

Lenze

Manual



Prepared Solutions

Cross Cutter

9300 Servo PLC / ECSxA

Important note:

This software is supplied to the user as described in this document. All risks resulting from its quality or use remain the responsibility of the user. The user must provide all safety measures protecting against possible maloperations.

We do not take any liability for direct or indirect damage, e.g. profit loss, order loss or any loss regarding business.

© 2006 Lenze Drive Systems GmbH

No part of this documentation may be reproduced or made available to third parties without written consent from Lenze Drive Systems GmbH.

All indications given in this documentation have been carefully selected and comply with the hardware and software described. Nevertheless, deviations cannot be ruled out. We do not take any responsibility or liability for damages which might possibly occur. Required corrections will be made in the following editions.

All product names mentioned in this documentation are trademarks of their corresponding owners.



1	Preface and general information.....	1-1
1.1	Version info.....	1-1
1.2	Scope of supply.....	1-2
1.3	General information about the prepared solution.....	1-4
1.4	Functional range and variants.....	1-5
1.4.1	Parameterisable variant.....	1-6
1.4.2	Programmable variant.....	1-8
1.5	Conventions used.....	1-10
2	Features of the "Cross Cutter" prepared solution.....	2-1
2.1	General features of the "Cross Cutter" prepared solution.....	2-1
2.2	Structure and functional principle.....	2-2
2.2.1	Typical structure of a cross-cutter application.....	2-2
2.2.2	Typical motion profiles.....	2-3
2.2.3	Example applications of the "Cross Cutter" prepared solution.....	2-9
2.3	Functional application criteria for the parameterisable variant.....	2-10
2.4	Machine parameters.....	2-15
2.5	Basic features.....	2-17
2.5.1	Reposition unlocked drive to motion profile.....	2-17
2.5.2	Cross-cutter operating modes.....	2-20
2.5.3	ManualJog.....	2-42
2.5.4	Following error monitoring.....	2-45
2.5.5	Homing.....	2-46
2.5.6	Simple positioning.....	2-57
2.5.7	Length-controlled or mark-controlled operation.....	2-61
2.5.8	Adjustable compensation speed for mark correction.....	2-63
2.5.9	Mark window.....	2-67
2.5.10	Manual X trimming.....	2-72
2.5.11	Reject gate control.....	2-78
2.5.12	Calculating the print-mark-sensor distance.....	2-81
2.5.13	Print-mark register.....	2-85
2.5.14	Torque precontrol.....	2-87
2.5.15	Evaluation of absolute value encoders at load and drive end.....	2-91
2.5.16	Evaluation of the master value signal.....	2-95
3	Installing and starting up the "CrossCutter" prepared solution.....	3-97
3.1	Required components.....	3-97
3.2	Installing the Cross Cutter prepared solution.....	3-98
3.3	Hardware structure.....	3-98
3.3.1	Wiring the control terminals.....	3-99
3.4	Setting the parameters of the prepared solution.....	3-101
3.4.1	Control/Status interface with master control.....	3-101
3.4.2	Preparing for installation and start-up: Generating profile data.....	3-107
3.4.3	Downloading the CrossCutter_V0005 program.....	3-117
3.4.4	Online start-up with GDC.....	3-123
3.4.5	Starting up length-controlled operation.....	3-135
3.4.6	Starting up mark-controlled operation.....	3-142
4	Functions.....	4-150
4.1	State machine.....	4-150
4.2	Conditions for state change.....	4-151
4.3	"StandBy" state.....	4-152
4.4	"ManualJog" state.....	4-152
4.5	"Homing" state.....	4-152
4.6	"Positioning" state.....	4-152
4.7	"Cam" state.....	4-152
4.8	"Trouble" state.....	4-152
5	Program expansions.....	5-1
5.1	Task management.....	5-1
5.2	UserErrors PRG.....	5-2



5.3	CrossCutterInterfaces PRG	5-2
6	Appendix	6-1
6.1	Global variables	6-1
6.1.1	VarCrossCutterBasics	6-1
6.1.2	VarCrossCutterMotionProfileCalculation	6-2
6.1.3	VarCrossCutterMultiplexer	6-14
6.1.4	VarCrossCutterPieceCounter	6-15
6.1.5	VarCrossCutterProfileHandling	6-16
6.1.6	VarCrossCutterRejectGate	6-18
6.1.7	VarCrossCutterTPHandling	6-19
6.1.8	VarCrossCutterTrouble	6-21
6.1.9	VarCrossCutterXAxisTrimming	6-22
6.1.10	VarBasicOperation	6-23
6.1.11	VarCamControl	6-25
6.1.12	VarHoming	6-26
6.1.13	VarManualJog	6-28
6.1.14	VarPositioning	6-29
6.1.15	VarStateMachine	6-31
6.1.16	VarTorquePrecontrol	6-32
6.1.17	VarTrouble	6-33
6.1.18	VarVertShaftExtPosition	6-34
6.1.19	VarVertShaftExtVelocity	6-34
6.1.20	VarVirtualMaster	6-36
6.1.21	VarXTouchProbeSynchronisation	6-37
6.1.22	VarYStretchCompress	6-38
6.1.23	VarYTouchProbeSynchronisation	6-38
6.2	Prepared solution codes	6-39
6.2.1	Table of application codes	6-39
6.2.2	Code initialisation values	6-57
6.3	Error messages	6-59
6.3.1	System error messages	6-59
6.3.2	Application error messages	6-65
6.3.3	User-defined error messages	6-68
6.3.4	ENUM list for error numbers	6-69
6.4	Signal flow diagram for core function	6-71
6.5	Interface signals	6-72
6.6	Description of the function blocks	6-79
6.6.1	MultiplexerInput function block	6-79
6.6.2	MultiplexerOutput function block	6-82
6.6.3	TP_Window – Masking out mark pulses	6-85
6.6.4	XPositionTrimming function block	6-90
6.6.5	RejectGateHandling function block	6-93
6.6.6	CalcTpDistance function block	6-95
6.6.7	CalcOffsetForShiftedTP function block	6-97
6.6.8	CalcStartPositionOfKnifeDrum function block	6-99
6.6.9	CrossCutterMotionProfileCalculation function block	6-100
6.6.10	TP_Register – Position tracking of mark pulses	6-104
6.6.11	VersionHandling function block	6-108



1 Preface and general information

1.1 Version info

Information about the prepared solution version: This documentation is valid for the Cross Cutter prepared solution, Version V1.x.

Version	ID number	Modifications
1.0 10/2006	ESPLV02XA0FC1	New document
1.1 03/2008	ESPLV02XA0FC1	Reworked document with target ECSxA

Code C3999/000 displays the prepared solution version in the following format:

Code	Possible settings:		Comment
	Default	Selection	
C3999	-	- {1} -	Display code: Version release of the prepared solution (subcode 1: project file, subcode 2: application library, subcode 3: basic library): ▪ The first and second numeral read from right-to-left define the service pack. ▪ The third and forth numeral read from right-to-left define the sub version. ▪ The fifth and sixth numeral read from right-to-left define the main version.
1			
2			
3			



1.2 Scope of supply

The CD-ROM supplied with the product contains the following files:

File type	Use
*.BIN	<u>Binary file:</u>
CrossCutter_SPLC_Vxxxxxx.bin	The binary file contains the ready-compiled project with all system and application codes (for application codes see Chapter 6.2.1) and can be uploaded to the target system using Lenze's GDLoader software tool. You will need the binary file if you are going to be working with the parameterisable variant (see Chapter 1.4.1).
*.LPC	<u>Project file:</u>
CrossCutter_SPLC_Vxxxxxx.lpc	The project file contains the source code for the prepared solution. You will need the project file if you are going to be using the programmable variant of the prepared solution (see Chapter 1.4.2). The file is a read only file to make sure that you always can come back to the original version. If you want to edit this file you first have save it under a different name. Then it can be edited, compiled and uploaded to the target system with Lenze's DDS software.
*.LIB	<u>Library files:</u>
CrossCutterVxxxxxx.lib InterfaceMultiplexerVxxxxxx.lib	The library files contain the core functionality of the prepared solution and provide the basis for the project file. You will not be able to compile the project file in DDS without errors if you do not have the library files.
*.PDB	<u>Device description file German:</u>
CrossCutter_SPLC_Vxxxxxx _S.pdb	You will need the device description file to set the parameters of the prepared solution (parameterisable variant) using Lenze's GDC software tool.
CrossCutter_SPLC_Vxxxxxx _1.pdb	<u>Device description file English:</u> You will need the device description file to set the parameters of the prepared solution (parameterisable variant) using Lenze's GDC software tool.
*.SDB	<u>Symbol file for the GDOscilloscope:</u>
CrossCutter_SPLC_Vxxxxxx.sdb	The symbol file for use in conjunction with Lenze's Global Drive Oscilloscope software contains a list of oscillograph-compatible signals associated with the prepared solution. When GDOscilloscope starts up, the file will need to be assigned in order to be able to access the required signals from the prepared solution.



*.CMM	<u>Project sample files for CamManager/CamDesigner Professional:</u> A sample project for CamDesigner Professional has been included to illustrate a sample application. The project comprises a CMM file and the following associated CAM files: ▪ Product1_Prof.cam ▪ Product2_Prof.cam ▪ Product3_Prof.cam ▪ Product4_Prof.cam ▪ Product5_Prof.cam ▪ Product6_Prof.cam ▪ Product7_Prof.cam ▪ Product8_Prof.cam ▪ Product9_Prof.cam ▪ Product10_Prof.cam ▪ The compiled sample cam file for the cross-cutter axis is called: CrossCutterProf.lc9
*.Cam	<u>Project sample files for CamDesigner Basic:</u> A sample project for CamDesigner Basic has been included to illustrate a sample application. The project comprises a CMM file and the following associated CAM files: ▪ Product1_Basic.cam ▪ Product2_Basic.cam ▪ Product3_Basic.cam ▪ Product4_Basic.cam ▪ Product5_Basic.cam ▪ Product6_Basic.cam ▪ Product7_Basic.cam ▪ Product8_Basic.cam ▪ Product9_Basic.cam ▪ Product10_Basic.cam ▪ The compiled sample cam file for the cross-cutter axis is called: CrossCutterBasic.lc9
*.PDF	<u>PDF file (manual) German:</u> The PDF file contains the manual to accompany the prepared solution (documentation) in German.
CrossCutter_vx-x_DE.pdf	
CrossCutter_vx-x_EN.pdf	<u>PDF file (manual) English:</u> The PDF file contains the manual to accompany the prepared solution (documentation) in English.

The files are installed using an installation wizard.

Users will need to generate additional project files for the "Cross Cutter" prepared solution (these include a file containing profile data for cross-cutter applications). A description of how to generate these files appears in Chapter 3.4.2.



Caution!

The cam files supplied (CMM, CAM and LC9 files) serve simply as a sample project for creating application-specific profile data and must not be used in the context of actual systems.



1.3 General information about the prepared solution

Lenze's prepared solutions are designed to help users implement complex drive functions in practical applications. Prepared solutions are supplied with the complete functionality of a machine unit (core functionality plus peripheral functionality) and can be put into operation quickly by performing just a few steps.

Firstly, this manual describes the basics of the application supported by the prepared solution. This familiarises the user with the fundamental features of the system and provides the user with the requisite basic understanding for installing and starting up the prepared solution and adapting it to meet the needs of his or her application.

Secondly, it provides detailed information about specific technical features (e.g. it lists the necessary components, user parameters and variables, describes the installation and start-up process step by step, along with diagnostics and fault elimination) in order to facilitate the rapid application of the prepared solution.



1.4 Functional range and variants

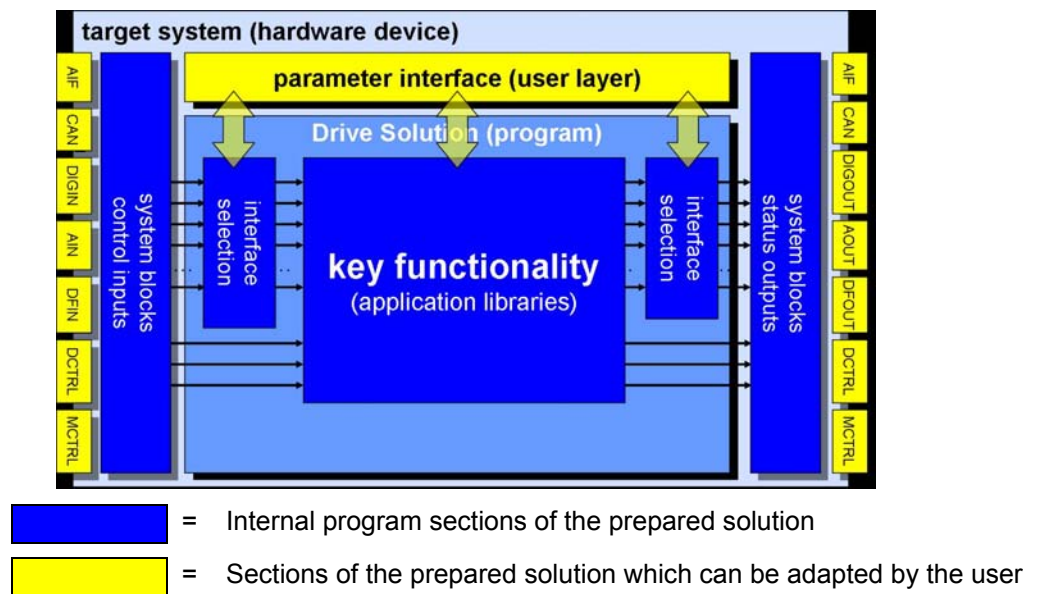
Prepared solutions essentially comprise a variety of files which provide users with a basis for their applications (the files used will vary depending on the application in question). Two variants of prepared solutions are available: In the parameterisable variant, the prepared solution reduces installation and start-up times to an absolute minimum with predefined interfaces and functions. The programmable variant provides an ideal platform for flexible adaptations to specific customer requirements. Corresponding control and status interfaces are available in accordance with whether the user is working with the parameterisable or the programmable variant.

	<i>Parameterisable variant</i>	<i>Programmable variant</i>
Predefined interfaces	✓	—
Free definition of the user interface	—	✓
Installation and start-up time reduced to a minimum thanks to parameter setting	✓	—
Functional adaptations/expansions by means of dedicated program sections created in the IEC programming languages	—	✓



1.4.1 Parameterisable variant

In the parameterisable variant, all basic functions and the selection of interface signals are set solely via parameters (codes). This provides the user with a standardised predefined interface. Typical signal wirings, terminal assignments, interfaces to data bus systems, etc. can be selected via parameters (codes). This means that the parameterisable variant of the prepared solution can reduce installation and start-up times to a minimum without restricting the functionality of the prepared solution:



When using a prepared solution, the system is configured entirely using code (parameter-setting interface). The functions of the individual codes used in the context of parameter setting are listed in Chapter 6.2.1.



You will need the following Lenze software to use the parameterisable variant:



Global Drive Control (GDC):
for operation, parameter setting and diagnostics



Global Drive Loader (GDLoader):
to upload the project's binary file and application data (profile data)



Global Drive Oscilloscope (GDO):
for diagnostics purposes and to record temporal characteristics

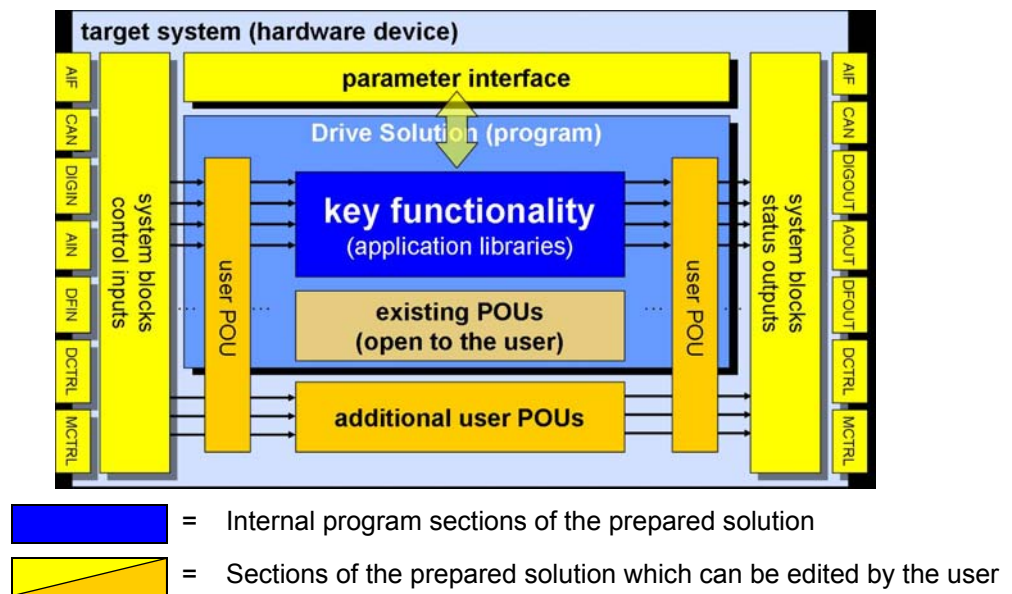


Cam Designer Basic/Cam Designer Professional:
to generate basic cams



1.4.2 Programmable variant

The programmable variant supports the functional range of the parameterisable variant but also allows the user to adapt the prepared solution program on the basis of IEC61131 (e.g. in respect of the definition of interfaces, adaptation in the external signal flow, etc.) or to add more user-specific program sections. This means that the prepared solution can be adapted specifically to meet individual customer requirements. Furthermore, the resources of the target platform can be put to optimum use without affecting core functionality.



In the programmable variant, the user is entirely free to both configure the signal flow outside the core function and change/adapt/expand existing program sections of the prepared solution (e.g. error handling). The interface between program sections containing the core functionality and between user program sections, as well as between system blocks, is provided by global variables (similar to the technique used in software package templates). The meanings of the global interface variables are listed in Chapter 6.1.



Note!

The core functionality of the prepared solution cannot be accessed by the user in either variant. This is a deliberate feature designed to prevent malfunctions being programmed or monitoring mechanisms disabled unintentionally.



You will need the following Lenze software to use the programmable variant:



Drive PLC Developer Studio (DDS):

to adapt the prepared solution program (editing of the PRO file)



Global Drive Control (GDC):

for operation, parameter setting and diagnostics



Global Drive Loader (GDLoader):

to upload the project's binary file and application data (profile data) if the customer project created needs to be duplicated



Global Drive Oscilloscope (GDO):

for diagnostics purposes, optimisations and to record temporal characteristics



Cam Designer Basic/Professional:

to generate basic cams



Cam Software Package:

contains the technology libraries for the electronic cam technology (used in conjunction with the DDS software tool)



1.5 Conventions used

This Manual uses the following conventions to distinguish between the different types of information:

Type of information	Representation	Example
Names of dialog boxes, input fields and selection lists	<i>italic</i>	The <i>Options</i> dialog box
Names of identifiers (names of variables) and program organisation units (POUs = functions, function blocks, programs)	<i>italic</i>	Setpoint generation for the drive takes place in the <i>MotionControl</i> section of the program.
Buttons	bold	Click OK to...
Menu commands	bold	You can use the Messages command to...
		If several commands need to be carried out in succession in order to execute a function, the individual commands will be separated by an arrow: Select File → Open to...
Keyboard commands	bold	Press <F2> to open the input assistance.
		A "+" will appear in between commands to indicate that a command requires a key combination: Press <Shift> + <ESC> to...
Program listing	Courier	IF var1 < var2 THEN...
Key words	Courier bold	...starts with FUNCTION and ends with END FUNCTION .
Important note		Caution! Do not use the Online → Controller inhibit command for an emergency stop via the PC as this command is sent to the controller subject to a delay.
Tip		TIP! Hover the mouse pointer over a symbol on the toolbar to display a tooltip indicating the corresponding command.

Variable names

The conventions used by Lenze for the variable names of its system blocks, function blocks and functions are based on what is known as "Hungarian Notation". This allows you to identify the most important properties (e.g. the data type) of a variable immediately by means of its name, e.g. DIGIN_bln1_b.

You will find information about the conventions in the appendix of the DDS Online documentation "Introduction to IEC1131-3 programming".

2 Features of the "Cross Cutter" prepared solution

2.1 General features of the "Cross Cutter" prepared solution

A brief list of the features of the "Cross Cutter" prepared solution appears below:

- Basic program for a cross-cutter application requiring high cutting length tolerance. The following individual functions are supported:
 - Synchronous or asynchronous operation of the cross cutter
 - Automatic calculation of motion profiles based on basic profile data
 - Selection of three interpolation modes to calculate motion profiles
 - Length-controlled or mark-controlled operation
 - Adjustable compensation speed for mark correction
 - Mark window
 - Mark register
 - Manual x trimming
 - Continuous cutting
 - Test cut
 - Reject operation with activation of a reject gate
 - Oversynchronous/undersynchronous cutting
 - Torque precontrol
 - Evaluation of absolute value encoders at load and drive end
 - Manual jog
 - Simple positioning
 - Homing with 8 different homing modes
- A detailed description of basic cam generation: The application is based on the "Electronic cam" technology function and uses motion profiles (cams) created with Lenze's **CamDesigner Basic** software for positioning. A detailed description of cam generation with **CamDesigner Basic** enables the user to create these motion profiles adapted to meet the needs of the given application.
- Sample project for Lenze's **CamDesigner Professional** software tool to illustrate the cam generation process: Contents include a sample project file (CMM file), the associated motion profiles as source files (CAM files) and the compiled cam file (LC9 files) for this sample.



Caution!

Application-specific profile data is required for the actual application; a description of how to create this data appears in Chapter 3.4.2. The sample project (CMM, CAM and LC9 files) must not be used in the context of actual systems.

Cross Cutter

Features of the "Cross Cutter" prepared solution

2.2 Structure and functional principle

2.2.1 Typical structure of a cross-cutter application

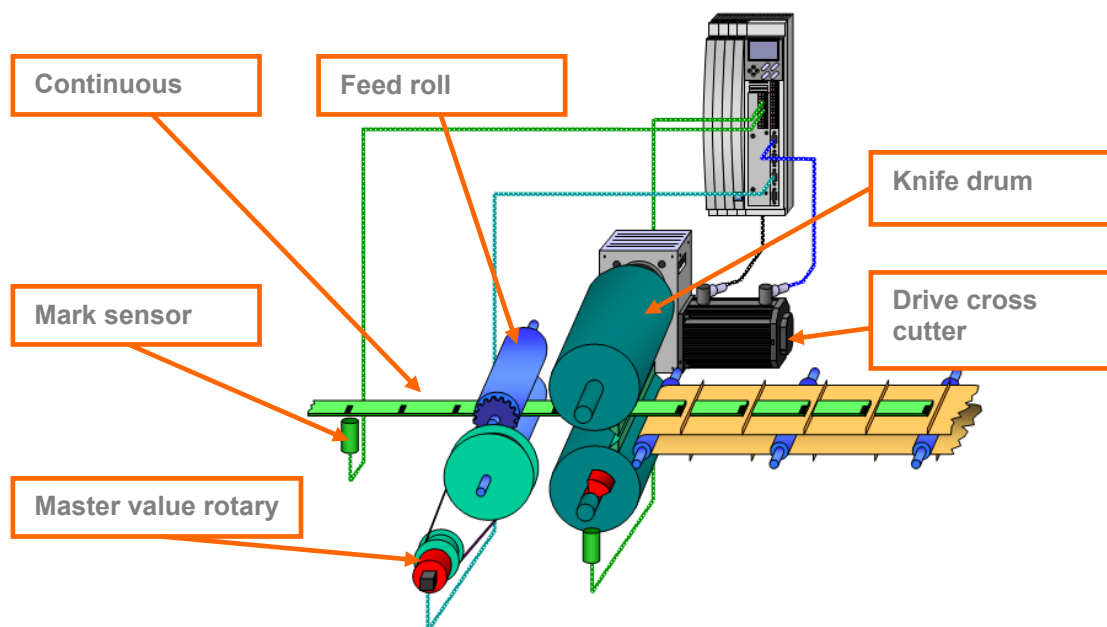


Figure 1: Typical structure of a cross-cutter application

The prepared solution is able to support applications in which continuous material is cut to length in defined cutting lengths. A fundamental distinction is drawn between asynchronous and synchronous cross cutters. In the prepared solution, the asynchronous cross cutter can be used for cutting lengths shorter than or equivalent to the cutting circle circumference. The synchronous cross cutter can be used for all cutting lengths shorter than or longer than the cutting circle circumference.

Cross Cutter

Features of the "Cross Cutter" prepared solution

2.2.2 Typical motion profiles

Synchronous cross cutter

The following diagrams illustrate the position characteristic (red) and the speed signal (blue) for three different cutting lengths in a cross-cutter application:

Concepts:

$y_{\max}$ = Final value of the slave (cross-cutter knife) motion profile. This value will vary depending on the number of knives on the cross-cutter drum.
For example: 1 knife $\Rightarrow y_{\max} = 360^\circ$; 2 knives $\Rightarrow y_{\max} = 180^\circ$

y_{end} = Final value of the asynchronous motion phase and start of the synchronous motion phase. $y_{\text{end}} = y_{\max} - y_{\text{sync}}$

y_{sync} = Indicates the cutting angle. In this range the knife is synchronised with the line speed.

