X_ReadyControlLoopOn** (for CNC versions up to V2.11.20xx):
The drive is ready to go to control. Once the corresponding drive enables [▶ 24] are set, the drive obeys the positions specified by the NC kernel.

5 Special features when using stepper motors

5.1 General

Stepper motors can be connected to the CNC by the terminals

- single-channel pulse train output terminals
- single-channel stepper motor terminals with/without incremental encoder interface

. Velocity command values output by the CNC through these terminals are converted to a signal format suitable to control stepper motors.

The single-channel pulse train output terminal generates an output signal suitable to control a stepper motor power output stage (e.g. a pulse/direction signal). The single-channel stepper motor terminals already contain power output stages for the direct control of 2-phase stepper motors.

5.2 Possible configurations

All the terminals listed above supply the number of currently output steps in the process map. The stepper motor terminal with incremental encoder interface also offers the possibility of connecting an incremental encoder directly to the terminal.

5.2.1 Operation without encoder

When operating without encoder, the impulses generated by the terminals are sent to the CNC-internal position controller as actual position. The velocity command values calculated by the position controller are output to the stepper motor terminal and cause the stepper motor to move.

Since the drive is operated in a pseudo position control loop, the stepper motor is parameterised in the axis parameter list similarly to the parameter definition of a drive with +/-10 V interface. This also takes place when velocity feedforward control is used.



Attention

In this operation mode, there is no real axis position feedback. Axis position lag monitoring is not possible.

Since a stepper motor stands still in the event of dynamic overload or it can lose steps (stepper motor falls out of step), special care is required when dimensioning the motor and parameterising the dynamic parameters of the motor.

5.2.2 Operation with encoder

When operating with encoder, the position actual value of an encoder is supplied to the position controller as a real position actual value of the motor. This takes place when the drive is configured by linking the encoder input of the axis to an encoder terminal and not by the position feedback of the stepper motor terminal.

An incremental encoder can be directly connected to the stepper motor terminal with incremental encoder interface. In this case, parameterise the terminal so that it displays the encoder signal in its output process map.

Even when operated with encoder, a stepper motor can stand still if subjected to dynamic overload or lose steps (stepper motor falls out of step). As a result, special care is required when dimensioning the motor and parameterising the dynamic parameters of the motor



Notice

However, as opposed to operating without encoder, it is possible to use position lag monitoring.

5.3 Function restrictions

Since the terminal KL2541 may not be provided with a position value latched in the terminal in the cyclic process map, the following functions are not available:

- Referencing to zero pulse
- Measuring

If they are activated anyway, an error message is output and the value is corrected automatically.

6 Appendix

6.1 Suggestions, corrections and the latest documentation

Did you find any errors? Do you have any suggestions or constructive criticism? Then please contact us at documentation@isg-stuttgart.de. The latest documentation is posted in our Online Help (DE/EN):



QR code link: <https://www.isg-stuttgart.de/documentation-kernel/>

The link above forwards you to:

<https://www.isg-stuttgart.de/fileadmin/kernel/kernel-html/index.html>



Notice

Change options for favourite links in your browser;

Technical changes to the website layout concerning folder paths or a change in the HTML framework and therefore the link structure cannot be excluded.

We recommend you to save the above "QR code link" as your primary favourite link.

PDFs for download:

DE:

<https://www.isg-stuttgart.de/produkte/softwareprodukte/isg-kernel/dokumente-und-downloads>

EN:

<https://www.isg-stuttgart.de/en/products/softwareproducts/isg-kernel/documents-and-downloads>

E-Mail: documentation@isg-stuttgart.de



© Copyright
ISG Industrielle Steuerungstechnik GmbH
STEP, Gropiusplatz 10
D-70563 Stuttgart
All rights reserved
www.isg-stuttgart.de
support@isg-stuttgart.de