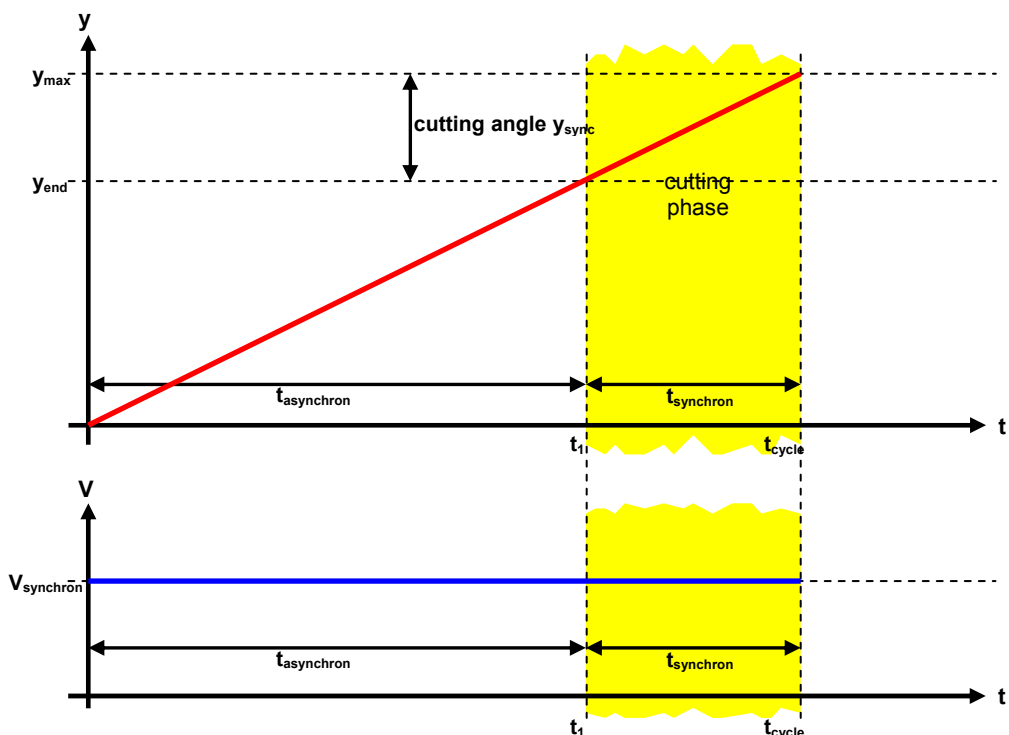
$t_{\text{asynchron}}$ = During this time the cross-cutter knife can move out of synchronism with the line speed.

t_{synchron} = During this time the movement of the cross-cutter knife is synchronised with the line speed.

t_{cycle} = Duration of a cross-cutter revolution

t_1 = Start of the synchronous phase

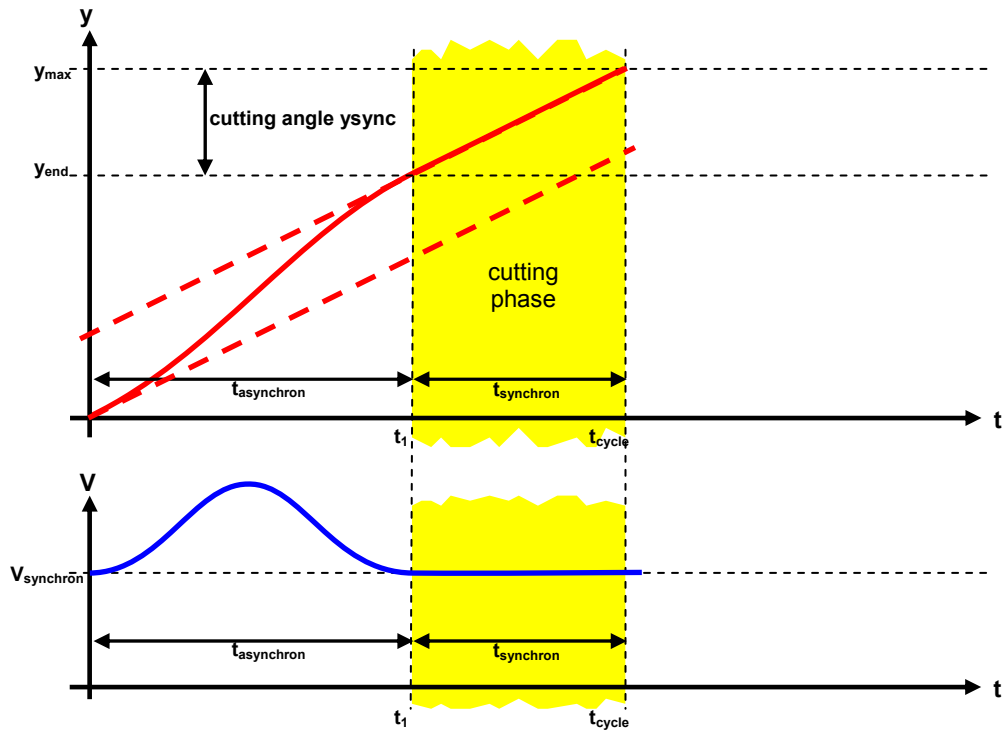
a) Synchronous cutting length
(cutting length $x_{\max} = \text{cutting circle circumference of knife drum } x_{\text{circumference}}$)



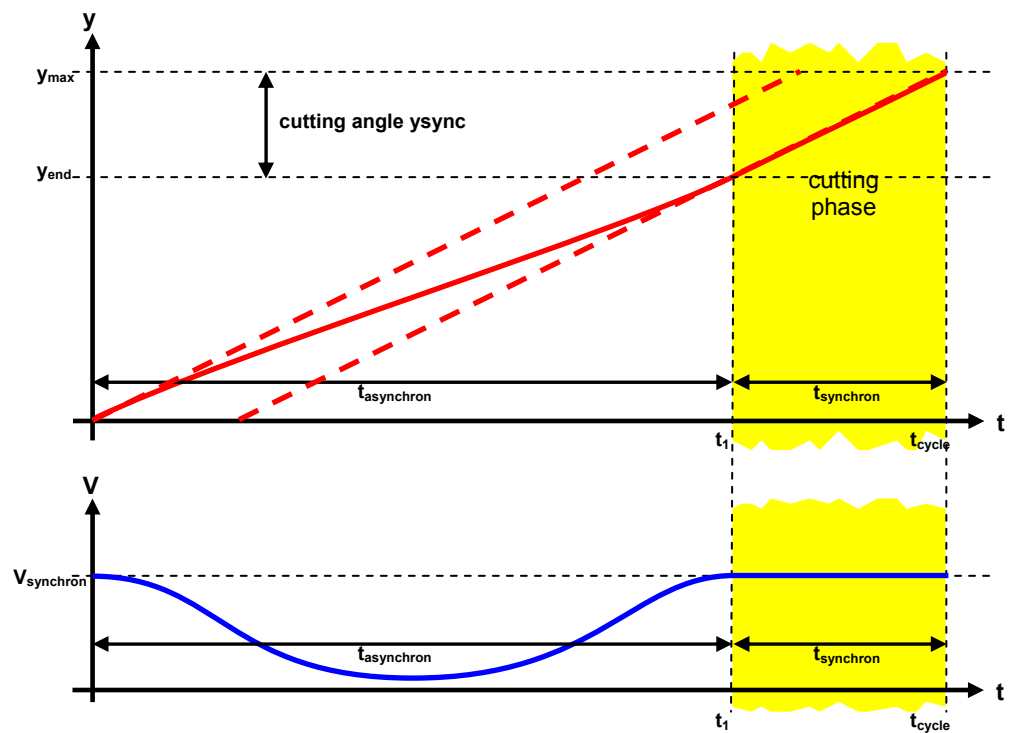
Cross Cutter

Features of the "Cross Cutter" prepared solution

- b) Oversynchronous cutting length
(cutting length $x_{\max} < \text{cutting circle circumference of knife drum } x_{\text{circumference}}$)



- c) Undersynchronous cutting length
(cutting length $x_{\max} > \text{cutting circle circumference of knife drum } x_{\text{circumference}}$)



Cross Cutter

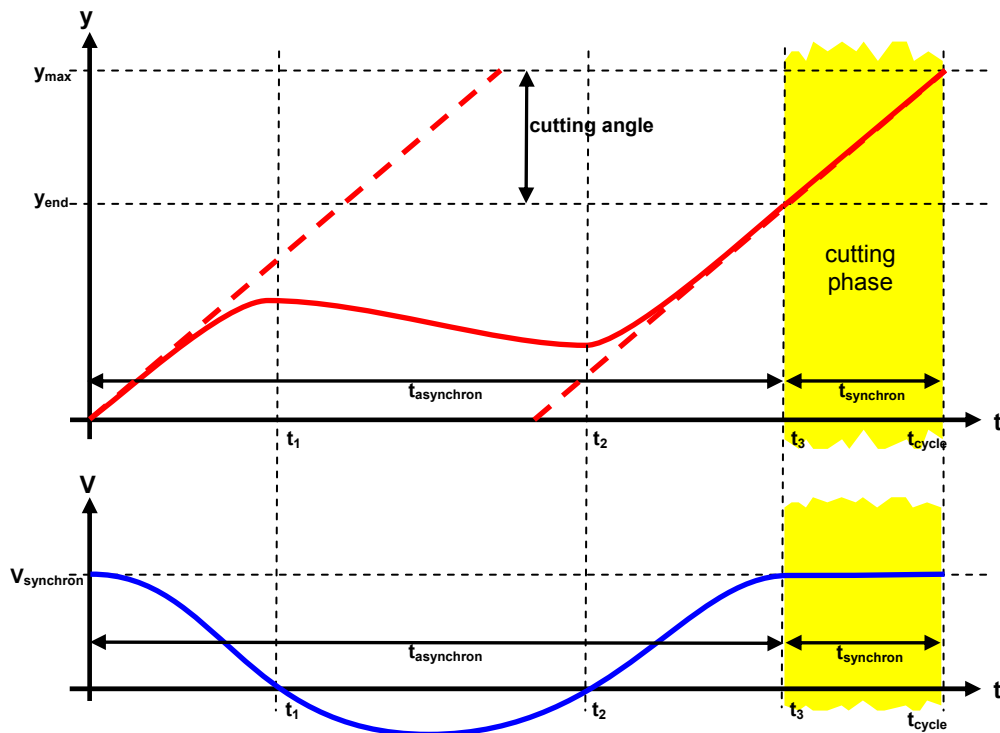
Features of the "Cross Cutter" prepared solution



TIP!

There are a variety of mathematical transfer functions for the motion characteristic in the asynchronous section (see, e.g. VDI 2143). The prepared solution only takes account of second order polynomials (linear speed ramps), fifth order polynomials and the sloping sine line.

Selecting a cutting length $x_{\max}$ a lot longer than the synchronous cutting length ($x_{\text{circumference}}$) is considered to be a special case. In this situation, the speed of the knife drum can fall to values less than zero, causing the knife drum to move in opposition to the direction of material flow:

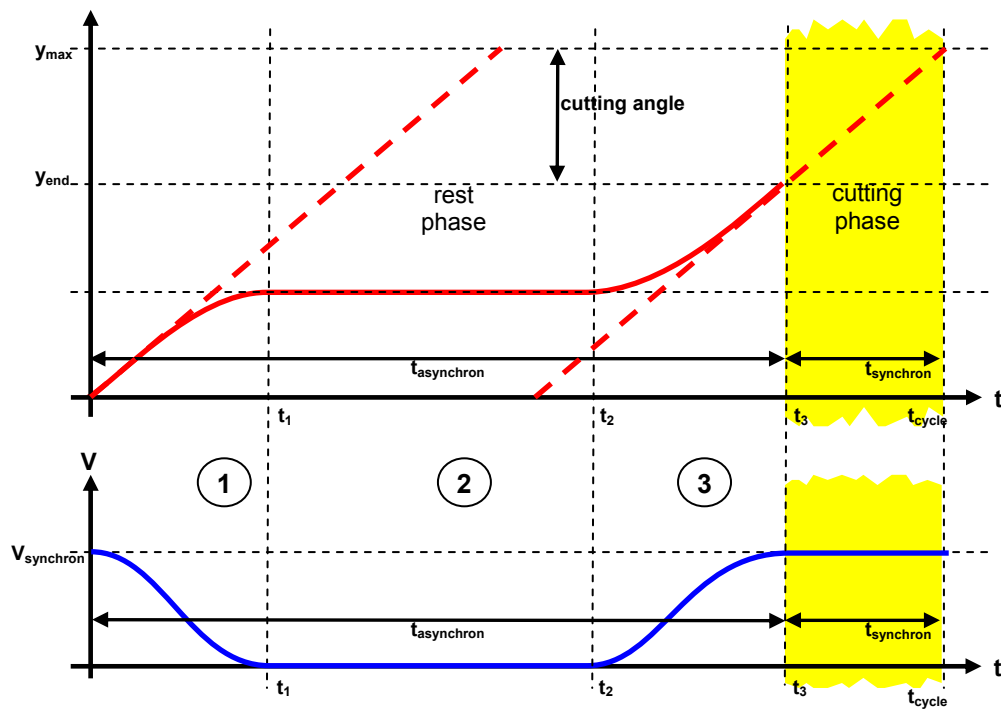


Cross Cutter

Features of the "Cross Cutter" prepared solution

In some cases, this effect may be desirable in order to reduce the dynamics of the knife drum on the basis of longer deceleration/acceleration ramps for exiting/entering synchronous mode.

In most cases, movement in opposition to the material flow is not permissible. In order to avoid reversing the knife drum it makes sense to split the asynchronous section into three subsections with a pause in the middle (section 2):



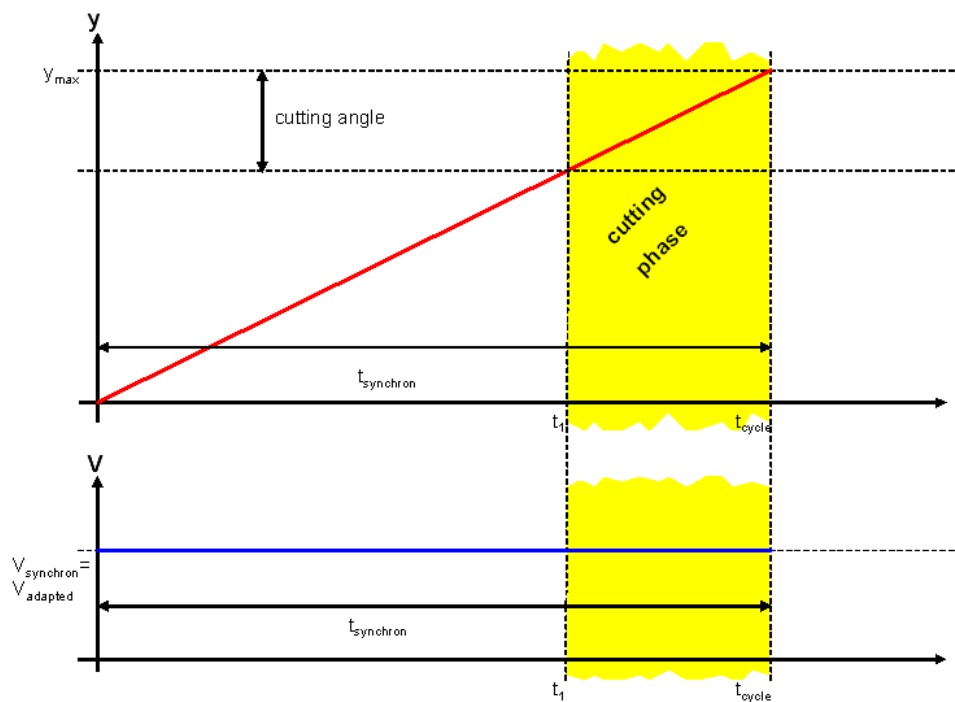
Cross Cutter

Features of the "Cross Cutter" prepared solution

Asynchronous cross cutter

The following diagrams illustrate the position characteristic (red) and the speed signal (blue) for three different cutting lengths in an asynchronous cross-cutter application:

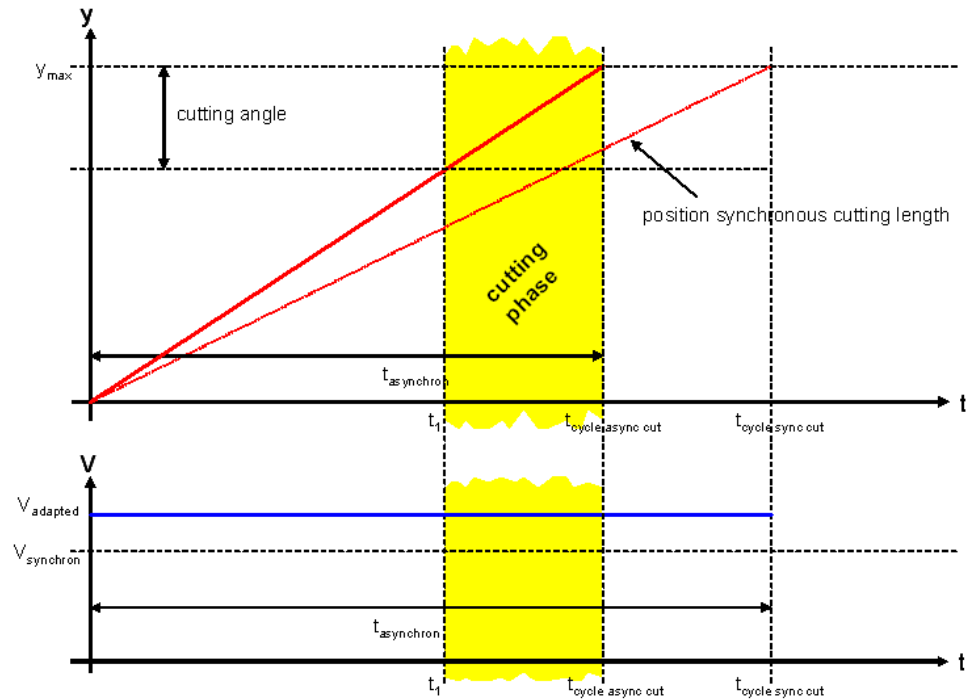
- a) Synchronous cutting length (cutting length $x_{\max}$ = cutting circle circumference of knife drum $x_{\text{circumference}}$; circumferential speed = line speed)



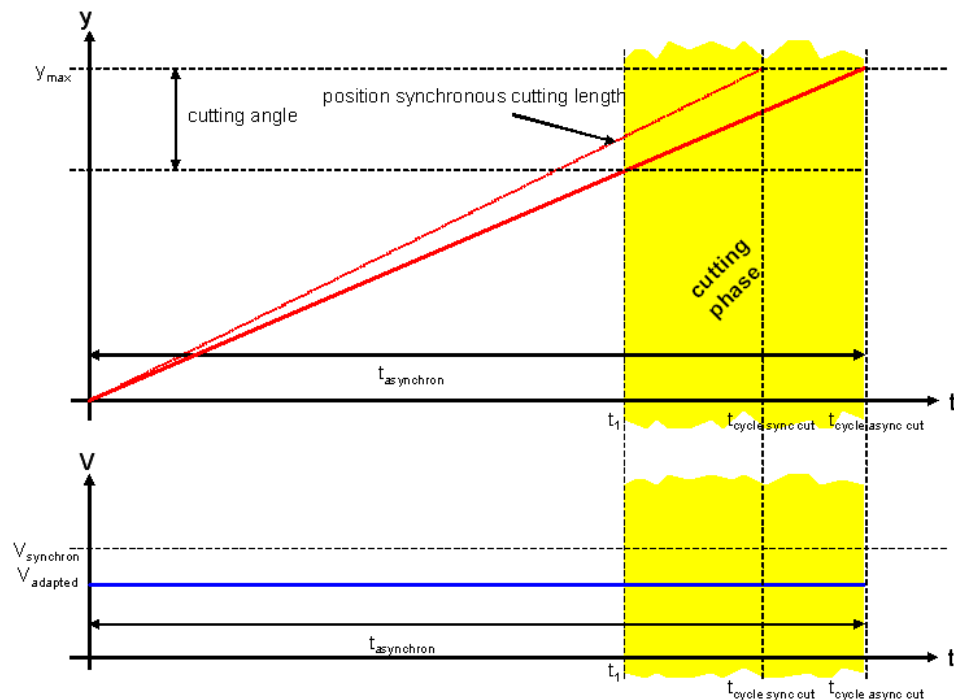
Cross Cutter

Features of the "Cross Cutter" prepared solution

- b) Oversynchronous cutting length (cutting length $x_{\max} < \text{cutting circle circumference of knife drum } x_{\text{circumference}}$; $V_{\text{adapted}} = \text{linespeed} \cdot \frac{\text{cutting circle circumference}}{\text{cutting length}}$)



- c) Undersynchronous cutting length (cutting length $x_{\max} > \text{cutting circle circumference of knife drum } x_{\text{circumference}}$; $V_{\text{adapted}} = \text{linespeed} \cdot \frac{\text{cutting circle circumference}}{\text{cutting length}}$)



Cross Cutter

Features of the "Cross Cutter" prepared solution

2.2.3 Example applications of the "Cross Cutter" prepared solution

Cross cutters are used in virtually every area of industrial production. A list of some typical applications appears below.

- Synchronous cross cutters in the paper industry, e.g. to cut paper webs and paper board containers
- Plastics processing industry downstream of extrusion lines to assemble products
- Application in the wood processing industry to cut laminate panels
- In the packaging industry to seal or emboss packaging
- Print and media industry for format cutting
- Beverage industry

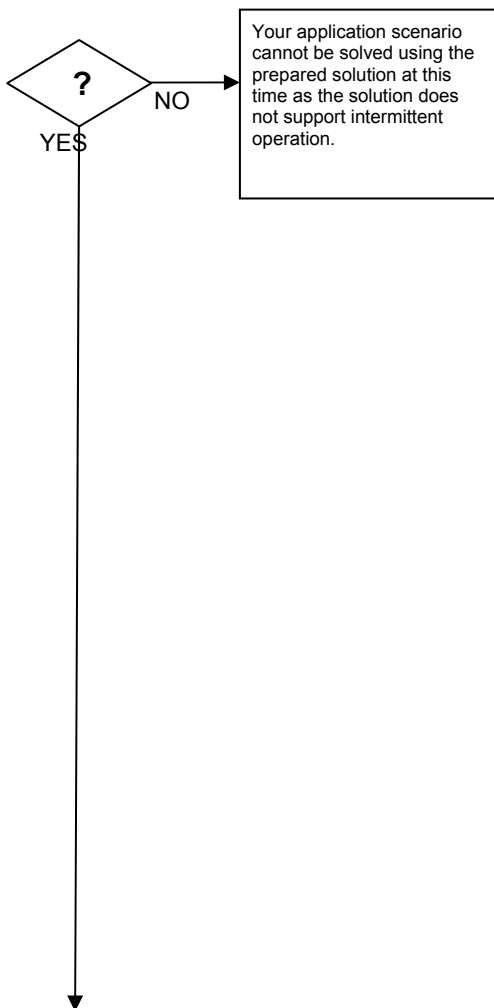
Cross Cutter

Features of the "Cross Cutter" prepared solution

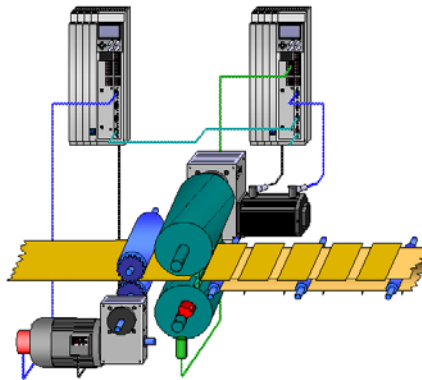
2.3 Functional application criteria for the parameterisable variant

In order to enable you to check the compatibility of the prepared solution for your application and estimate its usefulness, a list of decision-making criteria and limiting conditions to which the use of the prepared solution is subject appears below.

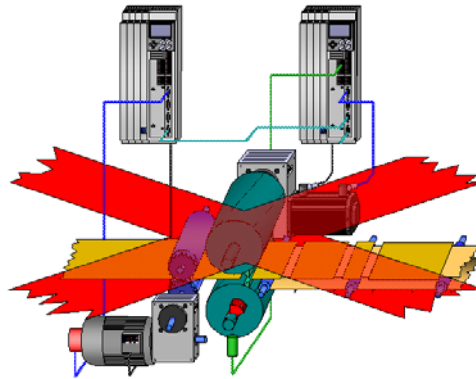
How do you intend to process the material?



Continuous cutting: You intend to cut panels of material to the same length continuously; the cutting length can be varied at will and the new cutting length applied once the calculation is complete. This operation is supported by the prepared solution.

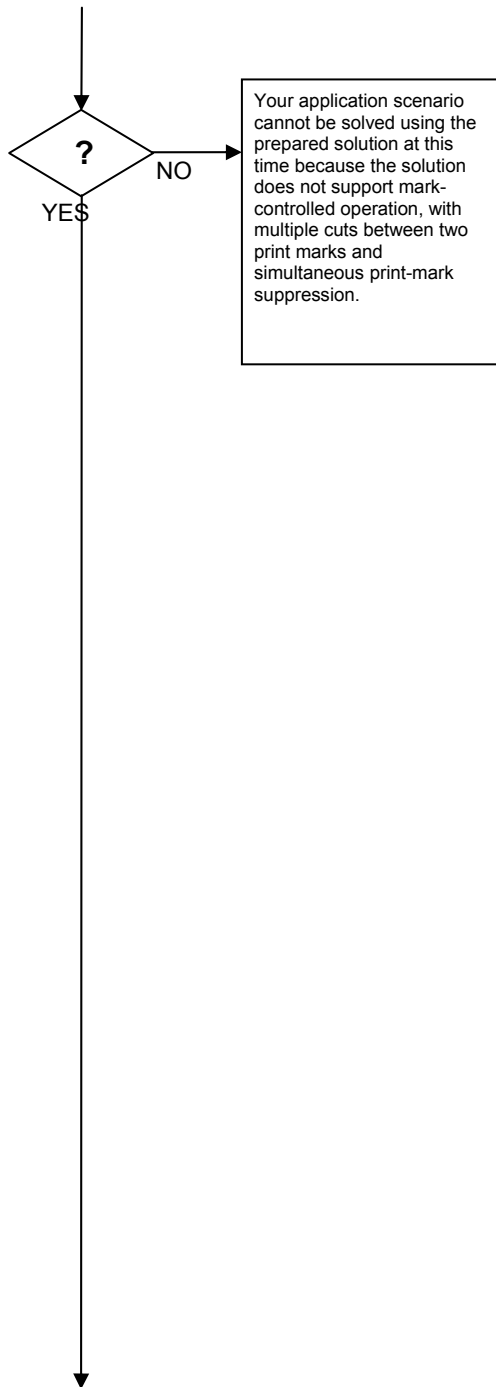


Intermittent operation: You intend to cut panels of material to different lengths continuously. This operation is not supported by the prepared solution.



Cross Cutter

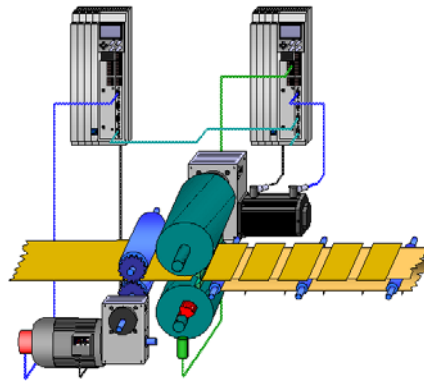
Features of the "Cross Cutter" prepared solution



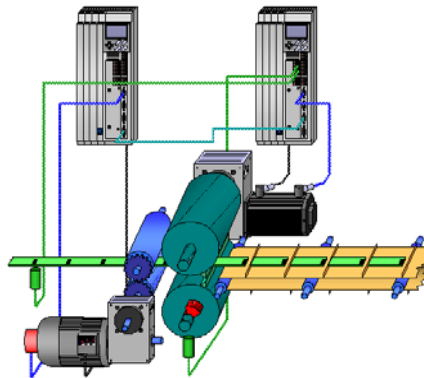
Are the combinatorial options sufficient for your application case?

Various combinatorial options can be considered for the use of the cross cutter depending on the application.

1. **Length-controlled operation:** In this operating mode the material to be cut has no print marks and the required cutting length is determined solely by the calculated motion profile. In this operating mode, the cross cutter can be changed over when idle and during operation.



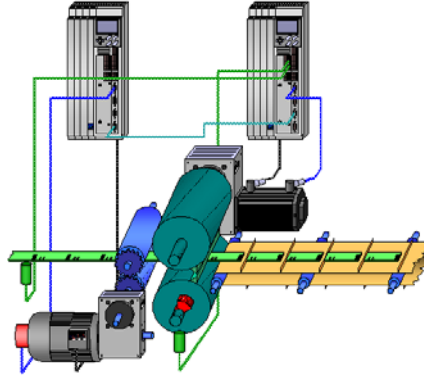
2. **Mark-controlled operation, without print-mark suppression:** In this operating mode the cross cutter works with mark control and higher-level length control. A mark is made on the material wherever a cut needs to be made. In this operating mode, the cross cutter can be changed over when idle and during operation.



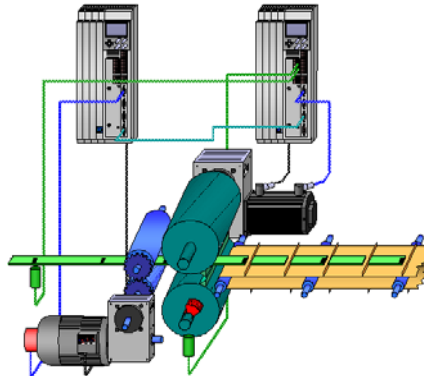
Cross Cutter

Features of the "Cross Cutter" prepared solution

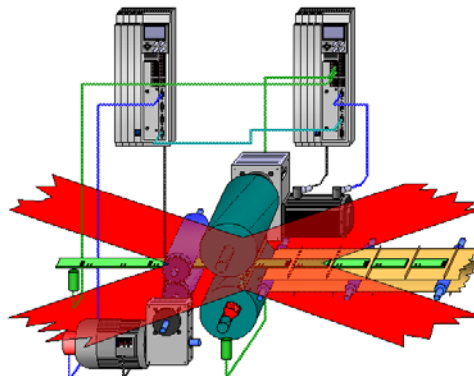
3. Mark-controlled operation, with print-mark suppression: In this operating mode, the cross cutter works with mark control and higher-level length control. A mark is made on the material wherever a cut needs to be made. Other marks which have to be ignored by the system appear before and after the print mark relevant for the cut. In this operating mode, the cross cutter can be changed over when idle. However, it cannot be changed over from operating status to idle state!



4. Mark-controlled operation, with multiple cuts (divisions) between two print marks: In this operating mode the print-mark distance between print marks is a multiple of the required cutting length. The cross cutter works with mark control and higher-level length control. In this operating mode, the cross cutter can be changed over when idle and during operation.

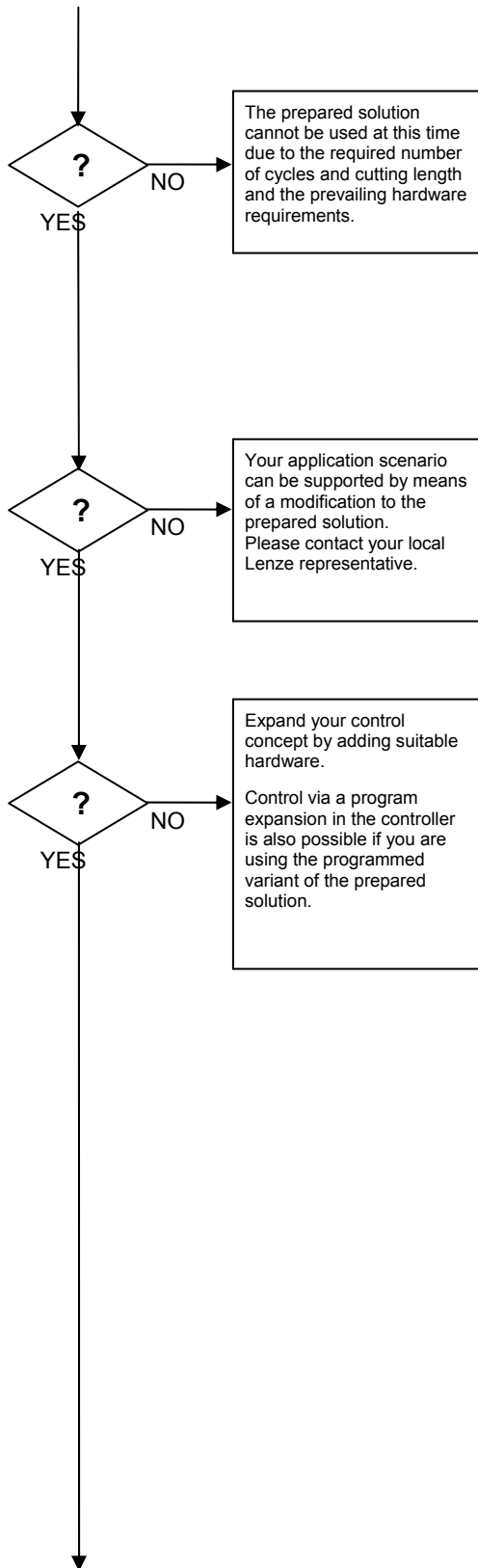


5. Mark-controlled operation, with multiple cuts between two print marks and print-mark suppression: In this operating mode the print-mark distance is a multiple of the required cutting length. Other marks which have to be ignored also appear in between the relevant print marks. The prepared solution does not support this type of operation!



Cross Cutter

Features of the "Cross Cutter" prepared solution



Is your application able to support the following physical requirements?

The following essential issues affecting design and configuration will have a bearing on whether or not you can use the prepared solution in your application:

- Number of knives on the knife drum (the prepared solution will support up to 6 knives on the knife drum)
- Maximum number of cycles on the system
- Maximum possible speed of the cross-cutter drum
- Maximum possible material speed

The required cutting length and required number of cycles may result in a speed which the cross-cutter drive is unable to support. Codes C3686 to C3689 indicate the maximum speed of a cutting profile for a specific cutting length.

Is your application able to support this process sequence?

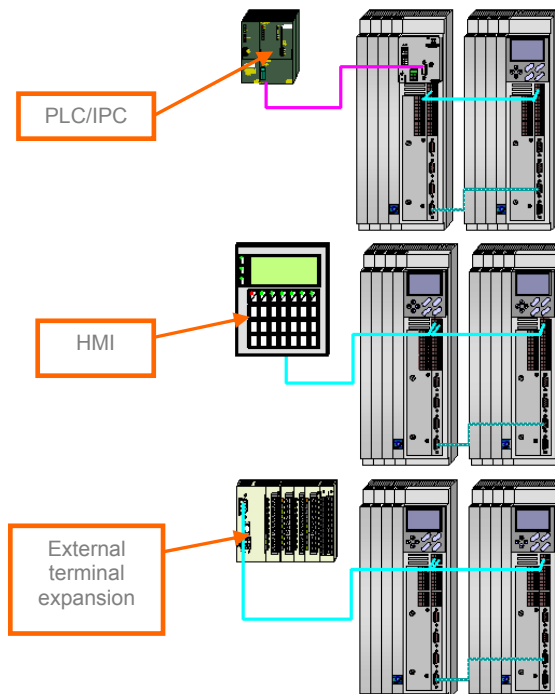
Process sequence

The process can only be started if the cross cutter is in the initial position:

- The knife/knives are in the initial position.
- There are no print marks in the sensor's detection range.

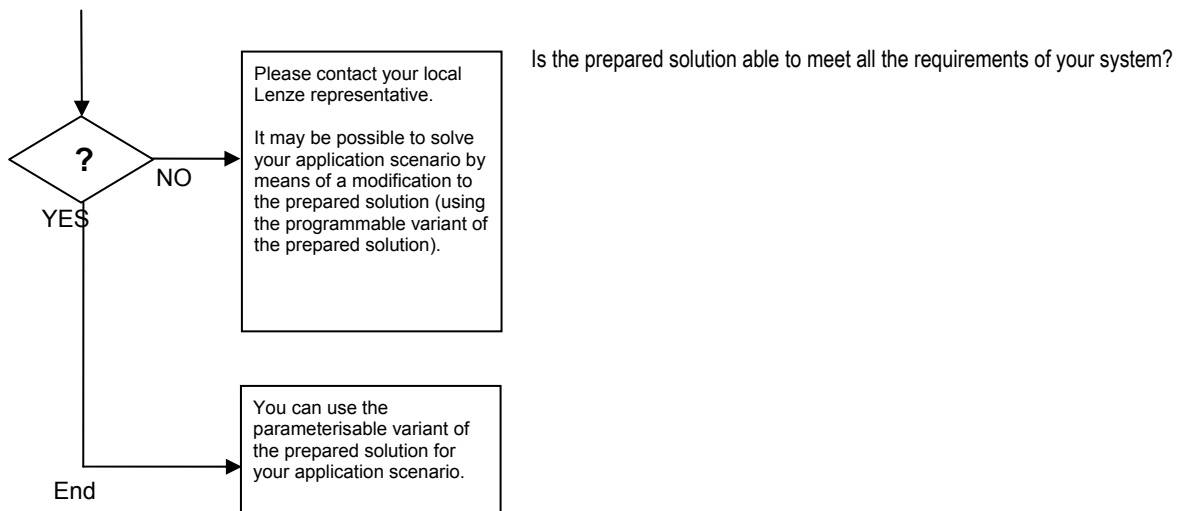
Is a higher-level PLC/IPC connected via a bus system, is a terminal extension (e.g. Lenze EPM terminal extensions) or an HMI connected via the system bus or the automation interface (AIF)?

In addition to the controllers, other components are required to generate the control signals for the drive and evaluate states and conditions:



Cross Cutter

Features of the "Cross Cutter" prepared solution



Cross Cutter

Features of the "Cross Cutter" prepared solution

2.4 Machine parameters

The machine parameters define the relationship prevailing between the user's measuring system and the incremental measuring system (the controller's internal measuring system). Where the user's measuring system is concerned, a distinction is drawn between slave units ([s_units]) and master units ([m_units]). The cross-cutter drum's measuring system is provided by the slave units. In this context, slave units are usually scaled in degrees. The feed axis' measuring system is provided by the master units which, depending on the application, might be scaled in [mm] or [inches]. The relationship between a motor revolution and a feed motion in the user's measuring system is established by means of the following variables:

- Feed constant: This indicates the number of application units by which the load will move forward when the drive completes a full revolution at the gearbox output end. It is usually specified in [units/rev].
Example: On a spindle drive the feed constant is equivalent to the pitch of the spindle, scaled in [mm/revolution].
- Gearbox ratio, divided into numerator and denominator: Both the numerator and the denominator are usually specified as integers (corresponding to the number of teeth in the gearbox stages).
- Resolution of a motor revolution: Internally, the ServoPLC resolves a motor revolution into 65536 (=2¹⁶) increments.

The conversion between the two measuring systems can be carried out with knowledge of these mechanical machine constants by applying the following formulae:

a) Conversion of application units [units] into incremental units [incr.]:

$$s[incr.] = s[s_units] \cdot \frac{Z_i}{N_i} \cdot \frac{65536[incr./rev]}{m_{FeedConstant}[units/rev]}$$

a) Conversion of incremental units [incr.] into application units [units]:

$$s[s_units] = s[incr.] \cdot \frac{N_i}{Z_i} \cdot \frac{m_{FeedConstant}[units/rev]}{65536[incr./rev]}$$

where	Z_i	= gearbox ratio numerator
	N_i	= gearbox ratio denominator
	$m_{FeedConstant}$	= feed constant

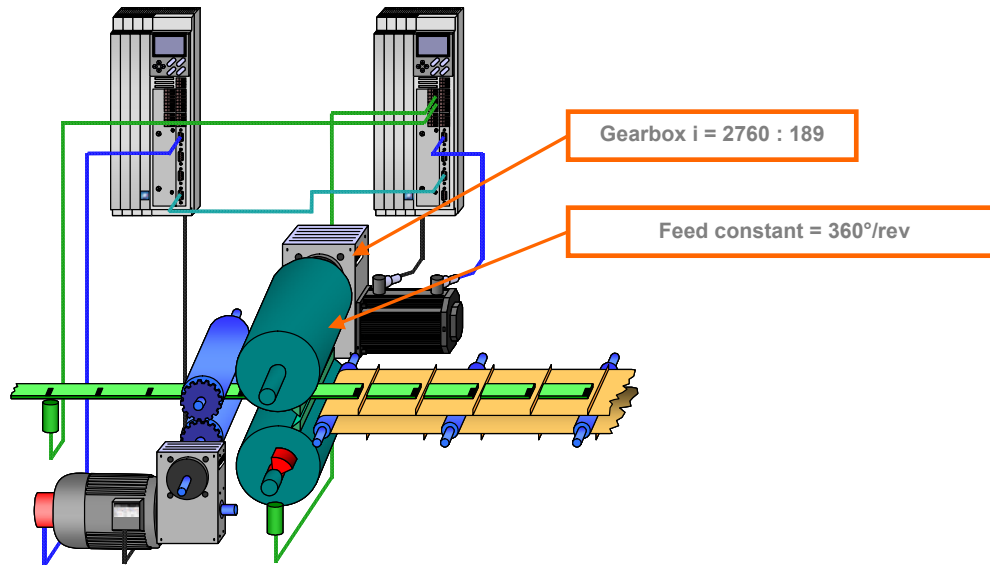
In the context of the Cross Cutter prepared solution, the mechanical machine constants are defined via profile data (LC9 file). A description of how this data is generated appears in the installation and start-up instructions (see Chapter 3.4.2).

Cross Cutter

Features of the "Cross Cutter" prepared solution

Example:

A position $s[\text{units}] = 165^\circ$ needs to be converted into an incremental value. For this example, the characteristics of the mechanical system are as follows:



$$s[\text{incr.}] = s[\text{units}] \cdot \frac{Z_i}{N_i} \cdot \frac{65536[\text{incr./rev}]}{m_{\text{FeedConstant}}[\text{units/rev}]} = 165^\circ \cdot \frac{2760}{189} \cdot \frac{65536[\text{incr./rev}]}{360^\circ/\text{rev}} = 438640 [\text{incr.}]$$



TIP!

If you are using the parameterisable variant of the Cross Cutter prepared solution, values in application units will be converted into incremental values automatically.

The machine parameters can be read from the following display codes using a parameter setting tool (e.g. **GDC** or **DDS**). The machine parameters are set using the **CamManager/CamDesigner** software tool (see Chapter 3.4.2). The values displayed in the codes are read from the profile data associated with the relevant controller (LC9 file):

Code	Possible settings:			Comments
	Default	Selection		
C3303	-	0 ...	{1} ... 65535	Display code: Gearbox ratio in master value phase (numerator)
C3304	-	0 ...	{1} ... 65535	Display code: Gearbox ratio in master value phase (denominator)
C3305	-	0.0000...	{0.0001 [m_units]/rev} ... 214748.0000	Display code: Feed constant in master value phase in application units per gearbox revolution
C3306	-	0 ...	{1} ... 65535	Display code: Gearbox ratio in cross-cutter train (numerator)
C3307	-	0 ...	{1} ... 65535	Display code: Gearbox ratio in cross-cutter train (denominator)
C3308	-	0.0000...	{0.0001 [m_units]/rev} ... 214748.0000	Display code: Feed constant in cross-cutter train in application units per gearbox revolution
C3636	0	0: Direct 1: Inverted		Possible setting for motor mounting position

Cross Cutter

Features of the "Cross Cutter" prepared solution

2.5 Basic features

2.5.1 Reposition unlocked drive to motion profile

The Prepared Solution 'Cross Cutter' offers different ways to reposition on the motion profile. Regarding this, three different modes are made available to the user. The tool can either be repositioned on the motion profile on the shortest way possible, in clockwise (cw) or counter-clockwise (ccw) direction. The repositioning mode in one direction only (cw or ccw) must be used whenever it is only allowed to move the tool in one direction only.

Variable names

Global variables		(Variable type: VAR_GLOBAL)	
Name	Data type	Value/Meaning	
g_byLockToCamMode	BYTE	Positioning mode Lock to cam	
		0	Move to cam position via shortest route.
		1	Move to cam position only in positive direction (CW).
		2	Move to cam position only in negative direction (CCW).

Code	Possible settings:		Comment
	Default	Selection	
C3630/000	0	0 {1}	2 Positioning mode Lock to cam
			0 Move to cam position via shortest route.
			1 Move to cam position only in positive direction (CW).
			2 Move to cam position only in negative direction (CCW).

Cross Cutter

Features of the "Cross Cutter" prepared solution

2.5.2 Start and Stop Cross Cutter Operation

Start and stop the cross cutter operation is done with the application control word. With application control word bit 14 the cross cutter operation is started or stopped. By starting cross cutter operation (application control word bit 14 = TRUE) the cross cutter is moving in its basic position (C3694), if it isn't already there. When the cross cutter reaches this position the status switches to Cam Operation (C3505=50). From this basic position (C3694) the cross cutter starts movement with the start cam and accelerate from speed zero up to line speed, so that the cross cutter is synchronous to line speed during the cut. The speed of the cross cutter is a result of the start cam profile and the incoming line speed and is not dependent on an acceleration ramp. This means, that there is no movement of the cross cutter as long as there is line speed zero. To stop the cross cutter operation the application control word bit 14 have to set to FALSE. By doing this the last started cut of the cross cutter will be finished and then the cross cutter runs on the stop cam from line speed to speed zero and stops in the basic position, where he switches to status Standby (C3505=10). To stop the cross cutter with application control word bit 14 it is necessary that the incoming line speed is bigger then zero. This is necessary because otherwise the cross cutter can't come in its basic position via the stop cam. Is the incoming line speed zero and the cross cutter should be stopped and should come to status Standby (C3505=10) it is possible to abort the cross cutter operation with application control word bit 24 = TRUE. To abort the cross cutter operation the line speed should be zero otherwise there is a possibility of jerk and of a follow error trip.

The basic position of the cross cutter depends on the number of knives and on the cutting angle size φ . Since the basic position is indicated under code C3694, the user does not need to calculate it. However, if desired the basic position of the cross cutter could be calculated by means of the following formula:

$$Y_{StartPos} = \frac{\frac{360^\circ}{n_{NumberOfKnives}} - \varphi}{2}$$

With: $Y_{StartPos}$ = Basic position of the cross cutter

φ = Cutting angle (synchronous angle) in degree

$n_{NumberOfKnives}$ = Number of knives on the cross cutter drum

It is recommended to stop the cross cutter at the basic position when homing has been completed. As a result, the cross cutter does not move during lock to cam operation. The following figure shows where the important positions, such as basic position, cutting angle, start- and stop point of the motion profile, are located on the cross cutter drum.

Cross Cutter

Features of the "Cross Cutter" prepared solution

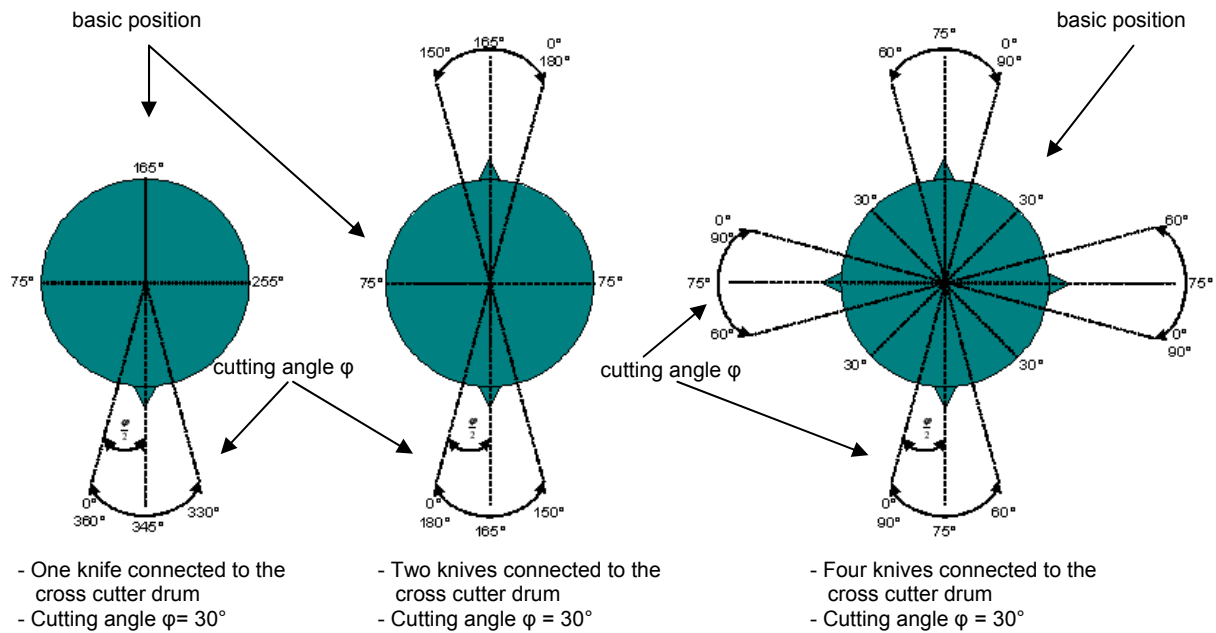


Figure 2: Important motion profile positions dependent on the number of knives

Variable names

Global variables		(Variable type: VAR_GLOBAL)	
Name	Data type	Value/Meaning	
g_bStartStopOperation	BOOL	Start and stop of cross cutter operation	
		TRUE	The cross cutter moves to its base position and starts cross cutter operation as long as there is line speed.
		FALSE	Cross cutter operation stops and the cross cutter moves to its base position.
g_bAbordCrossCutter	BOOL	Abort of cross cutter operation	
		TRUE	Abort Cross cutter operation Attention: Use only when line speed is zero!
		FALSE	There is no abort of cross cutter operation.

Code	Possible settings:		Comment
	Default	Selection	
C4135/ 000 Bit 14	0	0 {1}	1 Application control word bit 14: Start and stop of cross cutter operation.
			TRUE Start of operation
			FALSE Stop of operation
C4135/ 000 Bit 24	0	0 {1}	1 Application control word bit 24: Abort of cross cutter operation.
			TRUE Abort of operation
			FALSE No abort of operation
C3694/ 000	-	- {1[s_units]}	- Display code: Basic position or starting position of the cross cutter.

Cross Cutter

Features of the "Cross Cutter" prepared solution

2.5.3 Cross-cutter operating modes

Continuous cutting

In "Continuous cutting" mode, the cross cutter works in continuous mode and makes cuts to a specific cutting length in accordance with the values previously set until the user stops the cutting process or sets a different cutting length. A dedicated calculation block is available for calculating the motion profiles for the cutting process. Two basic cams are reserved for calculating cutting lengths in "Continuous cutting" mode. This ensures that the user can be calculating a new cutting length in the background whilst continuing to cut to a different cutting length. Once the new cutting length has been calculated, the corresponding motion cam can be preselected with the calculated functions and replace the old cutting length as soon as the current cut has been completed. "Continuous cutting" mode can be both length-controlled and mark-controlled (see Chapter 2.5.8). You can switch from "Test operation" and "Reject operation" to "Continuous cutting" mode without first having to stop the system. You must stop the system first in order to switch to "Continuous cutting" mode from "Asynchronous cross cutter" mode.

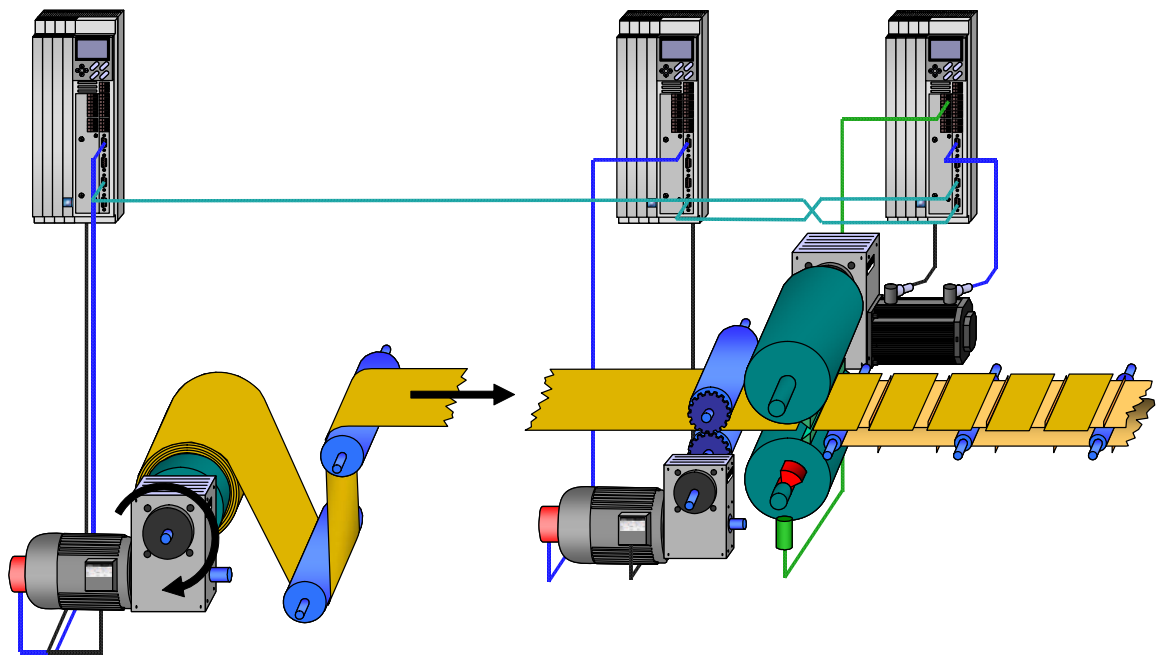


Figure 3: Example representation of continuous operation

Cross Cutter

Features of the "Cross Cutter" prepared solution

Test operation

The user can use the test cut to cut a material sample. This sample can then be used to check cut quality and material characteristics. Furthermore, it provides a means of determining the dimensional accuracy of the cut. A dedicated calculation block is available for calculating motion profiles for test operation. Two basic cams are reserved for calculating test cutting lengths in "Test operation". This ensures that the user can be calculating a new test cutting length in the background whilst continuing to cut to a different test cutting length. Once the new test cutting length has been calculated, the corresponding motion cam can be preselected with the calculated functions and replace the old test cutting length as soon as the current cut has been completed. This means that the user can define parameters for the test cut separately from other parameters for normal operation. "Test operation" can only be length-controlled (see Chapter 2.5.6). You can switch from "Continuous cutting" and "Reject operation" to "Test operation" without first having to stop the system. This means that it is possible to cut a test piece from the material flow during operation. You must stop the system first in order to switch to "Test operation" from "Asynchronous cross cutter" mode.

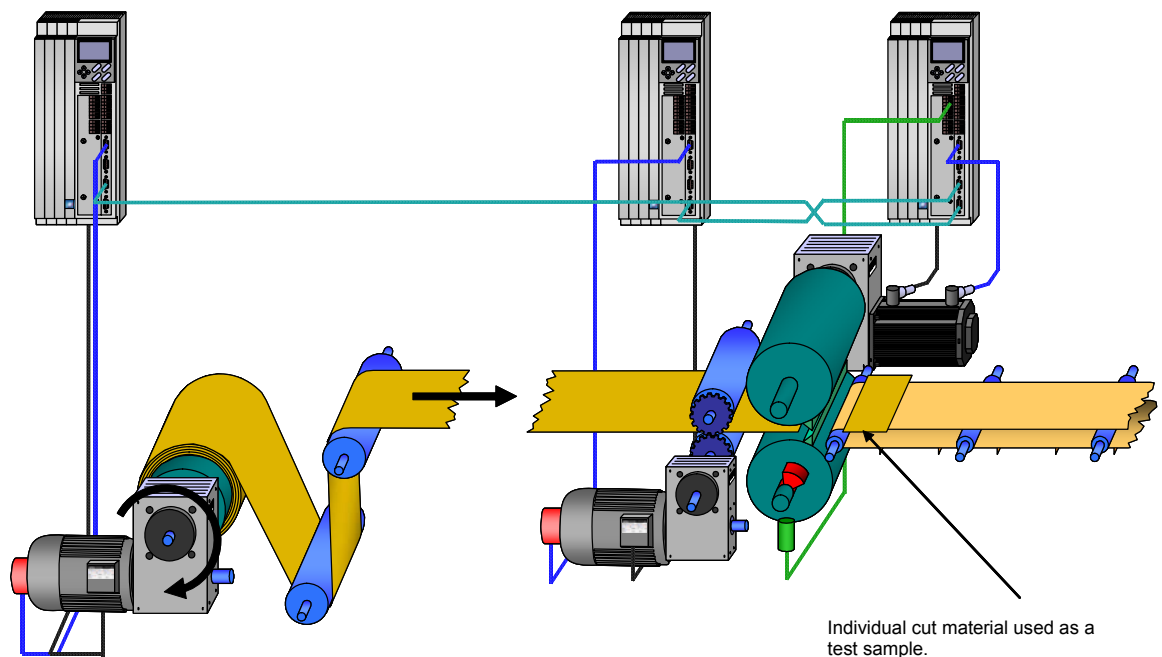


Figure 4: Example representation of test cut operation

Cross Cutter

Features of the "Cross Cutter" prepared solution

Reject operation

In practice, sections of material have to be cut into small pieces in order to be disposed of, for example, in a waste container. Reject operation of this type is required when threading new material or cutting out damaged sections of material. Here too, the user is once again able to define parameters for this application separately. A dedicated calculation block is available for calculating motion profiles for reject operation. Two basic cams are reserved for calculating reject cutting lengths in "Reject operation" mode. This ensures that the user can be calculating a new reject cutting length in the background whilst continuing to cut to a different reject cutting length. Once the new reject cutting length has been calculated, the corresponding motion cam can be preselected with the calculated functions and replace the old reject cutting length as soon as the current cut has been completed. "Reject operation" mode can only be length-controlled (see Chapter 2.5.6). You can switch from "Continuous cutting" and "Test operation" to "Reject operation" without first having to stop the system. This means that it is possible to cut reject sections from the material flow during operation. You must stop the system first in order to switch to "Reject operation" from "Asynchronous cross cutter" mode. The system activates reject gate control as soon as "Reject operation" is selected (see Chapter 2.5.12).

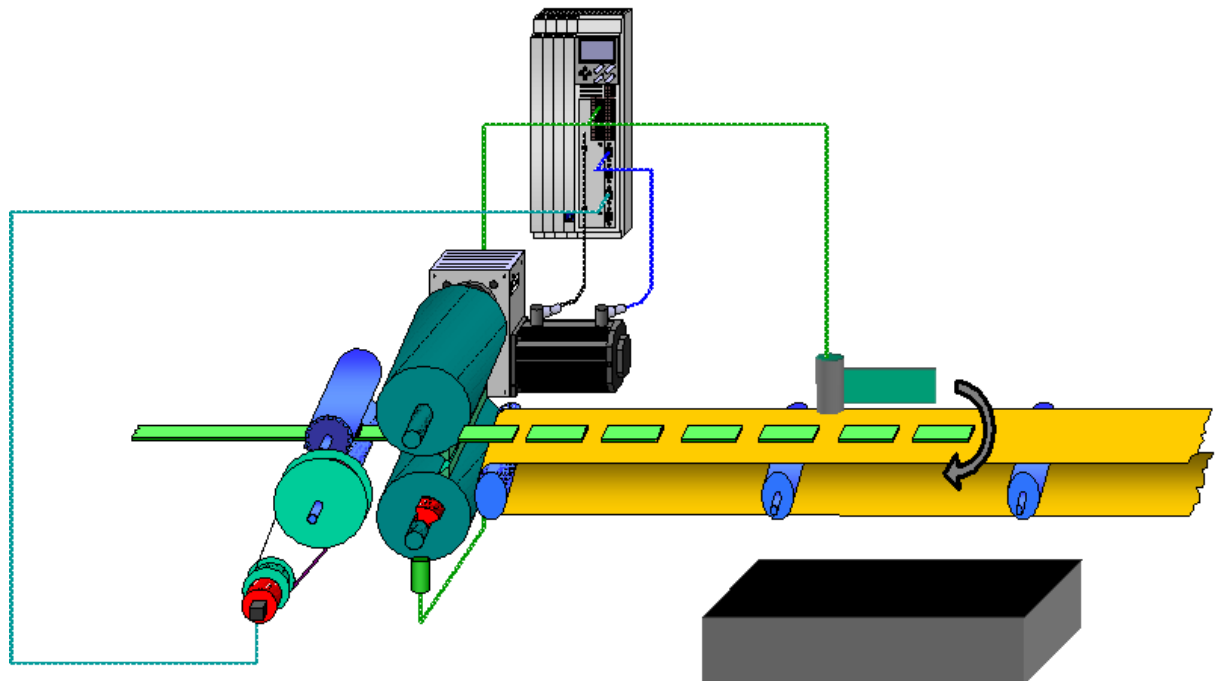


Figure 5: Example representation of reject cut operation

Cross Cutter

Features of the "Cross Cutter" prepared solution

Asynchronous cross cutter

"Asynchronous cross cutter" mode is a separate operating mode. It is frequently used when very short material lengths need to be cut at a very high line speed. However, at the same time, provision must also be made to ensure that the cut is made in asynchronism with the line speed. In the prepared solution the maximum cutting length is limited to the cutting circle circumference of the cross-cutter drum. This ensures that the cut is not made at an undersynchronous speed at which the material to be cut would jam. This means that the material is only ever cut at synchronous or oversynchronous speed. Unlike the scenario on the synchronous cross cutter, the required cutting length is not modified by calculating a motion profile. Instead, the master value is evaluated with an appropriate factor. The motion profile is provided by a basic cam which runs through at a faster or slower rate in accordance with cutting length (however, its speed will never drop below that of the line speed itself). This results in the longest cutting length of the asynchronous cross cutter being equivalent to the cutting circle circumference of the knife drum. The typical motion characteristics of an asynchronous cross cutter are presented in Chapter 2.2.2. Since the line speed is adapted in "Asynchronous cross cutter" mode and switching over to a different mode can cause a speed step change in the master value, measures have been put in place to prevent direct switching to or from this mode. You must stop the system first if you wish to switch from "Asynchronous cross cutter" mode to "Continuous cutting", "Test operation" and "Reject operation". To change the cutting length in asynchronous mode, you must first stop the cross cutter in order to avoid a speed step change.

Cross Cutter

Features of the "Cross Cutter" prepared solution

Variable names

Inputs		
Global variables (Variable type: VAR_GLOBAL)		
Name	Data type	Value/Meaning
g_bStartStopOperation	BOOL	Start and stop the selected cross-cutter operating mode. This value is written in code C4135/000 bit 14. This value is displayed in code C4136/000 bit 14.
		TRUE Start operation.
		FALSE Stop operation.
g_wProfileSelector	WORD	Select cross-cutter operating mode. Note: You can switch between modes 1 to 3 during operation. To switch to mode 4 from one of modes 1 to 3, you will first need to stop the cross cutter via <i>g_bStartStopOperation</i> . To switch from mode 4 to one of modes 1 to 3, you will first need to stop the cross cutter via <i>g_bStartStopOperation</i> . This value is displayed in code C4137/000; this variable can be written to this code.
		1 Synchronous cross cutter, continuous cutting
		2 Synchronous cross cutter, reject operation
		3 Synchronous cross cutter, test operation
		4 Asynchronous cross cutter, continuous cutting
g_bWorkWithNewCuttingLength	BOOL	Activate the variable to immediately apply a new cutting length you have calculated and work with the new cutting length. Note: The variable is only valid for the synchronous cross cutter! A new cutting length for the asynchronous cross cutter is always applied immediately! This value is written in code C4135/000 bit 15. This value is displayed in code C4136/000 bit 15.
		TRUE Apply new cutting length calculated.
		FALSE Do not apply new cutting length calculated, continue to work with the last cutting length calculated.
g_bWorkWithNewRejectLength	BOOL	Activate the variable to immediately apply a new reject length you have calculated and work with the new reject length. This value is written in code C4135/000 bit 16. This value is displayed in code C4136/000 bit 16.
		TRUE Apply new reject length calculated.
		FALSE Do not apply new reject length calculated, continue to work with the last reject length calculated.
g_bWorkWithNewTestLength	BOOL	Activate the variable to immediately apply a new test length you have calculated and work with the new test length. This value is written in code C4135/000 bit 17. This value is displayed in code C4136/000 bit 17.
		TRUE Apply new test length calculated.
		FALSE Do not apply new test length calculated, continue to work with the last test length calculated.

Cross Cutter

Features of the "Cross Cutter" prepared solution

Inputs		
Global variables		(Variable type: VAR_GLOBAL)
Name	Data type	Value/Meaning
g_bAbordCrossCutter	BOOL	Activate the variable to abort cross-cutter operation. This value is written in code C4135/000 bit 24. This value is displayed in code C4136/000 bit 24.
		TRUE Abort cross-cutter operation.
		FALSE Normal operation
g_nAdjustCuttingSpeed	INT	Percentage value which can be applied to adjust the speed in the cut. 100% = 16384 This value is displayed in code C3665/000; this variable can be written to this code.
g_dnCirumferenceSlave	DINT	Cutting circle circumference xCircumference of knife drum in 0.0001[m_units]: Example: The cutting circle diameter of the knife drum is 127.3240[mm]. The circumference is calculated by multiplying the diameter by π (= 3.141592654 ...) and results in a value of 400.0000[m_units]. You should therefore assign a value of 4000000 to input variable dnCirumferenceSlave. Note: Do not enter the synchronous cutting length here (this may not be the same as the cutting circle circumference on knife drums with multiple knives). Always enter the full circumference of the knife cutting circle here. This value is displayed in code C3653/000; this variable can be written to this code.
g_dnSetCuttingLength	DINT	Cutting length for synchronous and asynchronous operation. The value is entered in 0.0001[m_units]. The variable is a retain variable. This ensures that once set, a cutting length will be retained (even if the mains power is disconnected). This value is displayed in code C3655/000; this variable can be written to this code.
g_dnSetRejectLength	DINT	Reject length for synchronous operation. The value is entered in 0.0001[m_units]. This value is displayed in code C3656/000; this variable can be written to this code.
g_dnSetTestLength	DINT	Test length for synchronous operation. The value is entered in 0.0001[m_units]. This value is displayed in code C3657/000; this variable can be written to this code.
g_dnSetStartStopLength	DINT	Start/stop length for synchronous operation. The value is displayed in 0.0001[m_units]. The value is calculated automatically by the prepared solution.
g_dnSyncAngle	DINT	Cutting angle for which the knife has to run in synchronism with the material (ysync, the value is entered in 0.0001[s_units]). Example: A dnSyncAngle value = 300000 corresponds to a cutting angle of 30.0000[s_units] = 30.0000 [°]. This value is displayed in code C3654/000; this variable can be written to this code. The value is converted into increments internally. If you use this variable, to avoid inconsistencies, do not write variable g_dnSyncAngle_p!
g_dnSyncAngle_p		Cutting angle for which the knife has to run in synchronism with the material. The value is entered in [incr]. If you use this variable, to avoid inconsistencies, do not write variable g_dnSyncAngle!

Cross Cutter

Features of the "Cross Cutter" prepared solution

Inputs		
Global variables		(Variable type: VAR_GLOBAL)
Name	Data type	Value/Meaning
g_byProfileTypeOfCalc	BYTE	Type of motion for the asynchronous motion phase. This value is displayed in code C3691/000; this variable can be written to this code.
		0 Second order polynomial (linear speed ramps)
		1 Fifth order polynomial
		2 Sloping sine line
g_bResetCuttingProfileCounter	BOOL	Reset cutting counter for cutting operation. This value is written in code C4135/000 bit 11. This value is displayed in code C4136/000 bit 11.
		TRUE Reset counter.
		FALSE Counter will not be reset.
g_bResetRejectProfileCounter	BOOL	Reset cutting counter for reject operation. This value is written in code C4135/000 bit 12. This value is displayed in code C4136/000 bit 12.
		TRUE Reset counter.
		FALSE Counter will not be reset.
g_bResetTestProfileCounter	BOOL	Reset cutting counter for test operation. This value is written in code C4135/000 bit 13. This value is displayed in code C4136/000 bit 13.
		TRUE Reset counter.
		FALSE Counter will not be reset.

Cross Cutter

Features of the "Cross Cutter" prepared solution

Outputs		
Global variables (Variable type: VAR_GLOBAL)		
Name	Data type	Value/Meaning
g_nNumberOfKnives	INT	Number of knives mounted on the knife drum. This value is displayed in code C3690/000.
g_bStartStopOperationCuttingProfile	BOOL	Status signal: Cutting length induces start/stop operation of cross cutter. This value is displayed in code C3700/000.
		TRUE The selected cutting length exceeds the limit cutting length for start/stop operation => The pause will be extended accordingly in the asynchronous motion phase.
		FALSE The selected cutting length undershoots the limit cutting length for start/stop operation. (See Chapter 2.2.2 Synchronous cross cutter.)
g_bStartStopOperationRejectProfile	BOOL	Status signal: Reject length induces start/stop operation of cross cutter. This value is displayed in code C3701/000.
		TRUE The selected reject length exceeds the limit cutting length for start/stop operation => The pause will be extended accordingly in the asynchronous motion phase.
		FALSE The selected reject length undershoots the limit cutting length for start/stop operation. (See Chapter 2.2.2 Synchronous cross cutter.)
g_bStartStopOperationTestProfile	BOOL	Status signal: Test length induces start/stop operation of cross cutter. This value is displayed in code C3702/000.
		TRUE The selected test length exceeds the limit cutting length for start/stop operation => The pause will be extended accordingly in the asynchronous motion phase.
		FALSE The selected test length undershoots the limit cutting length for start/stop operation. (See Chapter 2.2.2 Synchronous cross cutter.)
g_bStartStopOperationStartStopProfile	BOOL	Status signal: Start/stop length induces start/stop operation of cross cutter. This value is displayed in code C3703/000.
		TRUE The selected start/stop length exceeds the limit cutting length for start/stop operation => The pause will be extended accordingly in the asynchronous motion phase.
		FALSE The selected start/stop length undershoots the limit cutting length for start/stop operation. (See Chapter 2.2.2 Synchronous cross cutter.)

Cross Cutter

Features of the "Cross Cutter" prepared solution

Outputs		
Global variables		(Variable type: VAR_GLOBAL)
Name	Data type	Value/Meaning
g_bCrossCutterCalcBusyCuttingProfile	BOOL	Status signal: Calculation sequence for new cross-cutter motion profile for a new cutting length in progress. This value is displayed in code C4150/000 bit 8.
		TRUE Calculation sequence for cross-cutter motion profile for a new cutting length in progress.
		FALSE Calculation sequence for new cross-cutter motion profile for a new cutting length completed successfully or aborted.
g_bCrossCutterCalcBusyRejectProfile	BOOL	Status signal: Calculation sequence for new cross-cutter motion profile for a new reject length in progress. This value is displayed in code C4150/000 bit 9.
		TRUE Calculation sequence for cross-cutter motion profile for a new reject length in progress.
		FALSE Calculation sequence for new cross-cutter motion profile for a new reject length completed successfully or aborted.
g_bCrossCutterCalcBusyTestProfile	BOOL	Status signal: Calculation sequence for new cross-cutter motion profile for a new test length in progress. This value is displayed in code C4150/000 bit 10.
		TRUE Calculation sequence for cross-cutter motion profile for a new test length in progress.
		FALSE Calculation sequence for new cross-cutter motion profile for a new test length completed successfully or aborted.
g_bCrossCutterCalcBusyStartStopProfile	BOOL	Status signal: Calculation sequence for new cross-cutter motion profile for a new start/stop length in progress. This value is displayed in code C4150/000 bit 11.
		TRUE Calculation sequence for cross-cutter motion profile for a new start/stop length in progress.
		FALSE Calculation sequence for new cross-cutter motion profile for a new start/stop length completed successfully or aborted.
g_bCrossCutterCalcDoneCuttingProfile	BOOL	Status signal: Calculation sequence for new cross-cutter motion profile for a new cutting length completed successfully. This value is displayed in code C4150/000 bit 12.
		TRUE Calculation sequence for new cross-cutter motion profile for a new cutting length completed successfully. Starting a new calculation sequence will reset this status output to FALSE.
		FALSE Calculation sequence for new cross-cutter motion profile for a new cutting length in progress or an error has occurred in the calculation.

Cross Cutter

Features of the "Cross Cutter" prepared solution

Outputs		
Global variables (Variable type: VAR_GLOBAL)		
Name	Data type	Value/Meaning
g_bCrossCutterCalcDoneRejectProfile	BOOL	Status signal: Calculation sequence for new cross-cutter motion profile for a new reject length completed successfully. This value is displayed in code C4150/000 bit 13.
		TRUE Calculation sequence for new cross-cutter motion profile for a new reject length completed successfully. Starting a new calculation sequence will reset this status output to FALSE.
		FALSE Calculation sequence for new cross-cutter motion profile for a new reject length in progress or an error has occurred in the calculation.
g_bCrossCutterCalcDoneTestProfile	BOOL	Status signal: Calculation sequence for new cross-cutter motion profile for a new test length completed successfully. This value is displayed in code C4150/000 bit 14.
		TRUE Calculation sequence for new cross-cutter motion profile for a new test length completed successfully. Starting a new calculation sequence will reset this status output to FALSE.
		FALSE Calculation sequence for new cross-cutter motion profile for a new test length in progress or an error has occurred in the calculation.
g_bCrossCutterCalcDoneStartStopProfile	BOOL	Status signal: Calculation sequence for new cross-cutter motion profile for a new start/stop length completed successfully. This value is displayed in code C4150/000 bit 15.
		TRUE Calculation sequence for new cross-cutter motion profile for a new start/stop length completed successfully. Starting a new calculation sequence will reset this status output to FALSE.
		FALSE Calculation sequence for new cross-cutter motion profile for a new start/stop cutting length in progress or an error has occurred in the calculation.

Cross Cutter

Features of the "Cross Cutter" prepared solution

Outputs		
Global variables		(Variable type: VAR_GLOBAL)
Name	Data type	Value/Meaning
g_nCrossCutterCalcStateCuttingProfile	INT	Status of calculation sequence for new cutting length. This value is displayed in code C3661/000.
		0 Calculation ok.
		1 - No LC9 file found. - Profile data not found in absolute data model. - Read error when accessing profile data.
		2 Selected motion profile not found.
		4 The angle of motion of the knife drum for the asynchronous motion phase is less than or equal to zero (check y final value of basic profile and cutting angle!).
		8 A value of zero has been entered for the knife drum's cutting circle circumference.
		16 The preset cutting length is less than the material length corresponding to the cutting angle.
		32 The selected motion profile contains fewer than 20 interpolation points.
		64 The selected motion profile does not start at zero (0.0000/0.0000).
		128 Cutting angle too small.
		256 The y final value of the basic profile does not correspond to value $360.0000[^\circ]/n_{\text{cuts/rev}}$. ($n_{\text{cuts/rev}}$ = Number of knives on knife drum, values of 1, 2, 3, 4, 5 or 6 are possible)
		512 Internal read access to code C0011 failed.
		If the <i>nState</i> status variable is exhibiting non-zero values, the calculated variables may exhibit invalid values, in which case it will not be possible to use them subsequently in the control program!

Cross Cutter

Features of the "Cross Cutter" prepared solution

Outputs		
Global variables		(Variable type: VAR_GLOBAL)
Name	Data type	Value/Meaning
g_nCrossCutterCalcStateRejectProfile	INT	Status of calculation sequence for new reject length. This value is displayed in code C3662/000.
		0 Calculation ok.
		1 - No LC9 file found. - Profile data not found in absolute data model. - Read error when accessing profile data.
		2 Selected motion profile not found.
		4 The angle of motion of the knife drum for the asynchronous motion phase is less than or equal to zero (check y final value of basic profile and cutting angle!).
		8 A value of zero has been entered for the knife drum's cutting circle circumference.
		16 The preset cutting length is less than the material length corresponding to the cutting angle.
		32 The selected motion profile contains fewer than 20 interpolation points.
		64 The selected motion profile does not start at zero (0.0000/0.0000).
		128 Cutting angle too small.
		256 The y final value of the basic profile does not correspond to value $360.0000[^\circ]/n_{\text{cuts/rev}}$. ($n_{\text{cuts/rev}}$ = Number of knives on knife drum, values of 1, 2, 3, 4, 5 or 6 are possible)
		512 Internal read access to code C0011 failed.
		If the nState status variable is exhibiting non-zero values, the calculated variables may exhibit invalid values, in which case it will not be possible to use them subsequently in the control program!

Cross Cutter

Features of the "Cross Cutter" prepared solution

Outputs		
Global variables (Variable type: VAR_GLOBAL)		
Name	Data type	Value/Meaning
g_nCrossCutterCalcStateTestProfile	INT	Status of calculation sequence for new test length. This value is displayed in code C3663/000.
		0 Calculation ok.
		1 - No LC9 file found. - Profile data not found in absolute data model. - Read error when accessing profile data.
		2 Selected motion profile not found.
		4 The angle of motion of the knife drum for the asynchronous motion phase is less than or equal to zero (check y final value of basic profile and cutting angle!).
		8 A value of zero has been entered for the knife drum's cutting circle circumference.
		16 The preset cutting length is less than the material length corresponding to the cutting angle.
		32 The selected motion profile contains fewer than 20 interpolation points.
		64 The selected motion profile does not start at zero (0.0000/0.0000).
		128 Cutting angle too small.
		256 The y final value of the basic profile does not correspond to value $360.0000[^\circ]/n_{\text{cuts/rev}}$. ($n_{\text{cuts/rev}}$ = Number of knives on knife drum, values of 1, 2, 3, 4, 5 or 6 are possible)
		512 Internal read access to code C0011 failed.
		If the nState status variable is exhibiting non-zero values, the calculated variables may exhibit invalid values, in which case it will not be possible to use them subsequently in the control program!

Cross Cutter

Features of the "Cross Cutter" prepared solution

Outputs		
Global variables (Variable type: VAR_GLOBAL)		
Name	Data type	Value/Meaning
g_nCrossCutterCalcStateStartStopProfile	INT	Status of calculation sequence for new start/stop length. This value is displayed in code C3664/000.
		0 Calculation ok.
		1 - No LC9 file found. - Profile data not found in absolute data model. - Read error when accessing profile data.
		2 Selected motion profile not found.
		4 The angle of motion of the knife drum for the asynchronous motion phase is less than or equal to zero (check y final value of basic profile and cutting angle!).
		8 A value of zero has been entered for the knife drum's cutting circle circumference.
		16 The preset cutting length is less than the material length corresponding to the cutting angle.
		32 The selected motion profile contains fewer than 20 interpolation points.
		64 The selected motion profile does not start at zero (0.0000/0.0000).
		128 Cutting angle too small.
		256 The y final value of the basic profile does not correspond to value $360.0000[^\circ]/n_{\text{cuts/rev}}$. ($n_{\text{cuts/rev}}$ = Number of knives on knife drum, values of 1, 2, 3, 4, 5 or 6 are possible)
		512 Internal read access to code C0011 failed.
		If the nState status variable is exhibiting non-zero values, the calculated variables may exhibit invalid values, in which case it will not be possible to use them subsequently in the control program!
g_bAllCalcDone	BOOL	Status signal: All calculation sequences for new cross-cutter motion profiles have been completed successfully. This value is displayed in code C4150/000 bit 31.
		TRUE All calculation sequences for new cross-cutter motion profiles have been completed successfully. Starting a new calculation sequence will reset this status output to FALSE.
		FALSE All calculation sequences for new cross-cutter motion profiles are still in progress.
g_bWrongSetCamProfileNoForCalc	BOOL	Status signal: A motion profile currently being used by the application has been calculated.
		TRUE A motion profile currently being used by the application has been calculated.
		FALSE The calculated motion profile is not currently being used by the application.
g_bErrorCrossCutterCuttingProfileCalcBlock	BOOL	Status signal: An error occurred whilst calculating the motion profile for the new cutting length.
		TRUE An error has occurred during calculation.
		FALSE Calculation completed without errors.

Cross Cutter

Features of the "Cross Cutter" prepared solution

Outputs		
Global variables		(Variable type: VAR_GLOBAL)
Name	Data type	Value/Meaning
g_bErrorCrossCutterRejectProfileCalcBlock	BOOL	Status signal: An error occurred whilst calculating the motion profile for the new reject length.
		TRUE An error has occurred during calculation.
		FALSE Calculation completed without errors.
g_bErrorCrossCutterTestProfileCalcBlock	BOOL	Status signal: An error occurred whilst calculating the motion profile for the new test length.
		TRUE An error has occurred during calculation.
		FALSE Calculation completed without errors.
g_bErrorCrossCutterStartStopProfileCalcBlock	BOOL	Status signal: An error occurred whilst calculating the motion profile for the new start/stop length.
		TRUE An error has occurred during calculation.
		FALSE Calculation completed without errors.
g_bUseSynchronusCrossCutter	BOOL	Status signal: The cross cutter is operating as a synchronous cross cutter. This value is displayed in code C4150/000 bit 21.
		TRUE The cross cutter is operating as a synchronous cross cutter.
		FALSE The cross cutter is not operating as a synchronous cross cutter.
g_bUseAsynchronusCrossCutter	BOOL	Status signal: The cross cutter is operating as an asynchronous cross cutter. This value is displayed in code C4150/000 bit 20.
		TRUE The cross cutter is operating as an asynchronous cross cutter.
		FALSE The cross cutter is not operating as an asynchronous cross cutter.
g_bUseCuttingProfile	BOOL	Status signal: The cross cutter is operating in "Continuous cutting" mode. This value is displayed in code C4150/000 bit 22.
		TRUE The cross cutter is operating in "Continuous cutting" mode.
		FALSE The cross cutter is not operating in "Continuous cutting" mode.
g_bUseRejectProfile	BOOL	Status signal: The cross cutter is operating in "Reject operation" mode. This value is displayed in code C4150/000 bit 23.
		TRUE The cross cutter is operating in "Reject operation" mode.
		FALSE The cross cutter is not operating in "Reject operation" mode.
g_bUseTestProfile	BOOL	Status signal: The cross cutter is operating in "Test operation" mode. This value is displayed in code C4150/000 bit 24.
		TRUE The cross cutter is operating in "Test operation" mode.
		FALSE The cross cutter is not operating in "Test operation" mode.
g_nMaxMasterVelForActCuttingProfile_v	INT	Entry indicating the maximum possible line speed for the set cutting length in [incr./ms].
g_nMaxMasterVelForActRejectProfile_v	INT	Entry indicating the maximum possible line speed for the set reject length in [incr./ms].
g_nMaxMasterVelForActTestProfile_v	INT	Entry indicating the maximum possible line speed for the set test length in [incr./ms].
g_nMaxMasterVelForActStartStopProfile_v	INT	Entry indicating the maximum possible line speed for the set start/stop length in [incr./ms].
g_wCounterValueCuttingProfile	WORD	Cutting counter for cutting operation The counter is limited to a maximum value of 65535; on reaching this value it will overflow to 0. This value is displayed in code C3658/000.
g_wCounterValueRejectProfile	WORD	Cutting counter for reject operation The counter is limited to a maximum value of 65535; on reaching this value it will overflow to 0. This value is displayed in code C3659/000.

Cross Cutter

Features of the "Cross Cutter" prepared solution

Outputs		
Global variables		(Variable type: VAR_GLOBAL)
g_wCounterValueTestProfile	WORD	Cutting counter for test operation The counter is limited to a maximum value of 65535; on reaching this value it will overflow to 0. This value is displayed in code C3660/000.

Code	Possible settings:		Comment
	Default	Selection	
C3653/000	628.3185	0.0000... {0.0001 m_units} ... 214748.0000	Cutting circle circumference of knife drum
C3654/000	30.0000	0.0000... {°} ... 360.0000	Cutting angle (synchronous angle)
C3655/000	300.0000	0.0001... {{0.0001 m_units} ... 214000.0000	Setpoint cutting length (valid for synchronous and asynchronous cross-cutter operation)
C3656/000	150.0000	0.0001... {{0.0001 m_units} ... 214000.0000	Setpoint for reject length
C3657/000	2400.0000	0.0001... {{0.0001 m_units} ... 214000.0000	Setpoint for test length
C3658/000	-	-	Cutting counter for cutting operation The counter is limited to a maximum value of 65535; on reaching this value it will overflow to 0.
C3659/000	-	-	Cutting counter for reject operation The counter is limited to a maximum value of 65535; on reaching this value it will overflow to 0.
C3660/000	-	-	Cutting counter for test operation The counter is limited to a maximum value of 65535; on reaching this value it will overflow to 0.
C3661/000	-	-	Status of calculation sequence for new cutting length.
			0 Calculation ok.
			-1 - No LC9 file found. - Profile data not found in absolute data model. - Read error when accessing profile data.
			-2 Selected motion profile not found.
			-4 The angle of motion of the knife drum for the asynchronous motion phase is less than or equal to zero (check y final value of basic profile and cutting angle!).
			-8 A value of zero has been entered for the knife drum's cutting circle circumference.
			-16 The preset cutting length is less than the material length corresponding to the cutting angle.
			-32 The selected motion profile contains fewer than 20 interpolation points.
			-64 The selected motion profile does not start at zero (0.0000/0.0000).
			-128 Cutting angle too small.
			-256 The y final value of the basic profile does not correspond to value $360.0000[°]/n_{cuts/rev}$. ($n_{cuts/rev}$ = Number of knives on knife drum, values of 1, 2, 3, 4, 5 or 6 are possible)
			If the nState status variable is exhibiting negative values, the output signals may exhibit invalid values, in which case it will not be possible to use them subsequently in the control program!

Cross Cutter

Features of the "Cross Cutter" prepared solution

Code	Possible settings:		Comment
	Default	Selection	
C3662/ 000	-	-	Status of calculation sequence for new reject length.
			0 Calculation ok.
			-1 - No LC9 file found. - Profile data not found in absolute data model. - Read error when accessing profile data.
			-2 Selected motion profile not found.
			-4 The angle of motion of the knife drum for the asynchronous motion phase is less than or equal to zero (check y final value of basic profile and cutting angle!).
			-8 A value of zero has been entered for the knife drum's cutting circle circumference.
			-16 The preset cutting length is less than the material length corresponding to the cutting angle.
			-32 The selected motion profile contains fewer than 20 interpolation points.
			-64 The selected motion profile does not start at zero (0.0000/0.0000).
			-128 Cutting angle too small.
			-256 The y final value of the basic profile does not correspond to value $360.0000[^\circ]/n_{\text{cuts/rev}}$. ($n_{\text{cuts/rev}}$ = Number of knives on knife drum, values of 1, 2, 3, 4, 5 or 6 are possible)
			If the $nState$ status variable is exhibiting negative values, the output signals may exhibit invalid values, in which case it will not be possible to use them subsequently in the control program!
C3663/ 000	-	-	Status of calculation sequence for new test length.
			0 Calculation ok.
			-1 - No LC9 file found. - Profile data not found in absolute data model. - Read error when accessing profile data.
			-2 Selected motion profile not found.
			-4 The angle of motion of the knife drum for the asynchronous motion phase is less than or equal to zero (check y final value of basic profile and cutting angle!).
			-8 A value of zero has been entered for the knife drum's cutting circle circumference.
			-16 The preset cutting length is less than the material length corresponding to the cutting angle.
			-32 The selected motion profile contains fewer than 20 interpolation points.
			-64 The selected motion profile does not start at zero (0.0000/0.0000).
			-128 Cutting angle too small.
			-256 The y final value of the basic profile does not correspond to value $360.0000[^\circ]/n_{\text{cuts/rev}}$. ($n_{\text{cuts/rev}}$ = Number of knives on knife drum, values of 1, 2, 3, 4, 5 or 6 are possible)
			If the $nState$ status variable is exhibiting negative values, the output signals may exhibit invalid values, in which case it will not be possible to use them subsequently in the control program!

Cross Cutter

Features of the "Cross Cutter" prepared solution

Code	Possible settings:		Comment
	Default	Selection	
C3664/ 000	-	-	- Status of calculation sequence for new start/stop length.
			0 Calculation ok.
			-1 - No LC9 file found. - Profile data not found in absolute data model. - Read error when accessing profile data.
			-2 Selected motion profile not found.
			-4 The angle of motion of the knife drum for the asynchronous motion phase is less than or equal to zero (check y final value of basic profile and cutting angle!).
			-8 A value of zero has been entered for the knife drum's cutting circle circumference.
			-16 The preset cutting length is less than the material length corresponding to the cutting angle.
			-32 The selected motion profile contains fewer than 20 interpolation points.
			-64 The selected motion profile does not start at zero (0.0000/0.0000).
			-128 Cutting angle too small.
			-256 The y final value of the basic profile does not correspond to value $360.0000[^\circ]/n_{\text{cuts/rev}}$. ($n_{\text{cuts/rev}}$ = Number of knives on knife drum, values of 1, 2, 3, 4, 5 or 6 are possible)
			If the $nState$ status variable is exhibiting negative values, the output signals may exhibit invalid values, in which case it will not be possible to use them subsequently in the control program!
C3665/ 000	100	80... { } ... 120	Cutting speed adaptation factor.
C3686/ 000	-	- {m_units/ min}	- Maximum possible line speed for set cutting length.
C36867/ 000	-	- {m_units/ min}	- Maximum possible line speed for set reject length.
C3688/ 000	-	- {m_units/ min}	- Maximum possible line speed for set test length.
C3689/ 000	-	- {m_units/ min}	- Maximum possible line speed for set start/stop length.
C3690/ 000	-	- {1}	- Number of knives mounted on the knife drum. Up to 6 knives can be mounted on the knife drum.
C3691/ 000	1	0 {1}	2 Selection of calculation method for asynchronous section of motion profile
			0 Second order polynomial (linear speed ramps)
			1 Fifth order polynomial
			2 Sloping sine line
C3700/ 000	-	- {1}	- Cutting length induces start/stop operation of cross cutter.
			TRUE The selected cutting length exceeds the limit cutting length for start/stop operation => The pause will be extended accordingly in the asynchronous motion phase.
			FALSE The selected cutting length undershoots the limit cutting length for start/stop operation.

Cross Cutter

Features of the "Cross Cutter" prepared solution

Code	Possible settings:		Comment
	Default	Selection	
C3701/ 000	-	{1}	- Reject length induces start/stop operation of cross cutter.
			TRUE The selected reject length exceeds the limit cutting length for start/stop operation => The pause will be extended accordingly in the asynchronous motion phase.
			FALSE The selected reject length undershoots the limit cutting length for start/stop operation.
C3702/ 000	-	{1}	- Test length induces start/stop operation of cross cutter.
			TRUE The selected test length exceeds the limit cutting length for start/stop operation => The pause will be extended accordingly in the asynchronous motion phase.
			FALSE The selected test length undershoots the limit cutting length for start/stop operation.
C3703/ 000	-	{1}	- Start/stop length induces start/stop operation of cross cutter.
			TRUE The selected start/stop length exceeds the limit cutting length for start/stop operation => The pause will be extended accordingly in the asynchronous motion phase.
			FALSE The selected start/stop length undershoots the limit cutting length for start/stop operation.
C4135/ 000 Bit 11	0	{1}	1 Application control word bit 11: Reset cutting counter for cutting operation.
			TRUE Reset counter.
			FALSE Counter will not be reset.
C4135/ 000 Bit 12	0	{1}	1 Application control word bit 12: Reset cutting counter for reject operation.
			TRUE Reset counter.
			FALSE Counter will not be reset.
C4135/ 000 Bit 13	0	{1}	1 Application control word bit 13: Reset cutting counter for test operation.
			TRUE Reset counter.
			FALSE Counter will not be reset.
C4135/ 000 Bit 14	0	{1}	1 Application control word bit 14: Start and stop selected cross-cutter operating mode.
			TRUE Start operation.
			FALSE Stop operation.
C4135/ 000 Bit 15	0	{1}	1 Application control word bit 15: Activate to immediately apply a new cutting length you have calculated and work with the new cutting length.
			TRUE Apply new cutting length calculated.
			FALSE Do not apply new cutting length calculated, continue to work with the last cutting length calculated.

Cross Cutter

Features of the "Cross Cutter" prepared solution

Code	Possible settings:		Comment
	Default	Selection	
C4135/ 000 Bit 16	0	0 {1}	1 Application control word bit 16: Activate to immediately apply a new reject length you have calculated and work with the new cutting length.
			TRUE Apply new reject length calculated.
			FALSE Do not apply new reject length calculated, continue to work with the last cutting length calculated.
C4135/ 000 Bit 17	0	0 {1}	1 Application control word bit 17: Activate to immediately apply a new test length you have calculated and work with the new cutting length.
			TRUE Apply new test length calculated.
			FALSE Do not apply new test length calculated, continue to work with the last cutting length calculated.
C4137/ 000	1	1 {1}	4 Select cross-cutter operating mode. Note: You can switch between modes 1 to 3 during operation. To switch to mode 4 from one of modes 1 to 3, you will first need to stop the cross cutter via g_bStartStopOperation. To switch from mode 4 to one of modes 1 to 3, you will first need to stop the cross cutter via g_bStartStopOperation.
			1 Synchronous cross cutter, continuous cutting
			2 Synchronous cross cutter, reject operation
			3 Synchronous cross cutter, test operation
			4 Asynchronous cross cutter, continuous cutting
C4150/ 000 Bit 8	-	-	- Application status word bit 8: Calculation sequence for new cross-cutter motion profile for a new cutting length in progress.
			TRUE Calculation sequence for cross-cutter motion profile for a new cutting length in progress.
			FALSE Calculation sequence for new cross-cutter motion profile for a new cutting length completed successfully or aborted.
C4150/ 000 Bit 9	-	-	- Application status word bit 9: Calculation sequence for new cross-cutter motion profile for a new reject length in progress.
			TRUE Calculation sequence for cross-cutter motion profile for a new reject length in progress.
			FALSE Calculation sequence for new cross-cutter motion profile for a new reject length completed successfully or aborted.
C4150/ 000 Bit 10	-	-	- Application status word bit 10: Calculation sequence for new cross-cutter motion profile for a new test length in progress.
			TRUE Calculation sequence for cross-cutter motion profile for a new test length in progress.
			FALSE Calculation sequence for new cross-cutter motion profile for a new test length completed successfully or aborted.

Cross Cutter

Features of the "Cross Cutter" prepared solution

Code	Possible settings:		Comment
	Default	Selection	
C4150/ 000 Bit 11	-	-	- Application status word bit 11: Calculation sequence for new cross-cutter motion profile for a new start/stop length in progress.
			TRUE Calculation sequence for cross-cutter motion profile for a new start/stop length in progress.
			FALSE Calculation sequence for new cross-cutter motion profile for a new start/stop length completed successfully or aborted.
C4150/ 000 Bit 12	-	-	- Application status word bit 12: Calculation sequence for new cross-cutter motion profile for a new cutting length completed successfully. Starting a new calculation sequence will reset this bit to FALSE.
			TRUE Calculation sequence for new cross-cutter motion profile for a new cutting length completed successfully.
			FALSE Calculation sequence for new cross-cutter motion profile for a new cutting length not completed.
C4150/ 000 Bit 13	-	-	- Application status word bit 13: Calculation sequence for new cross-cutter motion profile for a new reject length completed successfully. Starting a new calculation sequence will reset this bit to FALSE.
			TRUE Calculation sequence for new cross-cutter motion profile for a new reject length completed successfully.
			FALSE Calculation sequence for new cross-cutter motion profile for a new reject length not completed.
C4150/ 000 Bit 14	-	-	- Application status word bit 14: Calculation sequence for new cross-cutter motion profile for a new test length completed successfully. Starting a new calculation sequence will reset this bit to FALSE.
			TRUE Calculation sequence for new cross-cutter motion profile for a new test length completed successfully.
			FALSE Calculation sequence for new cross-cutter motion profile for a new test length not completed.
C4150/ 000 Bit 15	-	-	- Application status word bit 15: Calculation sequence for new cross-cutter motion profile for a new start/stop length completed successfully. Starting a new calculation sequence will reset this bit to FALSE.
			TRUE Calculation sequence for new cross-cutter motion profile for a new start/stop length completed successfully.
			FALSE Calculation sequence for new cross-cutter motion profile for a new start/stop length not completed.

Cross Cutter

Features of the "Cross Cutter" prepared solution

Code	Possible settings:		Comment
	Default	Selection	
C4150/ 000 Bit 20	-	-	Application status word bit 20: The cross cutter is operating as an asynchronous cross cutter.
			TRUE The cross cutter is operating as an asynchronous cross cutter.
			FALSE The cross cutter is not operating as an asynchronous cross cutter.
C4150/ 000 Bit 21	-	-	Application status word bit 21: The cross cutter is operating as a synchronous cross cutter.
			TRUE The cross cutter is operating as a synchronous cross cutter.
			FALSE The cross cutter is not operating as a synchronous cross cutter.
C4150/ 000 Bit 22	-	-	Application status word bit 22: The cross cutter is operating in "Continuous cutting" mode.
			TRUE The cross cutter is operating in "Continuous cutting" mode.
			FALSE The cross cutter is not operating in "Continuous cutting" mode.
C4150/ 000 Bit 23	-	-	Application status word bit 23: The cross cutter is operating in "Reject operation" mode.
			TRUE The cross cutter is operating in "Reject operation" mode.
			FALSE The cross cutter is not operating in "Reject operation" mode.
C4150/ 000 Bit 24	-	-	Application status word bit 24: The cross cutter is operating in "Test operation" mode.
			TRUE The cross cutter is operating in "Test operation" mode.
			FALSE The cross cutter is not operating in "Test operation" mode.
C4150/ 000 Bit 31	-	-	Application status word 31: All calculation sequences for new cross-cutter motion profiles have been completed successfully. Starting a new calculation sequence will reset this status output to FALSE.
			TRUE All calculation sequences for new cross-cutter motion profiles have been completed successfully.
			FALSE All calculation sequences for new cross-cutter motion profiles are still in progress.

The warning signals listed in the following table may appear when calculating cross-cutter motion profiles. Each warning signal will remain pending until its cause has been dealt with.

Each warning signal is generated in the specified network of the *UserErrors* block (POU).

Warning signals			
Number	Text	Global variable	Network in UserErrors POU
501	WrongProfileForCalc	g_bWrongSetCamProfileNoForCalc	2
503	ErrorCalcCuttingProfile	g_bErrorCrossCutterCuttingProfileCalcBlock	4
504	ErrorCalcRejectProfile	g_bErrorCrossCutterRejectProfileCalcBlock	5
505	ErrorCalcTestProfile	g_bErrorCrossCutterTestProfileCalcBlock	6
506	ErrorCalcStartStopProfile	g_bErrorCrossCutterStartStopProfileCalcBlock	7

Cross Cutter

Features of the "Cross Cutter" prepared solution

2.5.4 ManualJog

In ManualJog the drive/tool can be moved by means of manual operator inputs. This type of operation may be required, for example, to clean or change a tool.

In this context it is important that when functioning in manual mode, the drive moves in "inching mode" but does not follow the cam (or motion profile). ManualJog movement is not determined by the motion profile (cam) but follows linear speed curves independently of the profile. In this operating mode, a master value entry and the cam function will have no effect on drive motion. In ManualJog, the *g_bGlobalError* global variable is evaluated. Setting the *g_bGlobalError* global variable to TRUE during ManualJog in a positive or negative direction will stop the drive on the quick stop ramp. By resetting the error (*g_bGlobalError*=FALSE) the drive direct start moving in the selected direction as long as there is a signal for manual movement. There is no new positiv edge necessary to restart the manual movement.

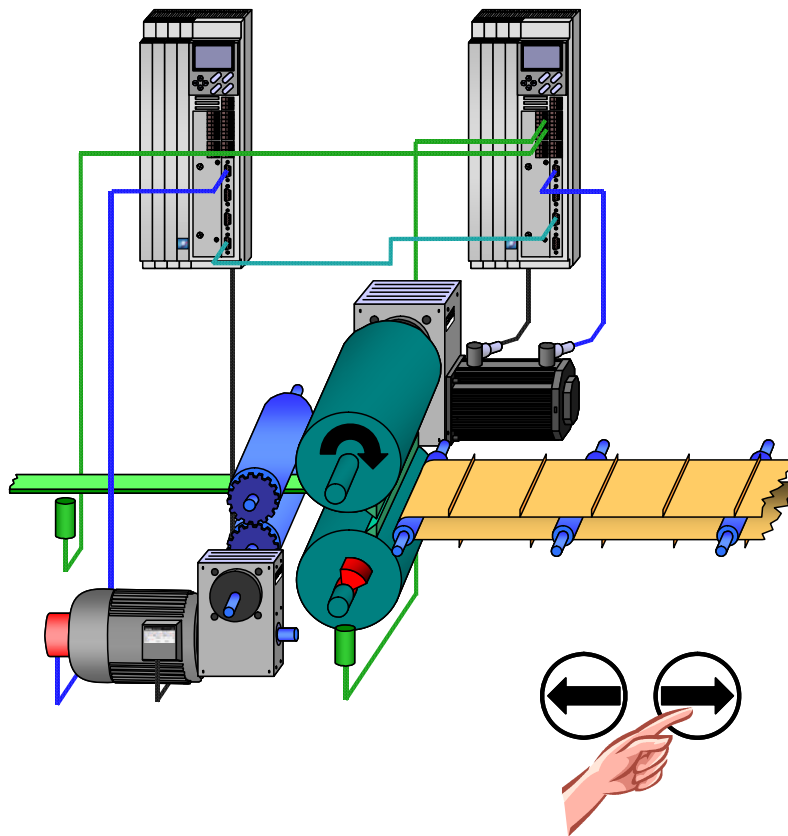


Figure 6: Schematic representation of manual mode

Cross Cutter

Features of the "Cross Cutter" prepared solution

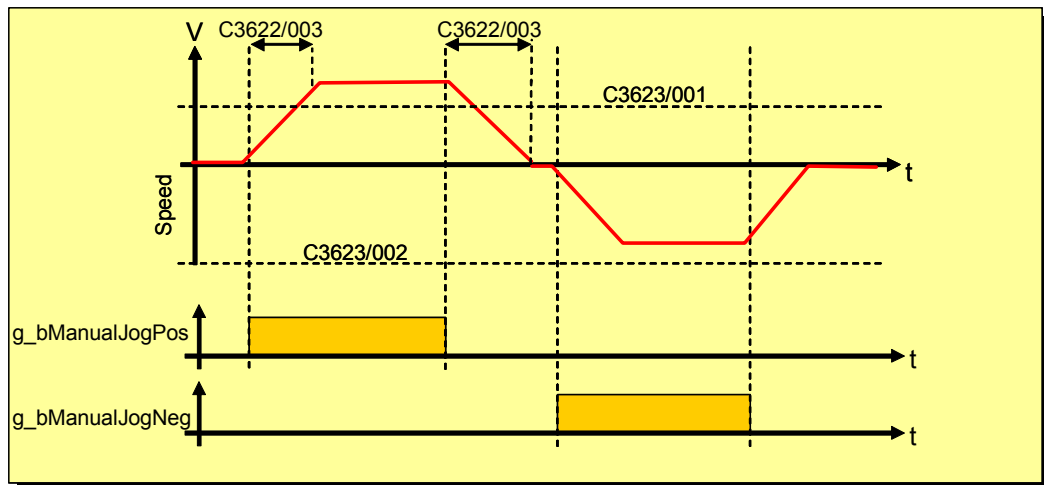


Figure 7: Schematic representation of signal generation for manual mode

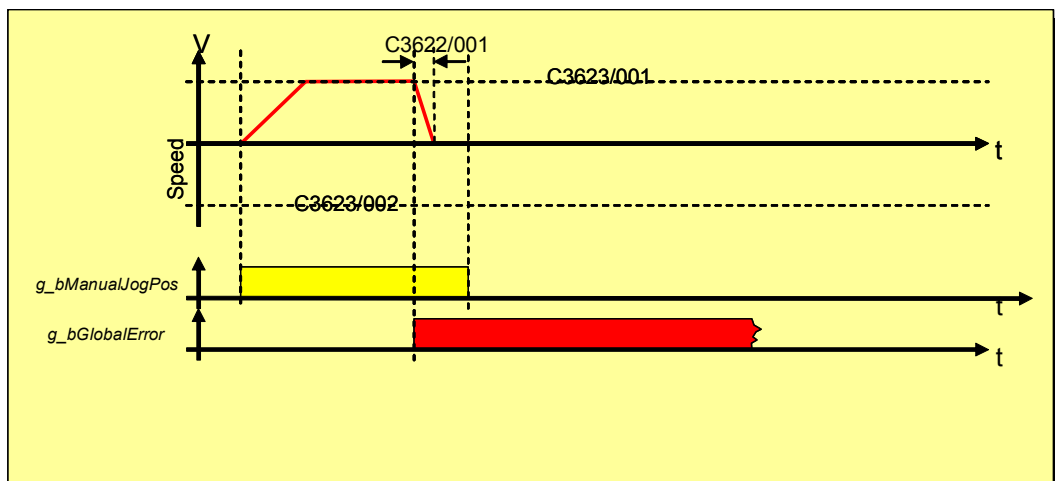


Figure 8: Schematic representation of signal generation for manual mode (error)

Cross Cutter

Features of the "Cross Cutter" prepared solution

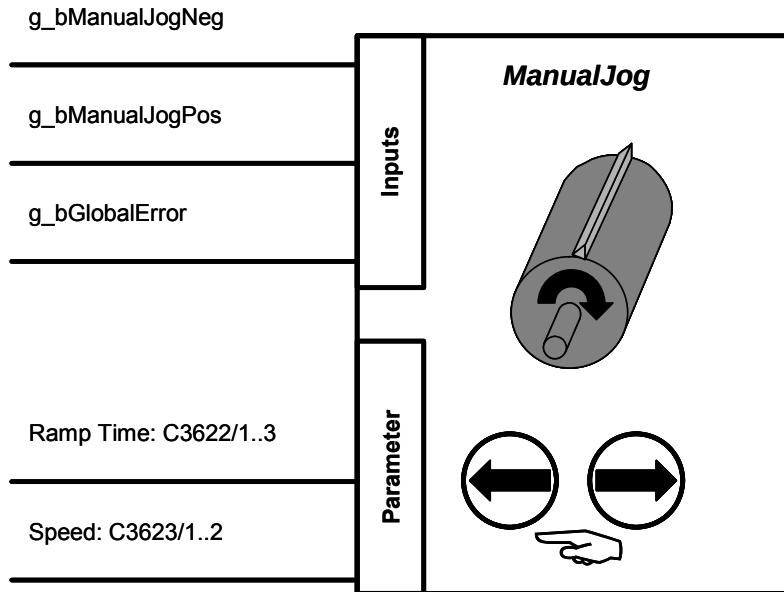


Figure 9: Control signals in ManualJog

Variable names

Global variables (Variable type: VAR_GLOBAL)		
Name	Data type	Value/Meaning
g_bManualJogNeg	BOOL	Manual jog in negative direction (CCW) This value is written in code C4135/000 bit 4. This value is displayed in code C4136/000 bit 4.
		TRUE Manual jog in negative direction
		FALSE Stop
g_bManualJogPos	BOOL	Manual jog in positive direction (CW) This value is written in code C4135/000 bit 3. This value is displayed in code C4136/000 bit 3.
		TRUE Manual jog in positive direction
		FALSE Stop

Code	Possible settings:				Comment
	Default	Selection			
C3622 /001	5.000	0.000... {0.0001[s]} ...129.99			Deceleration ramp for immediate stop Assuming 4000 rpm
C3622 /003	5.000	0.000... {0.0001[s]} ...129.99			Acceleration and deceleration ramp for manual jog Assuming 4000 rpm
C3623 /001	200	0 ... {1 rpm} ...15000			Speed for manual jog in positive direction (CW)
C3623 /002	200	0 ... {1 rpm} ...15000			Speed for manual jog in negative direction (CCW)
C4135 /000 Bit 3	0	0 {1}	1		Application control word bit 3: Manual jog in positive direction (CW)
				TRUE	Manual jog in positive direction
				FALSE	Stop
C4135 /000 Bit 4	0	0 {1}	1		Application control word bit 4: Manual jog in negative direction (CCW)
				TRUE	Manual jog in negative direction
				FALSE	Stop

Cross Cutter

Features of the "Cross Cutter" prepared solution

2.5.5 Following error monitoring

When the servo controller is enabled, the drive system will monitor the following error continuously. The following error Δs (green line) is the result of the difference between the set position s_{set} (blue line) and the actual position s_{act} (red line). If this deviation exceeds a defined limit value, a following error warning will be output. Following error warnings are defined as follows:

Following error warning limit (C3645)

If the difference between the set position and actual position exceeds the setting in C3645, a following error warning will be output.

Use C3646 to define the limit values for the following error warning. This is the value by which the following error warning limit must be undershot (C3645) in order for the following error warning

to be reset. The warning is reset automatically when the set value is undershot.

Following error limit (C3643)

If the difference between the set position and the actual position exceeds the setting in C3643, the drive will brake to standstill along the set quick stop ramp and switch to the "Trouble" state.

Use the corresponding motor control parameters (C0105) to set the quick stop ramp.

Use C3644 to define the limit values for the following error. This is the value by which the following error limit (C3643) must be undershot in order for the following error to be reset.

The user must reset the error. Following error monitoring is active whenever the drive is in motion.

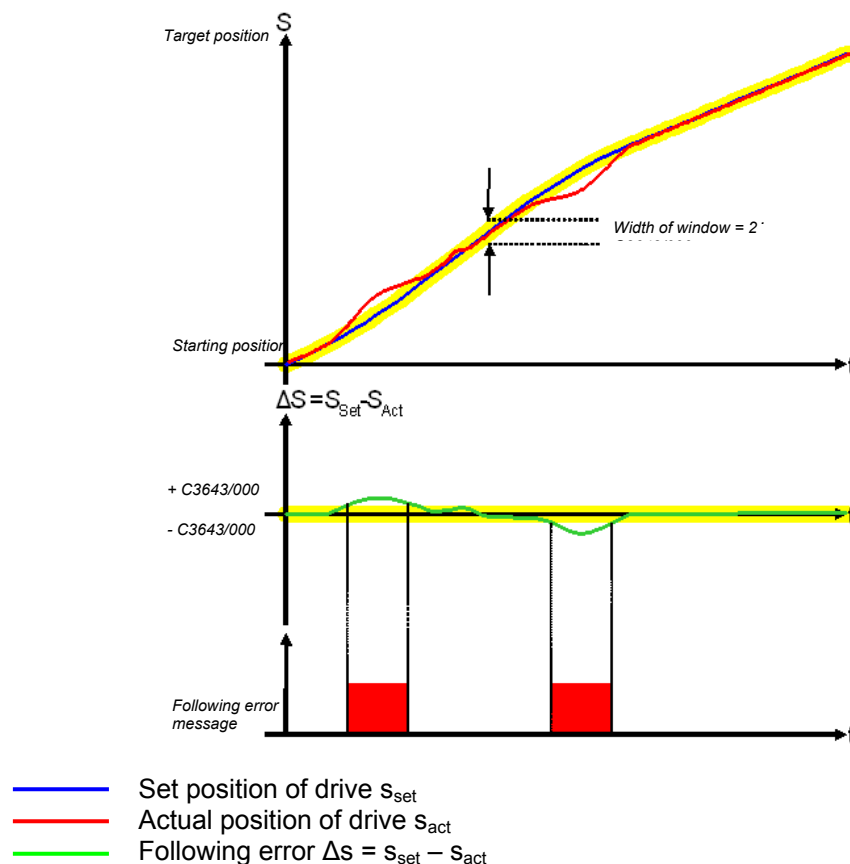


Figure 10: Generation of following error message

Cross Cutter

Features of the "Cross Cutter" prepared solution

Variable names

Code	Possible settings:		Comments
	Default	Selection	
C3643/ 000	3.0000	0.0001... {0.0001 ... 214748.0000 [s_units]}	Following error switch-off threshold
C3644/ 000	0.5000	0.0001... {0.0001 ... 214748.0000 [s_units]}	Following error hysteresis
C3645/ 000	1.5000	0.0001... {0.0001 ... 214748.0000 [s_units]}	Following error warning limit
C3646/ 000	0.5000	0.0001... {0.0001 ... 214748.0000 [s_units]}	Following error warning limit hysteresis

2.5.6 Homing

This function can be used to perform homing. The drive moves in a preselected mode and calculates the zero position (home) automatically on the basis of a home mark before declaring this value to the drive control. All position data refers to this home position. Homing is generally performed once following mains connection.

The following can be used as home mark signals:

- Zero pulse (signal edge from encoder system for position actual value, one per motor revolution)
- Touch probe (signal edge at corresponding touch-probe input of PLC)
- A preliminary stop position can be set via a homing switch. In this case, home will be the next zero position of the actual value encoder on the motor or the next touch probe (depending on the selected mode).

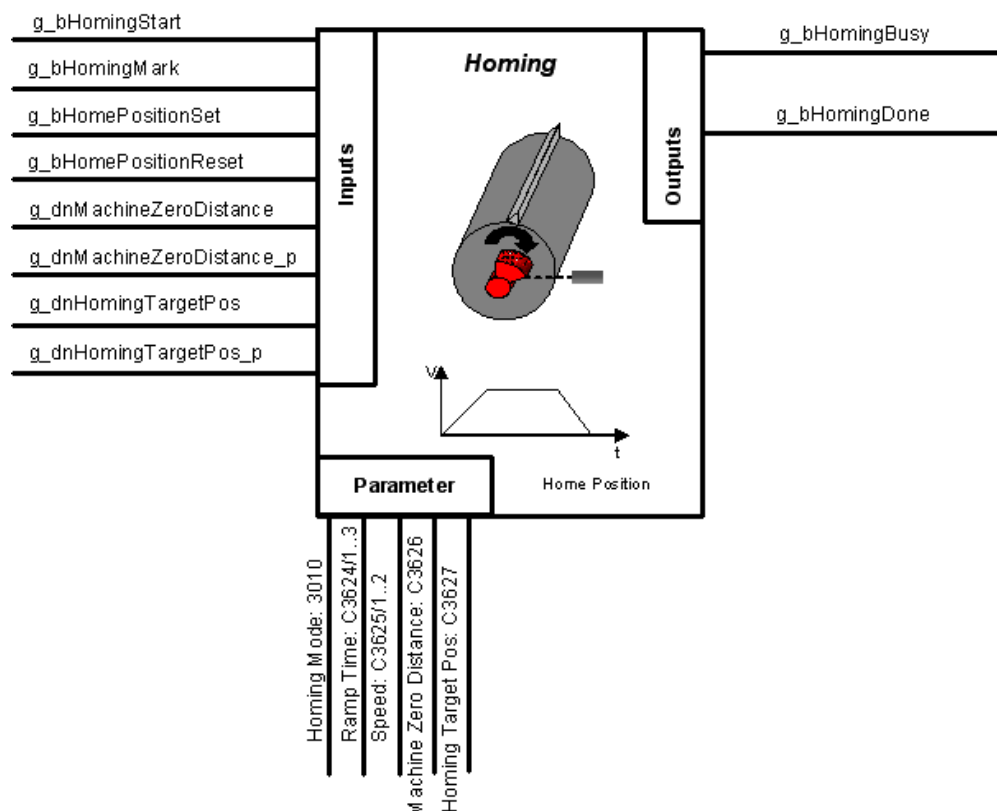


Figure 11: Control signals for homing

Cross Cutter

Features of the "Cross Cutter" prepared solution

The following homing modes can be selected:

- Homing mode 0:

The drive rotates in a positive direction (clockwise rotation) until the homing switch's negative edge is detected. The next zero pulse from the motor feedback system determines the home position (machine zero point).

If you do not wish the home position to be the zero point of the absolute measuring system used, you can use an offset value (machine zero distance, C3626/000) to set the home position to the zero point of the measuring system. Once homing is complete the drive will stop at position 0.000[s_units]:

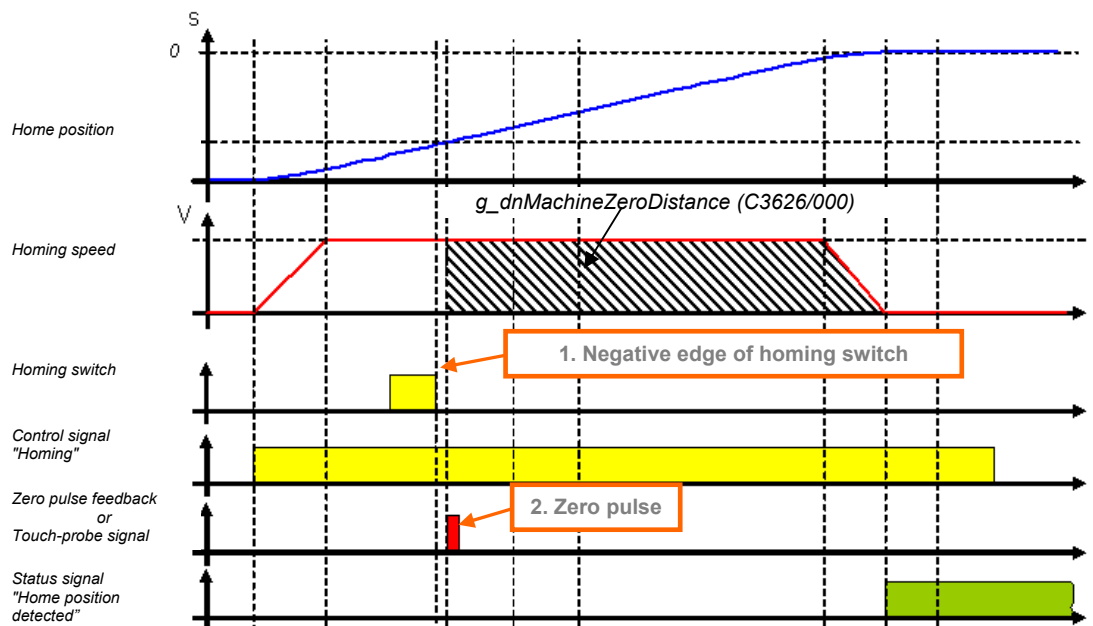


Figure 12: Graph of homing mode 0

In the example illustrated above, the home position has a negative value with reference to the measuring system zero point, because once the zero pulse/touch-probe signal has been detected, the drive continues to move in a positive direction to a position 0.0000[s_units].

If you do not wish homing to end at the measuring system zero point, you can in addition preset an end position to come after homing (homing target position, C3627/000). This provides the requisite conditions for setting the measuring system zero point outside the drive's traversing range.

- Homing mode 1:
(like homing mode 0 but with a negative traversing direction)

Cross Cutter

Features of the "Cross Cutter" prepared solution

- Homing mode 4:

The drive moves in a positive direction until the rising edge of the homing switch signal is detected, at which point it starts to reverse. The home position (machine zero point) is determined by the next zero pulse on the motor feedback system following the next falling signal edge or by the next touch-probe signal.

If you do not wish the home position to be the zero point of the absolute measuring system used, you can use an offset value (machine zero distance, C3626/000) to set the home position to the zero point of the measuring system. Once homing is complete the drive will stop at position 0.0000[s_units]:

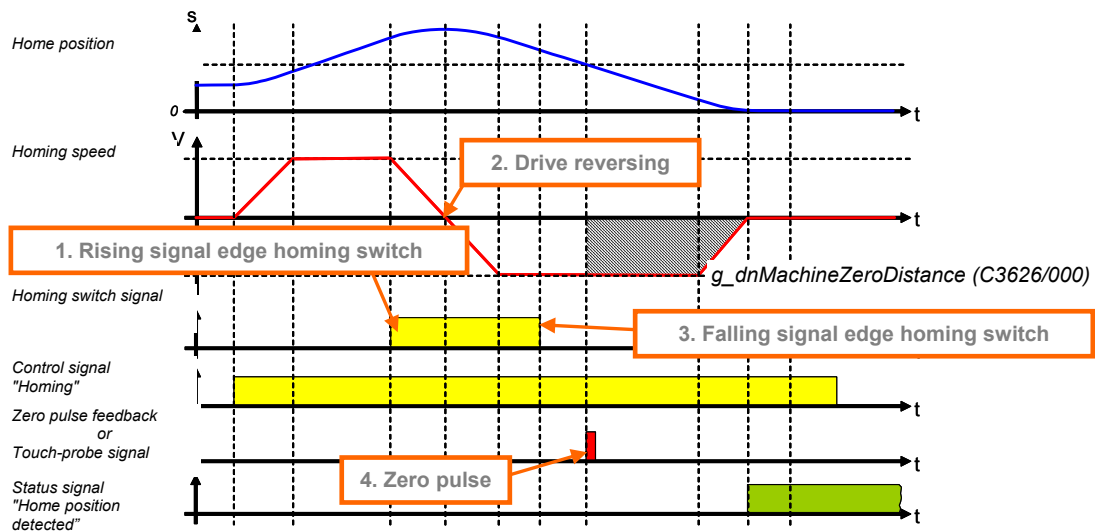


Figure 13: Graph of homing mode 4

In the example illustrated above, the home position has a positive value with reference to the measuring system zero point, because once the zero pulse/touch-probe signal has been detected, the drive continues to move in a negative direction to the measuring system zero point (0.0000[s_units]).

If you do not wish homing to end at the measuring system zero point, you can in addition preset an end position to come after homing (homing target position, C3627/000). This provides the requisite conditions for setting the measuring system zero point outside the drive's traversing range.

- Homing mode 5:

In principle, the sequence is the same as for homing mode 4, but with the direction of rotation reversed.



TIP!

The home position can be set both with the zero pulse from the feedback system and using a touch-probe signal, derived from digital input (I4 using Servo PLC and I2 using ECS). However, this is not standard practice for modes 4 and 5.

Cross Cutter

Features of the "Cross Cutter" prepared solution

- Homing mode 6:

The drive rotates in a positive direction (clockwise rotation) until the homing switch's negative edge is detected. The next touch-probe signal at the assigned digital input determines the home position (machine zero point).

If you do not wish the home position to be the zero point of the absolute measuring system used, you can use an offset value (machine zero distance, C3626/000) to set the home position to the zero point of the measuring system. Once homing is complete the drive will stop at position 0.000[s_units]:

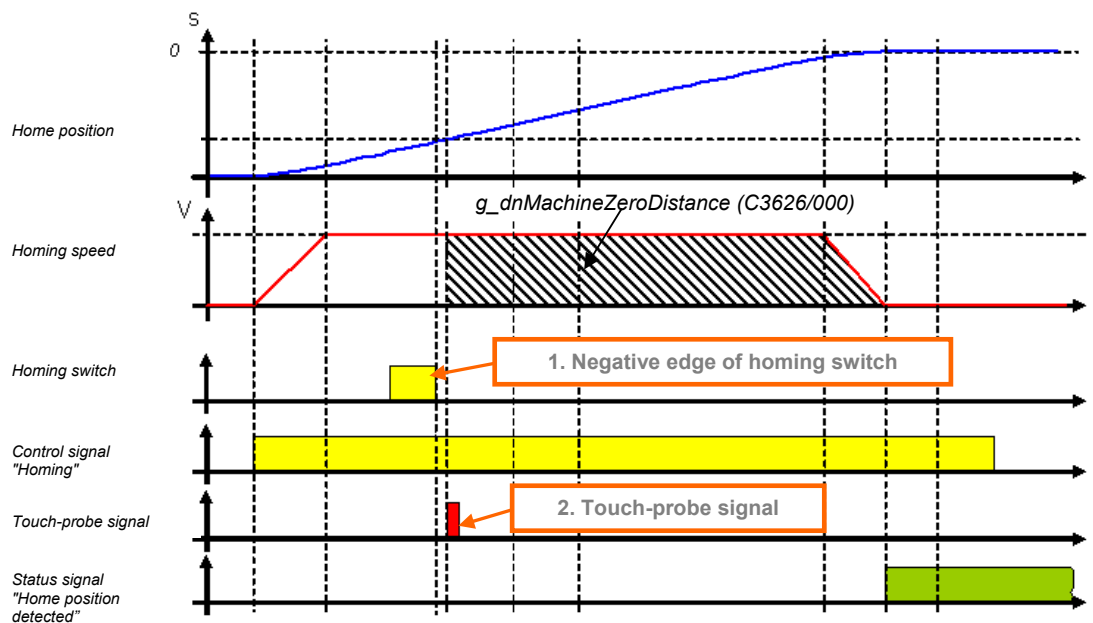


Figure 14: Graph of homing mode 6

In the example illustrated above, the home position has a negative value with reference to the measuring system zero point, because once the touch-probe signal has been detected, the drive continues to move in a positive direction to a position 0.0000[s_units].

If you do not wish homing to end at the measuring system zero point, you can in addition preset an end position to come after homing (homing target position, C3627/000). This provides the requisite conditions for setting the measuring system zero point outside the drive's traversing range.

- Homing mode 7:
(like homing mode 6 but with a negative traversing direction)

Cross Cutter

Features of the "Cross Cutter" prepared solution

- Homing mode 8:

The drive rotates in a positive direction (clockwise rotation) until a touch-probe signal is detected at the assigned digital input. This determines the home position (machine zero point).

If you do not wish the home position to be the zero point of the absolute measuring system used, you can use an offset value (machine zero distance, C3626/000) to set the home position to the zero point of the measuring system. Once homing is complete the drive will stop at position 0.0000[s_units]:

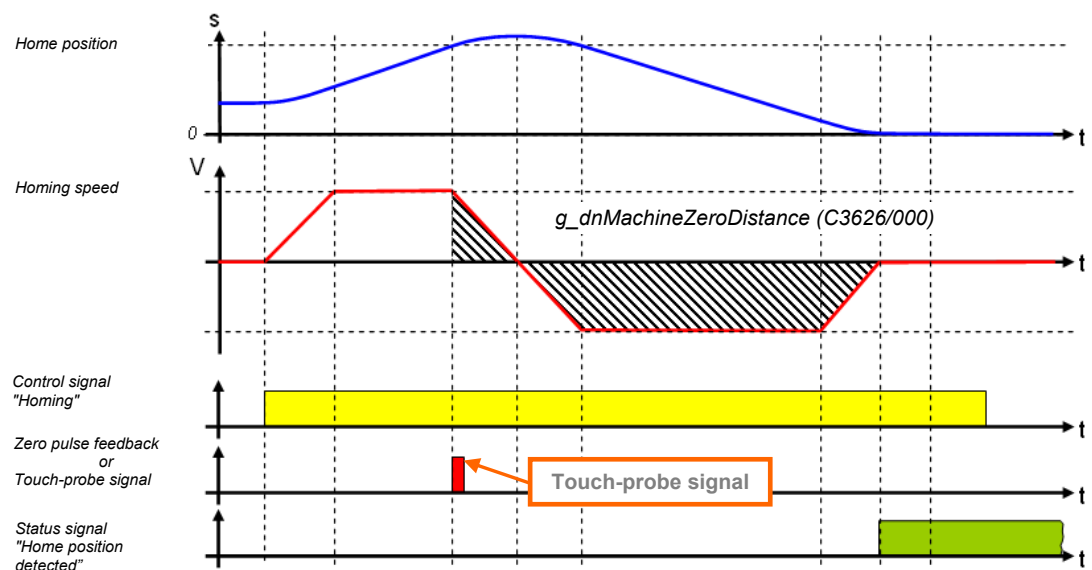


Figure 15: Graph of homing mode 8

In the example illustrated above, the home position has a positive value with reference to the measuring system zero point, because once the zero pulse/touch-probe signal has been detected, the drive reverses and moves back to the measuring system zero point (0.0000[s_units]).

If you do not wish homing to end at the measuring system zero point, you can in addition preset an end position to come after homing (homing target position, C3627/000). This provides the requisite conditions for setting the measuring system zero point outside the drive's traversing range.



TIP!

If a value of 345° is written to C3626 (position of the homing switch) and 165° to C3627 (target position after homing), the drive will first move in a positive direction to the homing switch before reversing in order to position itself at the target position after homing (165°). To prevent the drive reversing, write a value of -15° to C3626 instead of 345° (the drive will then move in a positive direction to the homing switch before continuing in a positive direction to 165° (C3627)).

- Homing mode 9:
(like homing mode 8 but with a negative traversing direction)



TIP!

Instead of the touch-probe signal via digital input (I4 using Servo PLC and I2 using ECS), the zero pulse signal from the feedback system can also be used as the signal to determine the home position (however, this is not standard practice in modes 8 and 9).

Cross Cutter

Features of the "Cross Cutter" prepared solution



TIP!

The homing should always end at the basic position of the cross cutter. As a result, the cross cutter does not move during lock to cam operation. The basic position of the cross cutter, i.e. the position where the cross cutter starts and stops, is displayed in code C3694. This value can then be entered in code C3627 (target position of the tool after homing). The calculation of the basic position and further important aspects concerning the cross cutter motion profile are dealt with in chapter 2.5.2.

Set home position

The prepared solution cross cutter provides the option to set the home position. This function is for example required when absolute value encoders are used. Please proceed as follows to set the home position:

- Inhibit the controller (CINH).
- Align the cross cutter drum in such a way that the knife is in basic position (C3694). See chapter 2.5.2 and pay special attention to Figure 2.
- Enter the basic position of the cross cutter, which is shown in C3694, in C3626.
- Set the home position via the control interface C4135/000 bit 1. In order to do so the controller must be inhibited!
- After the home position is known (status interface C4150/000 bit 17) bit 1 of the control interface C4135/000 can be reset.
- As a result, the home position is saved to the non-volatile memory and will remain available even if a new download is performed.



Attention!

Please note that the function 'set home position' can only be used in combination with absolute value encoder systems. Further information about absolute value encoders is provided in chapter 2.5.16.

Cross Cutter

Features of the "Cross Cutter" prepared solution

Variable names

Inputs		
Global variables (Variable type: VAR_GLOBAL)		
Name	Data type	Value/Meaning
g_bHomingStart	BOOL	Start/stop homing, only in conjunction with controller enable. This value is written in code C4135/000 bit 0. This value is displayed in code C4136/000 bit 0.
		Positive edge from FALSE to TRUE Start homing.
		FALSE Stop/abort homing.
g_bHomingMark	BOOL	Variable for homing switch; preliminary stop position
		TRUE Response in accordance with homing mode set in C3010
		FALSE Response in accordance with homing mode set in C3010
g_bHomePositionSet	BOOL	Setting of home position Only possible if controller inhibit (CINH) set! This value is written in code C4135/000 bit 1. This value is displayed in code C4136/000 bit 1.
		Positive edge from FALSE to TRUE = The setpoint defined in C3626 (position of homing switch) is applied and g_bHomingDone is set to TRUE.
g_bHomePositionReset	BOOL	Reset home. This value is written in code C4135/000 bit 2. This value is displayed in code C4136/000 bit 2.
		TRUE = The g_bHomingDone output is reset to FALSE. The "Home position detected" status is reset (g_bHomingDone = FALSE).
g_dnMachineZeroDistance	DINT	Position of homing switch in [unit] The value is entered in fixed point format with 4 decimal positions (1 [unit] x 10000). The value is converted into increments internally. If you use this variable, to avoid inconsistencies, do not write variable g_dnMachineZeroDistance_p! This value is displayed in code C3626/000; this variable can be written to this code.
g_dnMachineZeroDistance_p	DINT	Position of homing switch in [incr.] If you use this variable, to avoid inconsistencies, do not write variable g_dnMachineZeroDistance!
g_dnHomingTargetPos	DINT	Target position of the tool once homing is complete in [unit] With reference to the machine zero point. The value is entered in fixed point format with 4 decimal positions (1 [unit] x 10000). The value is converted into increments internally. If you use this variable, to avoid inconsistencies, do not write variable g_dnHomingTargetPos_p! This value is displayed in code C3627/000; this variable can be written to this code.
g_dnHomingTargetPos_p	DINT	Target position of the tool once homing is complete in [incr.] With reference to the machine zero point. If you use this variable, to avoid inconsistencies, do not write variable g_dnHomingTargetPos_p!

Cross Cutter

Features of the "Cross Cutter" prepared solution

Inputs		
Global variables (Variable type: VAR_GLOBAL)		
Name	Data type	Value/Meaning
g_byHomingMode	BYTE	Homing mode
		0 > _Rn_MP
		1 < _Rn_MP
		4 > _Rp < _Rn_MP
		5 < _Rp > _Rn_MP
		6 > _Rn > _TP
		7 < _Rn < _TP
		8 > _TP
		9 < _TP
		Symbols: > Movement in positive direction < Movement in negative direction Rp Positive edge of homing switch Rn Negative edge of homing switch TP Touch-probe edge of input I4 using Servo PLC MP Zero pulse edge of motor The execution of the last action in each case will set the home position (e.g. the zero pulse in the case of "MP") even if the drive continues to move subsequently.

Outputs		
Global variables (Variable type: VAR_GLOBAL)		
Name	Data type	Value/Meaning
g_bHomingBusy	BOOL	"Homing activated" status signal This value is displayed in code C4150/000 bit 16.
		TRUE Homing active.
		FALSE Homing not active.
g_bHomingDone	BOOL	"Home status" status signal This value is displayed in code C4150/000 bit 17.
		TRUE Homing complete, home position detected.
		FALSE Home position not detected (an encoder error may have occurred).
g_bHomingError	BOOL	"Error" status signal This value is displayed in code C4150/000 bit 18.
		TRUE An error occurred when setting the home position. See g_nHomingErrorNo for more detailed information about the error.
		FALSE No errors occurred when setting the home position.
g_nHomingErrorNo	INT	Error number
		0 OK
		13 Invalid value range
		111 More than 4 decimal positions entered or no index with fixed point format.
		-10 Controller inhibit (CINH) not set.

Cross Cutter

Features of the "Cross Cutter" prepared solution

Code	Possible settings:		Comment
	Default	Selection	
C3010	0	0 {1}	9 Homing mode
			0 >_Rn_MP
			1 <_Rn_MP
			4 >_Rp_<_Rn_MP
			5 <_Rp_>_Rn_MP
			6 >_Rn_>_TP
			7 <_Rn_<_TP
			8 >_TP
			9 <_TP
			Symbols for code C3010: > Movement in positive direction < Movement in negative direction Rp Positive edge of homing switch Rn Negative edge of homing switch TP Touch-probe edge of input I4 using Servo PLC or I2 using ECS MP Zero pulse edge of motor (one per motor revolution) The execution of the last action in each case will set the home position (e.g. the zero pulse in the case of "MP") even if the drive continues to move subsequently.
C3624 /001	5.00	0.01... {0.01[s]} ...129.99	Deceleration ramp for immediate stop (function abort) for homing Assuming 4000 rpm
C3624 /002	5.00	0.01... {0.01[s]} ...129.99	Acceleration ramp for homing Assuming 4000 rpm
C3624 /003	5.00	0.01... {0.01[s]} ...129.99	Deceleration ramp for homing Assuming 4000 rpm
C3625 /001	200	0 ... {1 rpm} ...15000	Speed for homing in positive direction (CW)
C3625 /002	200	0 ... {1 rpm} ...15000	Speed for homing in negative direction (CCW)
C3626 /000	0	0 {0.0001 [s_units]} 214000.0000	Entry indicating the position of the homing switch
C3627 /000	0	0 {0.0001 [s_units]} 214000.0000	Target position of tool following homing
C4135/000 Bit 0	0	0 {1}	1 Application control word bit 0: Start/stop homing, only in conjunction with controller enable.
			FALSE => TRUE Start homing.
			FALSE Stop/abort homing.
C4135/000 Bit 1	0	0 {1}	1 Application control word bit 1: Set home position.
			FALSE => TRUE The setpoint defined in C3626 (position of homing switch) is applied and g_bHomingDone is set to TRUE.
			FALSE Home position not set.
C4135/000 Bit 2	0	0 {1}	1 Application control word bit 2: Reset home position.
			TRUE The home position is reset. The g_bHomingDone output is reset to FALSE. The "Home position detected" status is reset.
			FALSE Home position not reset.

Cross Cutter

Features of the "Cross Cutter" prepared solution

Code	Possible settings:		Comment
	Default	Selection	
C4150/ 000 Bit 16	-	-	Application status word bit 16: Homing activated.
			TRUE Homing active.
			FALSE Homing not active.
C4150/ 000 Bit 17	-	-	Application status word bit 17: Home position detected.
			TRUE Homing complete, home position detected.
			FALSE Home position not detected (an encoder error may have occurred).
C4150/ 000 Bit 18	-	-	Application status word bit 18: "Homing error" status signal.
			TRUE An error occurred when setting the home position. See g_nHomingErrorNo for more detailed information about the error.
			FALSE No errors occurred when setting the home position.



Note!

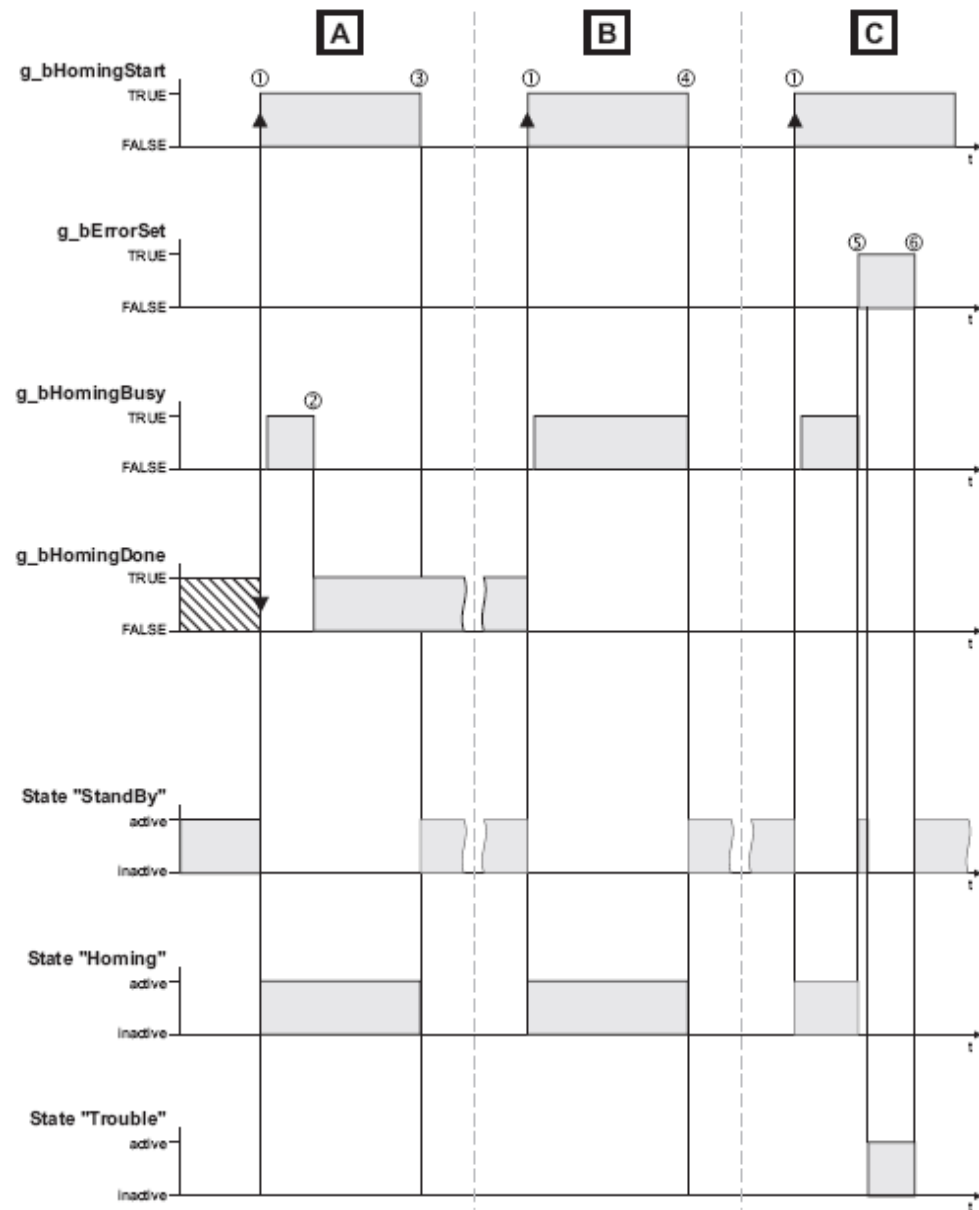
Tip for assigning the PLC's digital input to the homing switch: In the prepared solution, digital input 3 is designated as the homing switch.

The signals are wired in the CrossCutterInterfaces PRG (network 3).

Instead of the touch-probe signal via digital input (I4 using Servo PLC and I2 using ECS), the zero pulse signal from the feedback system can also be used as the signal to determine the home position.

Cross Cutter

Features of the "Cross Cutter" prepared solution



Info	Zustandswechsel
A Korrekte Ausführung der Funktion	
① Wechsel in den Zustand "Homing" (Flankenbewertung von <i>g_bHomingStart</i>).	Standby → Homing
② Die Referenzsuche wurde durchgeführt und die Referenz ist bekannt (<i>g_bHomingDone</i> = TRUE).	
③ Die Funktion Homing wird regulär beendet (Pegelauswertung von <i>g_bHomingStart</i>).	Homing → Standby
B Abbruch der Funktion	
① Wechsel in den Zustand "Homing" (Flankenbewertung von <i>g_bHomingStart</i>).	Standby → Homing
④ Die Funktion Homing wird beendet, bevor die Referenzsuche abgeschlossen ist.	Homing → Standby
C Fehlerfall	
① Wechsel in den Zustand "Homing" (Flankenbewertung von <i>g_bHomingStart</i>).	Standby → Homing
② Ein Fehler ist aufgetreten, dadurch wird in den Zustand "Trouble" gewechselt.	Homing → Standby → Trouble
⑥ Durch Quittieren des Fehlers wird der Zustand "Trouble" wieder verlassen.	Trouble → Standby

Figure 16: Behaviour of the "homing" operating mode

Cross Cutter

Features of the "Cross Cutter" prepared solution

2.5.7 Simple positioning

Users can use the Simple positioning mode to approach any position on the knife drum. Possible applications involving simple positioning on the cross cutter include approaching the knife change position or knife idle position. Four modes are available to the user for such applications:

- Approach target position via shortest route
- Approach target position clockwise
- Approach target position counter clockwise
- Relative positioning

The "Simple positioning" operating mode is not determined by the motion profile (cam) but follows linear ramps independently of the profile. In this operating mode, a master value entry and the cam function will have no effect on drive motion.

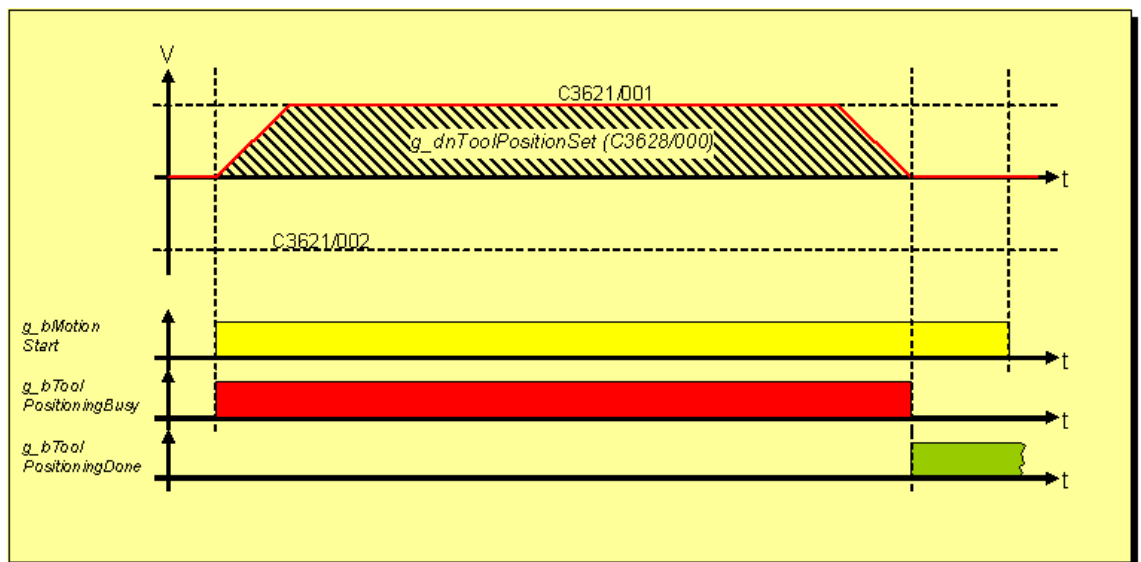


Figure 17: Signal characteristics for positioning

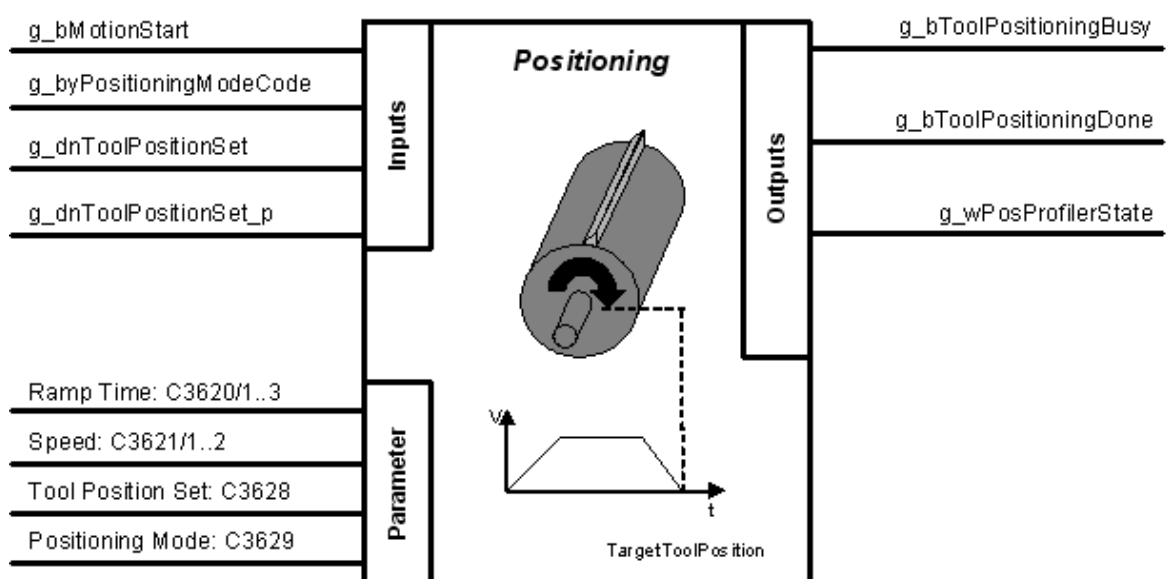


Figure 18: Control signals in "Positioning" mode

Cross Cutter

Features of the "Cross Cutter" prepared solution

Variable names

Inputs		
Global variables (Variable type: VAR_GLOBAL)		
Name	Data type	Value/Meaning
g_bMotionStart	BOOL	Execute positioning. This value is written in code C4135/000 bit 5. This value is displayed in code C4136/000 bit 5.
		TRUE The drive is moved to the target position g_dnToolPositionSet(_p) along the set ramps and at the set speed.
		FALSE Positioning interrupted.
g_byPositioningModeCode	BYTE	Positioning mode This value is written/displayed in C3629/000.
		0 = Move to target position via shortest route. 1 = Move to target position only in positive direction (CW). 2 = Move to target position only in negative direction (CCW). 4 = Positioning relative to entry in g_dnToolPositionSet(_p).
g_dnToolPositionSet	DINT	Target position in [unit] With reference to the machine zero point. The value is entered in fixed point format with 4 decimal positions (1 [unit] = 10000). The value is converted into increments internally. If you use this variable, to avoid inconsistencies, do not write variable g_dnToolPositionSet_p! This value is displayed in code C3628/000; this variable can be written to this code.
g_dnToolPositionSet_p	DINT	Target position in [incr.] With reference to the machine zero point. If you use this variable, to avoid inconsistencies, do not write variable g_dnToolPositionSet!

Outputs		
Global variables (Variable type: VAR_GLOBAL)		
Name	Data type	Value/Meaning
g_bToolPositioningBusy	BOOL	"Positioning in progress" status signal This value is displayed in code C4150/000 bit 3.
		TRUE Positioning in progress.
		FALSE Positioning not in progress.
g_bToolPositioningDone	BOOL	"Target position reached" status signal This value is displayed in code C4150/000 bit 4.
		TRUE Target position g_dnToolPositionSet(_p) reached.
		FALSE Target position g_dnToolPositionSet(_p) not reached.
g_wPosProfilerState	WORD	Status of positioning:
		0 = NoAction 1 = PositioningDone 13 = PositioningBusy 14 = PositioningToCamBusy 102 = AbortsActive Target position g_dnToolPositionSet(_p) reached. Positioning in progress. Positioning to cam in progress. Positioning aborted.

Cross Cutter

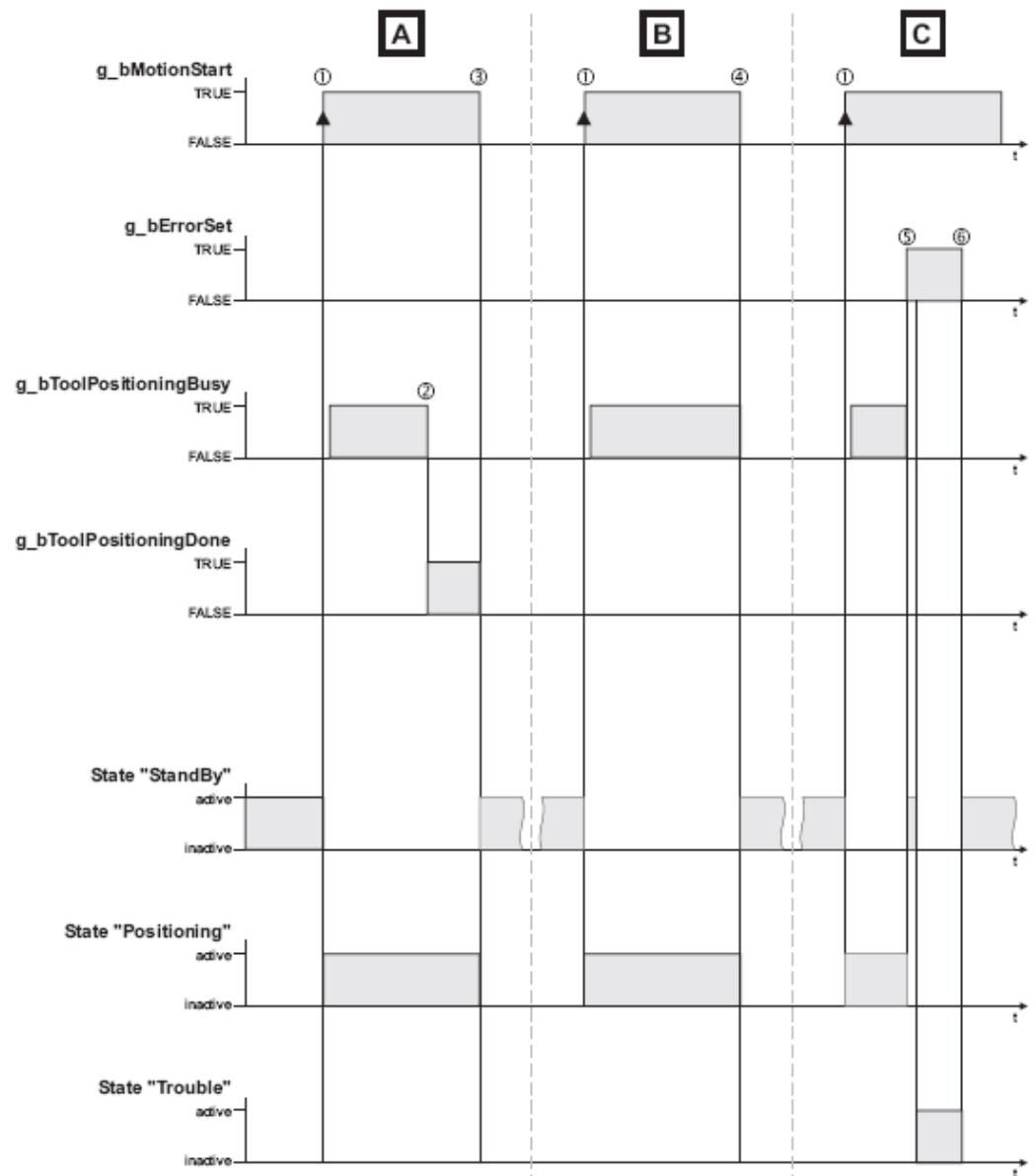
Features of the "Cross Cutter" prepared solution

Code	Possible settings:				Comment
	Default	Selection			
C3620/001	5.00	0.01...	{0.01[s]}	...129.99	Deceleration ramp for immediate stop Assuming 4000 rpm
C3620/002	5.00	0.01...	{0.01[s]}	...129.99	Acceleration ramp for positioning Assuming 4000 rpm
C3620/003	5.00	0.01...	{0.01[s]}	...129.99	Deceleration ramp for positioning Assuming 4000 rpm
C3621/001	200	0 ...	{1 rpm}	...15000	Speed for positioning in positive direction (CW)
C3621/002	200	0 ...	{1 rpm}	...15000	Speed for positioning in negative direction (CCW)
C3628/000	0.0000	0	{0.0001 [s_units]}	360.0000	Target position of positioning
C3629/000	0	0	{1}	4	Positioning mode 0 = Move to target position via shortest route. 1 = Move to target position only in positive direction (CW). 2 = Move to target position only in negative direction (CCW). 4 = Positioning relative to entry in g_dnToolPositionSet(_p).
C4135/000 Bit 5	0	0	{1}	1	Application control word bit 5: Execute positioning.
					TRUE The drive is moved to the target position along the set ramps and at the set speed.
					FALSE Positioning interrupted.
C4150/000 Bit 3	-	-	-	-	Application status word bit 3: Positioning in progress.
					TRUE Positioning in progress.
					FALSE Positioning not in progress.
C4150/000 Bit 4	-	-	-	-	Application status word bit 4: Target position reached.
					TRUE Target position reached.
					FALSE Target position not reached.

To illustrate the positioning function, a graph showing the signal characteristic of a positioning operation appears on the following page. The "StandBy", "Positioning" and "Trouble" states shown are displayed in code C3505 or global variable *g_nTemplateState*.

Cross Cutter

Features of the "Cross Cutter" prepared solution



Info	Zustandswechsel
A Korrekte Ausführung der Funktion	
① Wechsel in den Zustand "Positioning" (Flankenbewertung von $g_bMotionStart$).	Standby → Positioning
② Die Positionierung wurde durchgeführt und die Zielposition ist erreicht ($g_bToolPositioningDone = TRUE$).	
③ Die Funktion Positioning wird regulär beendet (Pegelauswertung von $g_bMotionStart$).	Positioning → Standby
B Abbruch der Funktion	
① Wechsel in den Zustand "Positioning" (Flankenbewertung von $g_bMotionStart$).	Standby → Positioning
④ Die Funktion Positioning wird beendet, bevor die Zielposition erreicht wurde.	Positioning → Standby
C Fehlerfall	
① Wechsel in den Zustand "Positioning" (Flankenbewertung von $g_bMotionStart$).	Standby → Positioning
⑤ Ein Fehler ist aufgetreten, dadurch wird in den Zustand "Trouble" gewechselt.	Positioning → Standby → Trouble
⑥ Durch Quittierung des Fehlers wird der Zustand "Trouble" wieder verlassen.	Trouble → Standby

Figure 19: Behaviour of the "Positioning" mode

Cross Cutter

Features of the "Cross Cutter" prepared solution

2.5.8 Length-controlled or mark-controlled operation

The "Cross Cutter" prepared solution supports both length-controlled and mark-controlled operation.

Length-controlled operation:

Synchronous cross cutter:

In length-controlled operation on the synchronous cross cutter, motion profiles are calculated online in the target system¹ and this is the only calculation made. Because the profiles are calculated online in the target system and not in advance on a PC (e.g. using the **CamDesigner** software tool), the motion profile can be quickly adapted to meet the needs of the user. In rotary cross-cutter applications where (in theory) an infinite number of cutting lengths are possible in particular, the number of motion profiles required usually exceeds the number of motion profiles that can actually be saved (9300ET/ECS/SWP-CAM: maximum 48 motion profiles). In such cases, online calculation of cross-cutter motion profiles in the target system offers an alternative. The cross-cutter motion profile is created in two sections (synchronous phase and asynchronous phase) which are adapted on the basis of cutting length (see also Chapter 2.2.2).

Asynchronous cross cutter:

In length-controlled operation on the asynchronous cross cutter, a maximum cutting length corresponding to the circumference of the cutting circle knife applies. This restriction is necessary to prevent the material to be cut jamming.

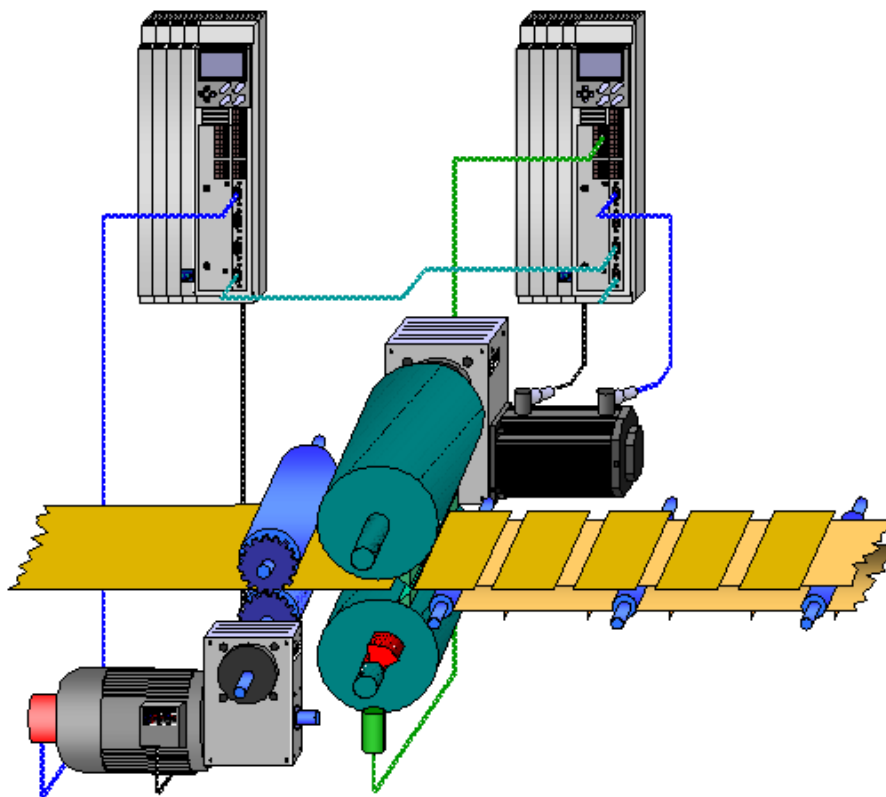


Figure 20: Length-controlled operation of a cross-cutter application

¹ No calculation in real time!

Cross Cutter

Features of the "Cross Cutter" prepared solution

Mark-controlled operation:

In mark-controlled operation, mark control is implemented in addition to length control. Mark control compensates inaccuracies via touch-probe correction. This ensures that positions are corrected accordingly during operation. High-speed hardware signals ("touch probe") which determine the deviation between the set and actual position on an interrupt-specific basis in the ServoPLC are used for this purpose. An example application of mark control is illustrated in Figure 21. In this application, mark control is used to cut a defined length from the material being processed using print marks. In an application of this type, the length to be cut is sent to the system and the length calculation function converts this value into an appropriate motion profile. The print marks on the material to be processed are then used to correct the length calculation. As soon as the touch-probe sensor detects a print mark, it compares the position set accordingly with the actual position. In accordance with the deviation a signal is then generated with a positive or negative sign to compensate the prevailing path difference. Mark control can be used equally for the synchronous and asynchronous cross cutter. Mark correction is only activated in Continuous cutting mode.

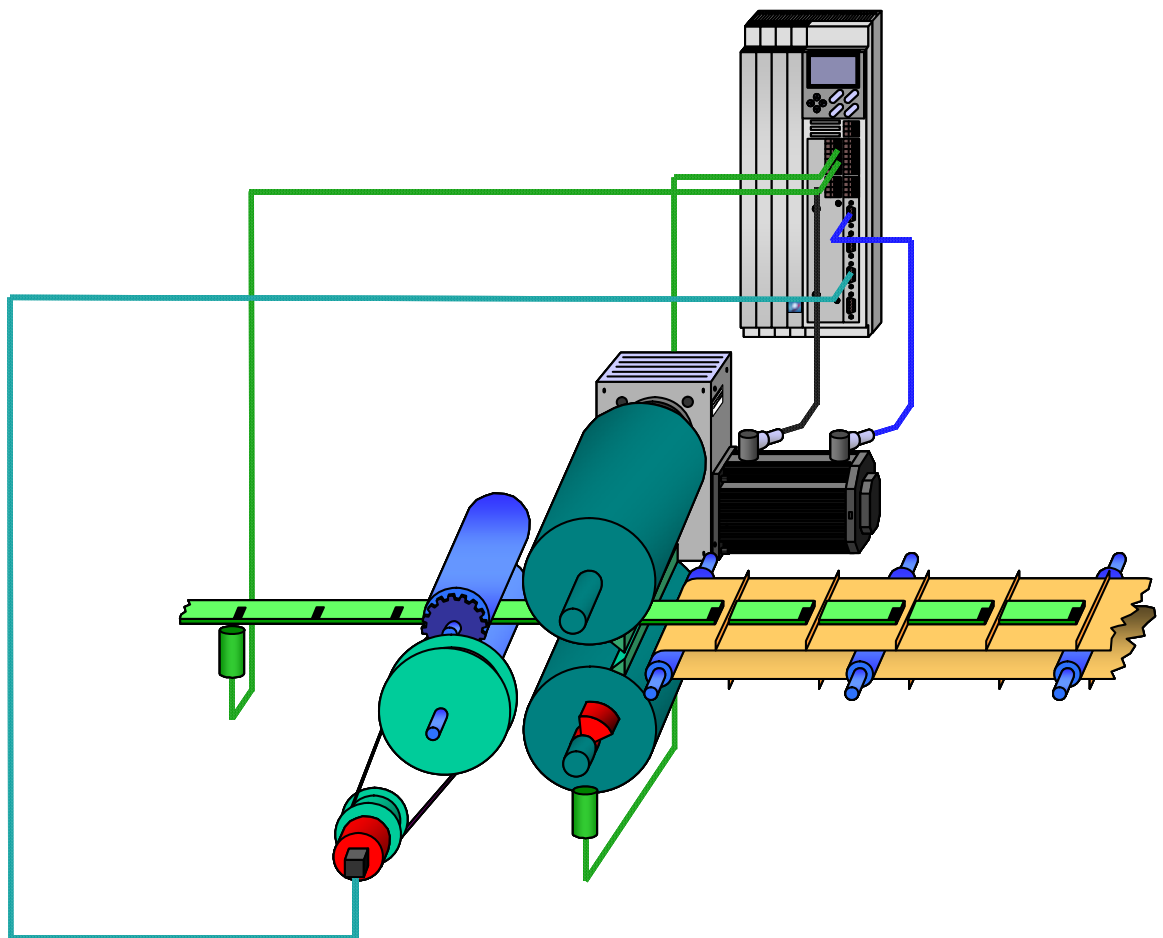


Figure 21: Mark-controlled operation of a cross-cutter application

Cross Cutter

Features of the "Cross Cutter" prepared solution

2.5.9 Adjustable compensation speed for mark correction

There are two ways of compensating the detected path difference. Either a speed signal can be applied to the master speed directly to compensate the path difference or the detected path difference can be compensated by superimposing a positioning profile. When compensating the path difference please note that the path difference may not be compensated if the knife is cutting the material. If compensation is applied during this phase, the speed of the knife will no longer be synchronised with the material speed and this will damage the material. A function block interconnection has been implemented in the prepared solution accordingly to prevent this happening. Any mark correction in progress will be aborted as soon as the knife cuts into the material. Mark correction is only activated in Continuous cutting mode.

Mark-controlled operation with touch-probe synchronisation without ramp generator:

The detected path difference is compensated with a fixed additional speed on the actual cam value (see Figure 22). For this purpose, a fixed additional speed is applied to the master value until the path difference is compensated. At this point the additional speed is withdrawn. If the cross-cutter knife cuts into the material whilst mark correction is active, correction will be aborted by setting the speed to 0. This will prevent deviations in the synchronous speed during cutting. To activate mark-controlled operation with mark synchronisation but no ramp generator, bit 21 must be enabled in application control word code C4135/000 (*g_bXTpEnable* variable = TRUE) and bit 22 disabled (*g_bRfgTPEnable* variable = FALSE). The value of the fixed additional speed is written in code C3650/001 and transmitted to global variable *g_nTPMakeUpLeeway_v* (the signals are wired in the CrossCutterInterfaces PRG (network 6)). Bit 7 of application status code C4150/000 indicates that mark correction is active (*g_bXTpSyncBusy* variable = TRUE).

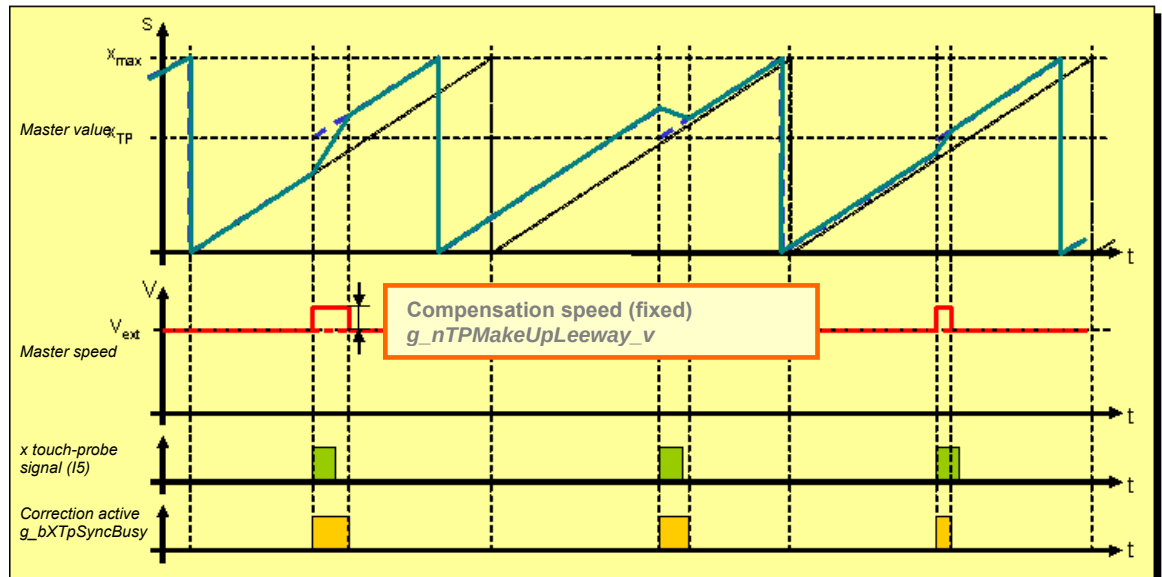


Figure 22: Graph illustrating mark control with touch-probe synchronisation but no ramp generator

Cross Cutter

Features of the "Cross Cutter" prepared solution

Mark-controlled operation with mark synchronisation and ramp generator:

A positioning profile is superimposed to compensate the detected path difference (see Figure 23). This prevents speed step changes in the master value signal during path difference compensation (such step changes are typical of mark correction without a ramp generator). A speed signal generated from the positioning profile is superimposed on the master value signal to compensate the path difference. The use of positioning profile ramps enables the compensation speed to be ramped up slowly and then back down at the end. To activate mark-controlled operation with mark synchronisation and ramp generator, bit 21 must be enabled in application control word code C4135/000 (*g_bXTpEnable* variable = TRUE) along with bit 22 (*g_bRfgTPEnable* variable = TRUE). A maximum possible correction speed can be written in code C3650/001 and a maximum negative correction speed in code C3650/002. The ramp times for the abort ramp, acceleration ramp and deceleration ramp can be written in codes C3649/001, C3649/002 and C3649/003 respectively. The abort ramp is used whenever the cross-cutter knife cuts into the material when mark correction is active. In such an instance, mark correction will be aborted with the time set in the abort ramp. This will prevent deviations in the synchronous speed during cutting. The abort ramp is controlled via global variable *g_bRfgXTpExecute*. This variable is always TRUE if the knife is not cutting the material. In such circumstances it will therefore be possible to compensate a detected position deviation. Bit 7 of application status code C4150/000 indicates that mark correction is active (*g_bXTpSyncBusy* variable = TRUE).

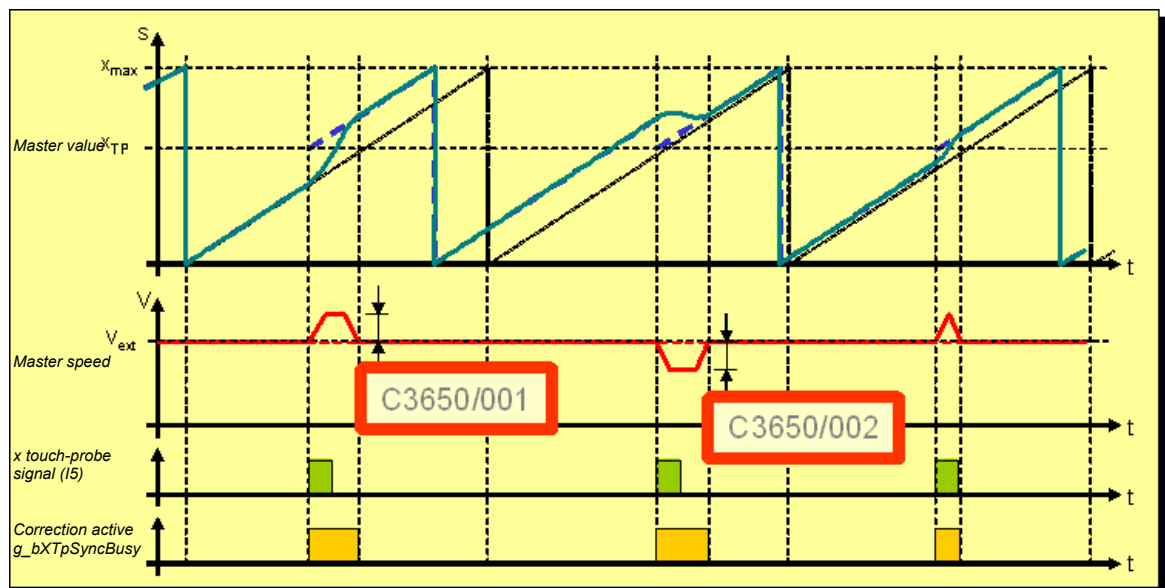


Figure 23: Graph illustrating mark control with touch-probe synchronisation and ramp generator

Cross Cutter

Features of the "Cross Cutter" prepared solution

Variable names

Inputs		
Global variables (Variable type: VAR_GLOBAL)		
Name	Data type	Value/Meaning
g_bXTpEnable	BOOL	Enable print-mark synchronisation. This value is written in code C4135/000 bit 21. This value is displayed in code C4136/000 bit 21.
		TRUE Print-mark synchronisation enabled; length-controlled operation disabled.
		FALSE Print-mark synchronisation disabled; length-controlled operation enabled.
g_bRfgXTpEnable	BOOL	Enable ramp generator for print-mark synchronisation. This value is written in code C4135/000 bit 22. This value is displayed in code C4136/000 bit 22.
		TRUE Ramp generator for print-mark synchronisation enabled.
		FALSE Ramp generator for print-mark synchronisation disabled.
g_bRfgXTpExecute	BOOL	Start/abort ramp generator for correcting position deviations. This global variable is used to abort active mark correction with a ramp generator if the knife is cutting into the material.
		TRUE The detected position deviation (g_dnXDifference_p) is corrected using the internal ramp generator.
		FALSE The ramp generator is not active, the detected position deviation is not corrected. If mark correction is active, it is stopped via the abort ramp.
g_dnMeasuredTPDistance	DINT	Distance measured from mark sensor to 6 o'clock position of knife in [unit]. This value is displayed in code C3668/000; this variable can be written to this code. The value is entered in fixed point format with 4 decimal positions (1 [unit] = 10000). The value is converted into increments internally. If you use this variable, to avoid inconsistencies, do not write variable g_dnMeasuredTPDistance_p!
g_dnMeasuredTPDistance_p	DINT	Distance measured from mark sensor to 6 o'clock position of knife in [incr.]. If you use this variable, to avoid inconsistencies, do not write variable g_dnMeasuredTPDistance!
g_dnXTpPos	DINT	Distance from zero point to touch-probe sensor in [unit] The value for this global variable is calculated automatically! Internal limitation to 0 ... 2 * g_dnVertShaftLengthAct_p.
g_dnXTpPos_p	DINT	Distance from zero point to print-mark sensor in [incr.]. The value for this global variable is calculated automatically! If you use this variable, to avoid inconsistencies, do not write variable g_dnXTpPos!
g_nXTpMakeUpLeeway_v	INT	Speed in [incr./ms] for correcting the detected position deviation. Only relevant for correction without a ramp generator (g_bRfgXTpEnable = FALSE). This value is displayed in code C3650/001; this variable can be written to this code.
g_bXTpReceived	BOOL	Trigger input for calculating position deviation.
g_DFIN_dnIncLastScan_p	DINT	Δ inc between mark sensor signal and start of task. This global variable is linked directly to SB DFIN_IO_DigitalFrequency.

Cross Cutter

Features of the "Cross Cutter" prepared solution

Outputs		
Global variables (Variable type: VAR_GLOBAL)		
Name	Data type	Value/Meaning
g_bXTpSyncBusy	BOOL	"Mark correction in progress" status signal This value is displayed in code C4150/000 bit 7.
		TRUE = Mark correction active, correcting position difference. FALSE = Mark correction not active at this time.
g_bKnifeNotInMaterial	BOOL	"Knife not in material" status signal This value is displayed in code C4150/000 bit 29.
		TRUE = Knife is not in material. FALSE = Knife is in material.
g_dnXDifference_p	DINT	Correction value of mark correction This value is only valid if mark correction with a ramp generator has been selected (C4135/000 Bit 22 = TRUE, or g_bRfgXTpEnable = TRUE).
g_dnXTpDifference_p	DINT	Difference between two print-mark signals This value is only valid if mark correction with a ramp generator has been selected (C4135/000 Bit 22 = TRUE, or g_bRfgXTpEnable = TRUE).

Code	Possible settings:				Comment
	Default	Selection			
C3649/001	0.100	0.000... {0.01[s]} ...129.99			Abort ramp for ramp generator: Assumes a speed of 4000 rpm. Used if the knife cuts into the material whilst mark correction is in progress.
C3649/002	5.000	0.000... {0.01[s]} ...129.99			Acceleration ramp for ramp generator: Assumes a speed of 4000 rpm.
C3649/003	5.000	0.000... {0.01[s]} ...129.99			Deceleration ramp for ramp generator: Assumes a speed of 4000 rpm.
C3650/001	200	0 {1 rpm} 15000			Maximum positive correction speed for mark synchronisation with a ramp generator and correction speed for mark synchronisation without a ramp generator.
C3650/002	200	{1 rpm} 15000			Maximum negative correction speed for mark synchronisation with a ramp generator.
C3668/000	370.0000	0.0000 {1.0000 m_units} 214000.0000			Distance measured from mark sensor to 6 o'clock position of knife.
C4135/000 Bit 21	0	0 {1} 1			Application control word bit 21: Activate print-mark synchronisation.
			TRUE		Print-mark synchronisation activated.
			FALSE		Print-mark synchronisation deactivated.
C4135/000 Bit 22	0	0 {1} 1			Application control word bit 22: Activate ramp generator for print-mark synchronisation.
			TRUE		Ramp generator activated.
			FALSE		Ramp generator deactivated.
C4150/000 Bit 7	-	- - -			Application status word bit 7: Mark correction active.
			TRUE		Mark correction active.
			FALSE		Mark correction not active.
C4150/000 Bit 29	-	- - -			Application status word bit 29: Knife is in material.
			TRUE		Knife in material.
			FALSE		Knife not in material.



Note!

Digital input I5 on the Servo PLC or I1 on the ECS is permanently assigned to the print-mark-sensor signal. The user needs to assign the print-mark-sensor signal to this input!

Cross Cutter

Features of the "Cross Cutter" prepared solution

2.5.10 Mark window

It may be necessary to only evaluate certain mark signals. This will be the case if there are other marks on the material to be processed in addition to the print marks but these other marks will not generate a response (see Figure 24). A mark window is defined for such scenarios. This provides a means of checking whether the print-mark signal received is located in a specific position window. The incoming print-mark signal must be located in this position window in order to be evaluated. The size of the position window is set by the user in code C3669/000 (*g_dnTPPositionWindow* variable). All print-mark signals outside this window are ignored (see Figure 25 and Figure 26). If there are no print-mark signals in this position window, a message will be generated and displayed in bit 26 of application status code C4150/000 (*g_bMaxNumberOfTPOutOfWindowIncreased* variable = TRUE). The user can set a limit indicating when a message should be generated (the user should write the value of this limit in code C3670/000 (*g_wMaxNumberOfTpOutOfWindow* variable)). This warning signal is reset automatically when the next valid mark signal appears within the position window. The mark window is only active when mark correction is selected. Mark sensor signals will be blocked when mark-controlled operation is deselected.

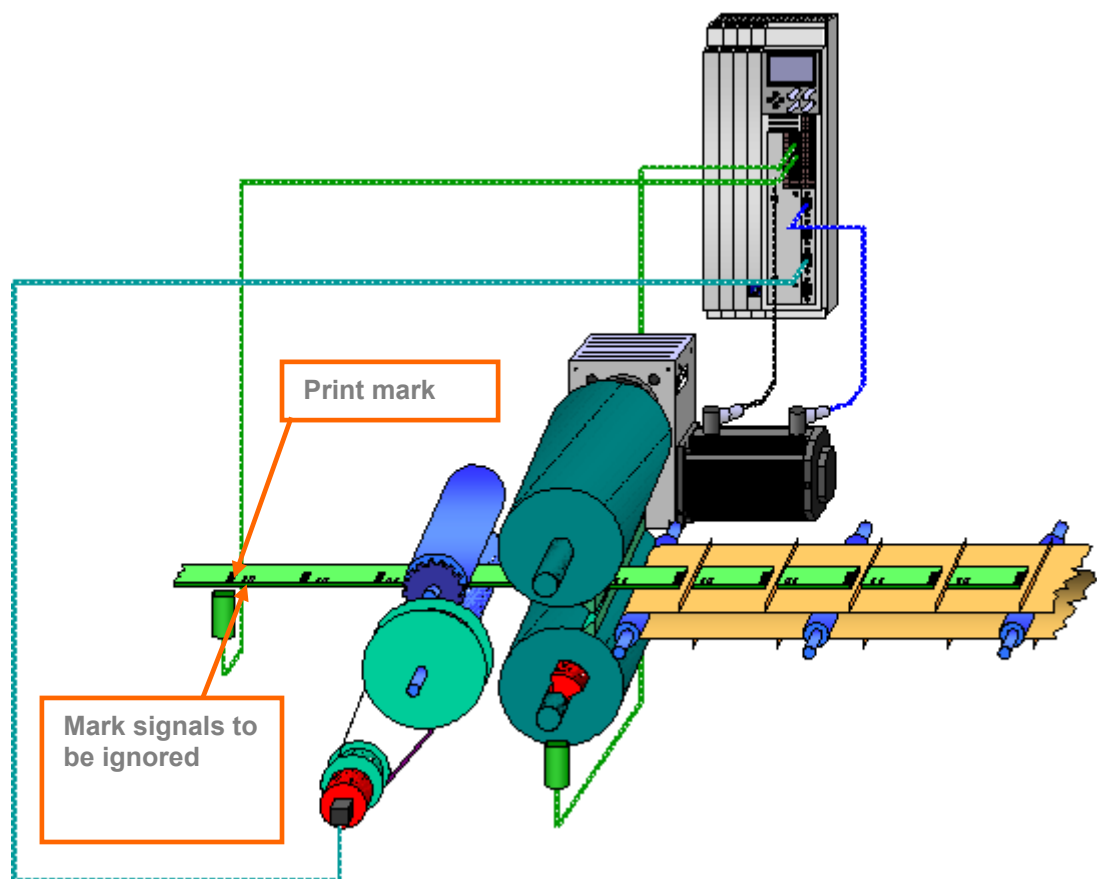


Figure 24: Example of an application in which mark windows are used

Cross Cutter

Features of the "Cross Cutter" prepared solution

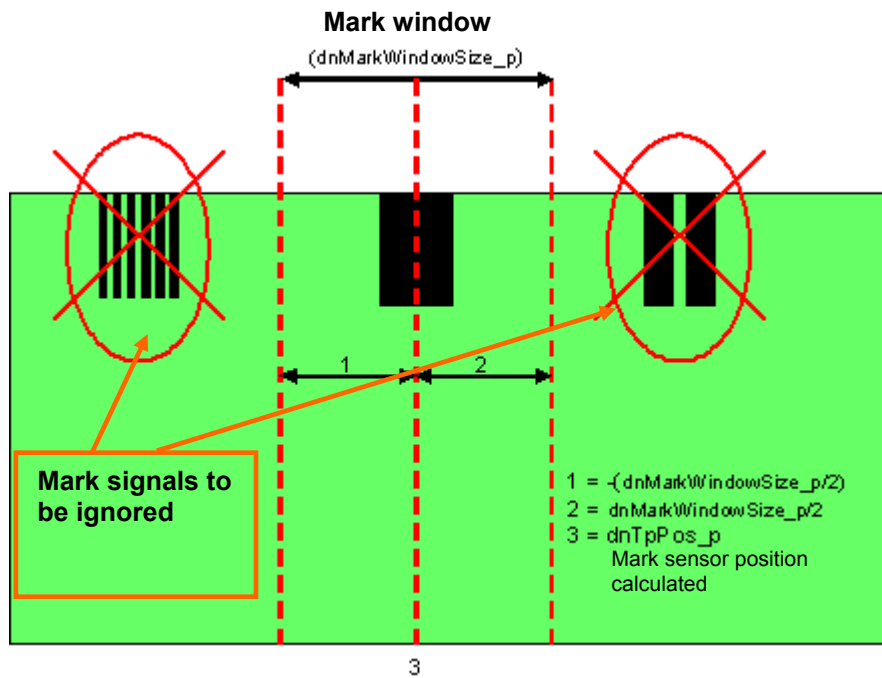


Figure 25: Schematic representation illustrating how the mark window works

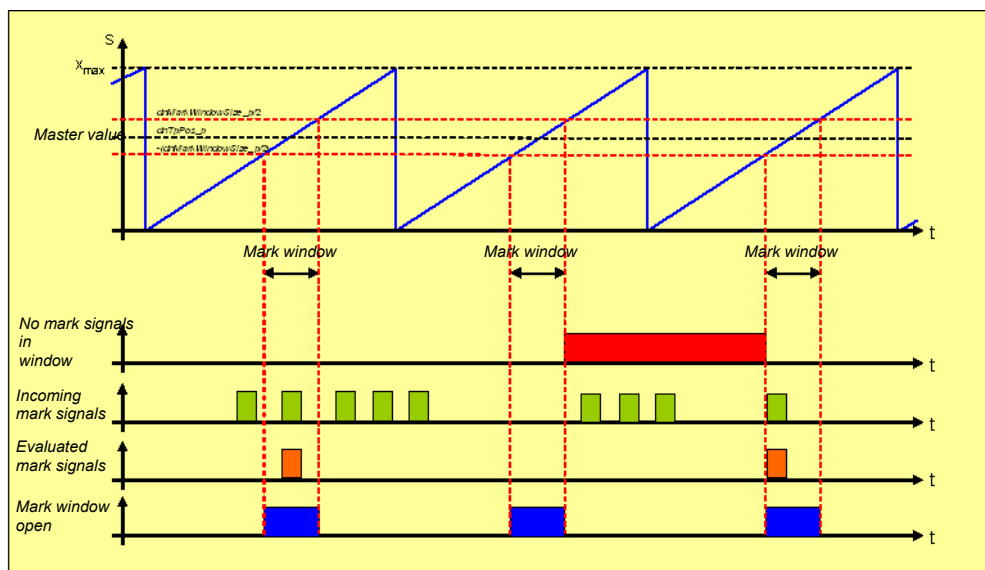


Figure 26: Schematic representation of signal generation

Cross Cutter

Features of the "Cross Cutter" prepared solution

Variable names

Inputs		
Global variables (Variable type: VAR_GLOBAL)		
Name	Data type	Value/Meaning
g_bResetCounterTPOutOfWindow	BOOL	Variable to reset the counter for mark signals located outside the mark window.
		TRUE Reset counter.
		FALSE Do not reset counter.
g_DFIN_bTPReceived_b	BOOL	Trigger input for calculating position deviation. This signal is use for incoming print-mark-sensor signals. This global variable is linked directly to <i>SB DFIN_IO_DigitalFrequency</i> . Digital input I5 is assigned to the mark sensor!
g_DFIN_dnIncLastScan_p	DINT	Δ incr between mark sensor signal and start of task. This global variable is linked directly to <i>SB DFIN_IO_DigitalFrequency</i> .
g_dnTPPositionWindow	DINT	Size of mark window in 0.0001[m_units]. This value is displayed in code C3669/000; this variable can be written to this code. The value is converted into increments internally. If you use this variable, to avoid inconsistencies, do not write variable g_dnTPPositionWindow_p!
g_dnTPPositionWindow_p	DINT	Size of mark window in [incr]. If you use this variable, to avoid inconsistencies, do not write variable g_dnTPPositionWindow!
g_wMaxNumberOfTpOutOfWindow	WORD	Number of cycles permitted with no mark signals appearing in the mark window. If this number is exceeded, a warning signal will be generated. This value is displayed in code C3670/000; this variable can be written to this code.

Outputs (Variable type: VAR_GLOBAL)		
Name	Data type	Value/Meaning
g_bXTpReceived	BOOL	Output signal for mark correction. This output signal outputs all print-mark signals located inside the defined mark window. All other print-mark signals are not output.
g_bTpWindowOpen	BOOL	This output indicates the time at which the window is open.
		TRUE The mark window is open. Incoming mark signals are output at g_bXTpReceived.
		FALSE The mark window is closed. Incoming mark signals are not output at g_bXTpReceived.
g_bTPWindowActive	BOOL	This output indicates that the mark window is active and the first print-mark signal has been received.
		TRUE The mark window is active.
		FALSE The mark window is not active. The first print mark has not yet been detected.
g_dnTPDistance_p	DINT	This output signal indicates the distance between two print marks in increments.

Cross Cutter

Features of the "Cross Cutter" prepared solution

Outputs (Variable type: VAR_GLOBAL)		
Name	Data type	Value/Meaning
g_bMaxNumberOfTPOutOfWindow Increased	BOOL	This output signal indicates that the maximum number of cycles completed without a mark signal appearing in the mark window has been exceeded. This value is displayed in code C4150/000 bit 26.
		TRUE The maximum number of cycles without a mark signal appearing in the mark window has been reached.
		FALSE The maximum number of cycles without a mark signal appearing in the mark window has not been reached.
g_wNumberOfTpOutOfWindow	WORD	Number of cycles completed without a mark signal appearing in the mark window.

Cross Cutter

Features of the "Cross Cutter" prepared solution

Code	Possible settings:		Comment
	Default	Selection	
C3669/ 000	60.0000	0.0000... {0.0001 ... 214000.0000 m_units}	Size of mark window
C3670/ 000	1	0 ... {1} ... 65535	Number of cycles permitted without a mark signal appearing in the mark window. A warning signal is generated if this number is exceeded.
C4150/ 000 Bit 26	-	-	Application status word bit 26: This status signal indicates that the maximum number of cycles completed without a mark signal appearing in the mark window has been exceeded.
			TRUE The maximum number of cycles without a mark signal appearing in the mark window has been reached.
			FALSE The maximum number of cycles without a mark signal appearing in the mark window has not been reached.

The warning signals listed in the table below may occur during monitoring of the mark window. Each warning signal will remain pending until the next print-mark signal appears in the mark window. Warning signals can also be displayed via bit 26 of the application status word (C4150/000).

Each warning signal is generated in the specified network of the *UserErrors* block (POU).

Warning signals			
Number	Text	Global variable	Network in UserErrors POU
507	MaxNumberOfTPOutOfWin	g_bMaxNumberOfTPOutOfWindowIncreased	8



Note!

Digital input 5 on the Servo PLC or I1 on the ECS is permanently assigned to the print-mark-sensor signal. The user needs to assign the print-mark-sensor signal to this input!

Cross Cutter

Features of the "Cross Cutter" prepared solution

2.5.11 Manual X trimming

Manual X trimming allows the machine operator to shift the cutting position in a positive or negative direction. This is usually carried out during machine setting-up operation. The machine operator can move the cutting position by pressing two pushbuttons until the required format is reached (see Figure27). Two bits are available to the user for this operation in the application status word. For trimming in a positive direction, bit 19 in application control code C4135/000 needs to be activated (*g_bXAxisTrimmingPosDirection* global variable = TRUE). For trimming in a negative direction, bit 20 in application control code C4135/000 needs to be activated (*g_bXAxisTrimmingNegDirection* global variable = TRUE). Code C3678/000 (*g_nXAxisTrimmingFactor_v* global variable) contains the inching speed value for trimming in a positive and negative direction. The acceleration ramp and the deceleration ramp for inching are specified in codes C3676/000 and C3677/000. Where long cutting lengths are concerned, we recommend performing X trimming by entering a value directly. Users can enter an absolute target position in code C3674/000 (*g_dnSetXAxisTrimmingTPPos* global variable). To enable the absolute positioning bit 25 in application control code C4135/000 needs to be activated (*g_bLoadSetXAxisTrimmingTPPos* global variable = TRUE). As soon as the value in the code changes, the new position will be approached. With Code C3671/000 (*g_dnXAxisTrimmingTPPosOut* global variable) the user can see the actual trimming position. The speed and ramp times for this positioning operation are set in codes C3673/1-2 and C3672/2-3 respectively. To work with manual X trimming, operation usually has to be activated via bit 18 in application control code C4135/000 (*g_bUseXPositionTrimming* global variable = TRUE). When X trimming is activated, the value calculated for the distance between the mark sensor and the cam zero position is replaced by a value generated by X trimming. This means that X trimming can even be performed when mark correction is active. X trimming is only permitted in cutting operation.

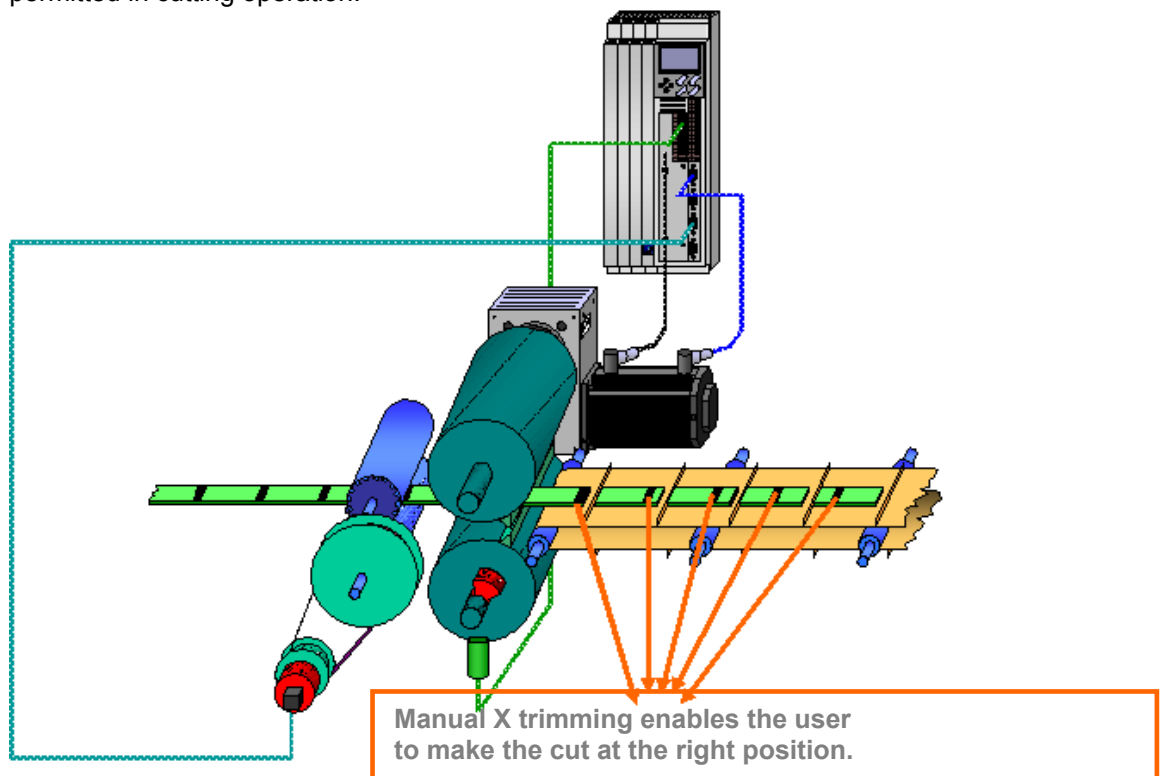


Figure27: Representation of manual X trimming

Cross Cutter

Features of the "Cross Cutter" prepared solution

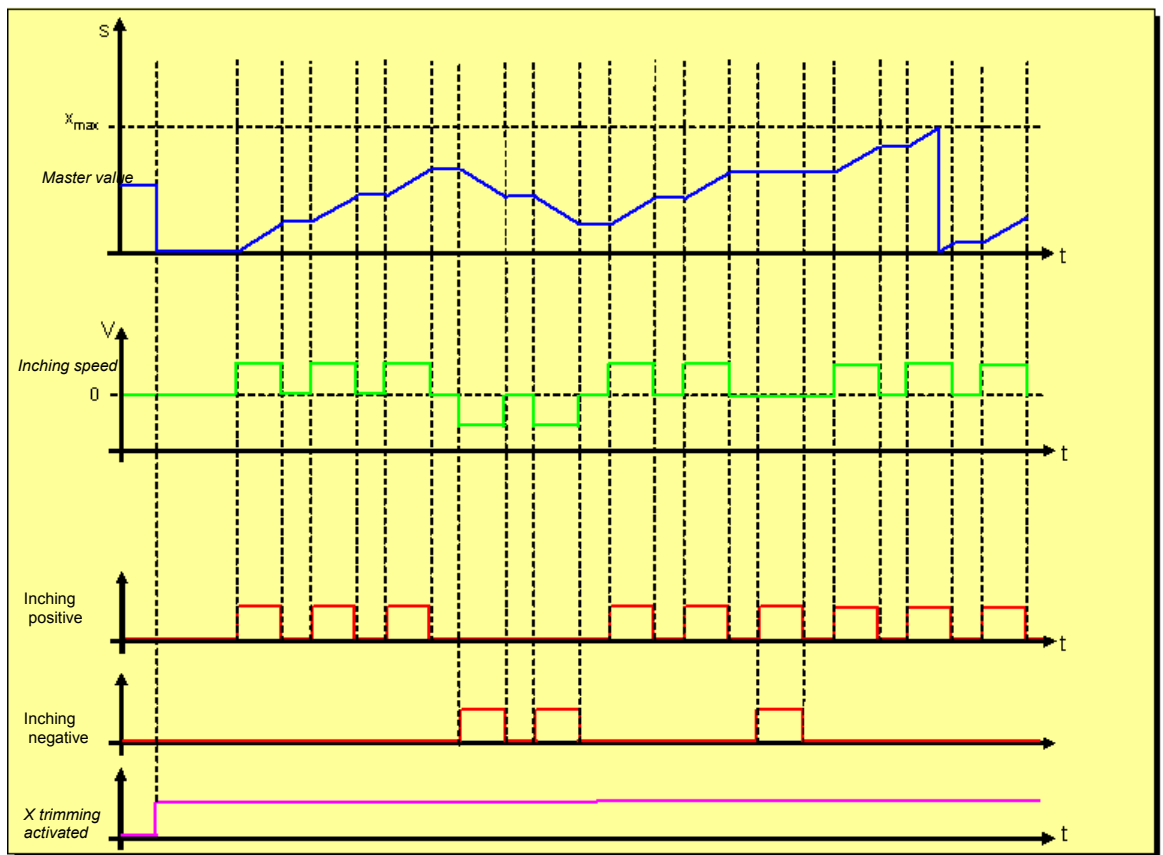


Figure 28: Schematic representation of signal generation for manual X trimming

Cross Cutter

Features of the "Cross Cutter" prepared solution

Variable names

Inputs		
Global variables (Variable type: VAR_GLOBAL)		
Name	Data type	Value/Meaning
g_bUseXPositionTrimming	BOOL	Activates X trimming and switches from the calculated mark sensor distance to the mark sensor distance specified by X trimming. This value is written in code C4135/000 bit 18. This value is displayed in code C4136/000 bit 18.
		TRUE X trimming activated.
		FALSE X trimming deactivated.
g_bXAxisTrimmingPosDirection	BOOL	Key this input to perform X trimming in a positive direction at the set speed. This value is written in code C4135/000 bit 19. This value is displayed in code C4136/000 bit 19.
		TRUE Perform X trimming in a positive direction.
		FALSE Do not perform X trimming.
g_bXAxisTrimmingNegDirection	BOOL	Key this input to perform X trimming in a positive direction at the set speed. This value is written in code C4135/000 bit 20. This value is displayed in code C4136/000 bit 20.
		TRUE Perform X trimming in a negative direction.
		FALSE Do not perform X trimming.
g_bLoadSetXAxisTrimmingTPPos	BOOL	Perform absolute positioning at the set speed. This value is written in code C4135/000 bit 25. This value is displayed in code C4136/000 bit 25.
		TRUE Perform absolute positioning in accordance with the value of variable g_dnSetXAxisTrimmingTPPos.
		FALSE Abort absolute positioning.
g_nXAxisTrimmingFactor_v	INT	Selection of the inching speed for X trimming in a positive and negative direction. This value is displayed in code C3678/000; this variable can be written to this code.
g_dnSetXAxisTrimmingTPPos	DINT	Target position of absolute positioning for X trimming in 0.0001[m_units]. This value is displayed in code C3674/000; this variable can be written to this code. The value is converted into increments internally. If you use this variable, to avoid inconsistencies, do not write variable g_dnSetXAxisTrimmingTPPos_p!
g_dnSetXAxisTrimmingTPPos_p	DINT	Target position of absolute positioning for X trimming in [incr]. If you use this variable, to avoid inconsistencies, do not write variable g_dnSetXAxisTrimmingTPPos!

Cross Cutter

Features of the "Cross Cutter" prepared solution

Outputs		
Global variables (Variable type: VAR_GLOBAL)		
Name	Data type	Value/Meaning
g_dnXAxisTrimmingTPPosOut	DINT	Displays the actual X trimming position in 0.0001[m_units]. This output indicates the mark sensor distance set by X trimming. Whenever X trimming is active, this value will serve as the mark sensor distance for mark correction. This value is displayed in code C3671/000.
g_nXAxisVelTrimmingFactor_v	INT	This variable contains the speed value generated for X trimming. It is added to the DFIN signal in order to effect an immediate change in the position value when using X trimming.

Inputs/Outputs		
Global variables (Variable type: VAR_GLOBAL)		
Name	Data type	Value/Meaning
g_dnXAxisTrimmingTPPosOut_p	DINT	This input/output indicates the mark sensor distance set by X trimming. Whenever X trimming is active, this value will serve as the mark sensor distance for mark correction. The variable is a retain variable (this means that once an X trimming value has been calculated, it will remain accessible even after mains switching).

Code	Possible settings:		Comment
	Default	Selection	
C3671/000	-	-	Display code: X trimming outputvalue scaled in 1.0000 m_units
C3672/002	20.0000	0.0100... {0.01 s} ... 130.0000	Acceleration ramp time for position trimming
C3672/003	20.0000	0.0100... {0.01 s} ... 130.0000	Deceleration ramp time for position trimming
C3673/001	200.0000	0 ... {1 rpm} ...15000	Speed for position trimming in a positive direction
C3673/002	200.0000	0 ... {1 rpm} ...15000	Speed for position trimming in a negative direction
C3674/000	0.0000	-214748.0000... {0.0001 m_units} ... 214748.0000	Selection of the absolute target trimming position
C3676/000	10.0000	0.000... {0.001 s} ... 999.999	Acceleration ramp time for inching trimming
C3677/000	10.0000	0.000... {0.001 s} ... 999.999	Deceleration ramp time for inching trimming
C3678/000	0	0 ... {1 rpm} ...15000	Selection of the inching speed
C4135/000 Bit 18	0	0 {1}	Application control word 18: Activates X trimming and switches from the calculated mark sensor distance to the mark sensor distance specified by X trimming.
			TRUE X trimming activated.
			FALSE X trimming deactivated.
C4135/000 Bit 19	0	0 {1}	Application control word bit 19: Key this bit to perform X trimming in a positive direction at the set speed.
			TRUE Perform X trimming in a positive direction.
			FALSE Do not perform X trimming.
C4135/000 Bit 20	0	0 {1}	Application control word bit 20: Key this bit to perform X trimming in a negative direction at the set speed.
			TRUE Perform X trimming in a negative direction.
			FALSE Do not perform X trimming.
C4150/000 Bit 30	-	-	Application status word bit 30: X position trimming is complete and the target position has been reached.
			TRUE The target position has been reached. Positioning is complete.
			FALSE The target position has not been reached. Positioning is not complete.

Cross Cutter

Features of the "Cross Cutter" prepared solution

Calculating and setting acceleration and deceleration times when using inching trimming (*g_bXAxisTrimmingPosDirection* and *g_bXAxisTrimmingNegDirection*)

The acceleration and deceleration times relate to a change in the output value from 0 to 100% (100% = 16384). The times to be set for the acceleration time T_{ir} and the deceleration time T_{if} can be calculated using the following formulae:

$$\text{acceleration } T_{ir} = t_{ir} \cdot \frac{100\%}{w2 - w1}$$

$$\text{deceleration } T_{if} = t_{if} \cdot \frac{100\%}{w2 - w1}$$

Following calculation, you will need to enter the values T_{ir} and T_{if} in the corresponding codes (C3676/000 = acceleration time T_{ir} ; C3677/000 = deceleration time T_{if}).

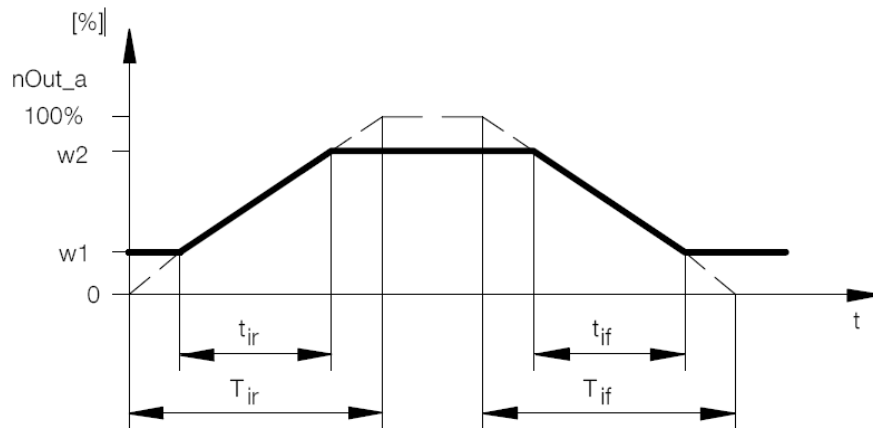


Figure 29: Graph for calculating acceleration and deceleration times when using inching trimming

Example: The inching speed is to be 50 rpm. This speed value needs to be reached within 1 sec. This results in the following values for T_{ir} and T_{if} when the formula below is applied:

$$\text{acceleration } T_{ir} = t_{ir} \cdot \frac{100\%}{w2 - w1} = 1 \text{ sec} \cdot \frac{15000}{50 - 0} = \underline{\underline{300 \text{ sec}}}$$

$$\text{deceleration } T_{if} = t_{if} \cdot \frac{100\%}{w2 - w1} = 1 \text{ sec} \cdot \frac{15000}{50 - 0} = \underline{\underline{300 \text{ sec}}}$$



Note!

100% corresponds to the value entered in code C0011/000. In the example calculation, a value of 15,000 rpm has been assumed.

Calculating and setting acceleration and deceleration times when using position trimming (C3674/000 = Selection of the absolute target trimming position)

The acceleration and deceleration times relate to 4000 rpm. The times to be set for the acceleration time T_{ir} and the deceleration time T_{if} can be calculated using the following formulae:

Cross Cutter

Features of the "Cross Cutter" prepared solution

$$\text{acceleration } T_{ir} = t_{ir} \cdot \frac{4000 \text{ rpm}}{C3673/001}$$

$$\text{deceleration } T_{if} = t_{if} \cdot \frac{4000 \text{ rpm}}{C3673/002}$$

Following calculation, you will need to enter the values T_{ir} and T_{if} in the specially designated code (C3672/002 = acceleration time T_{ir} ; C3672/003 = deceleration time T_{if} respectively).

Example: The forward and return speeds are to be 200 rpm. Each speed value needs to be reached within 1 sec. This results in the following values for T_{ir} and T_{if} when the formula below is applied:

$$T_{ir} = t_{ir} \cdot \frac{4000 \text{ rpm}}{C3673/001} = 1 \text{ sec} \cdot \frac{4000 \text{ rpm}}{200 \text{ rpm}} = \underline{\underline{20 \text{ sec}}}$$

$$T_{if} = t_{if} \cdot \frac{4000 \text{ rpm}}{C3673/002} = 1 \text{ sec} \cdot \frac{4000 \text{ rpm}}{200 \text{ rpm}} = \underline{\underline{20 \text{ sec}}}$$

Cross Cutter

Features of the "Cross Cutter" prepared solution

2.5.12 Reject gate control

A reject gate can be used to divert material to a different material flow direction in order, for example, to collect rejects in a waste container. The reject gate can be controlled using a digital signal. The signal is output dependent on the line speed (to ensure that the reject gate opens at an appropriate speed in relation to the prevailing line speed, see Figure 31). In the prepared solution, this signal is both permanently linked to digital output 3 (*DIGOUT_bOut3_b*) using 9300 Servo PLC or digital output 1 (*DIGOUT_bOut1_b*) using ECS and linked to bit 25 of application status code C4150/000 (*g_bOpenRejectGate* global variable). The reject gate should open as soon as this signal becomes active (*g_bOpenRejectGate* global variable = *TRUE*). Delays in the opening or closing of the reject gate can be compensated by entering a switching delay time (C3680/000). The distance between the reject gate and the cross cutter is defined in code C3679/000 (see Figure 30). There must only ever be up to one reject material block and one good material block in the space between the knife drum and the reject gate (see Figure 32). Reject gate control is always activated automatically whenever the cross cutter switches to reject operation. This is done in code C4137/000 by selecting operating mode 2 for reject operation (*g_wProfileSelector* global variable = 2).

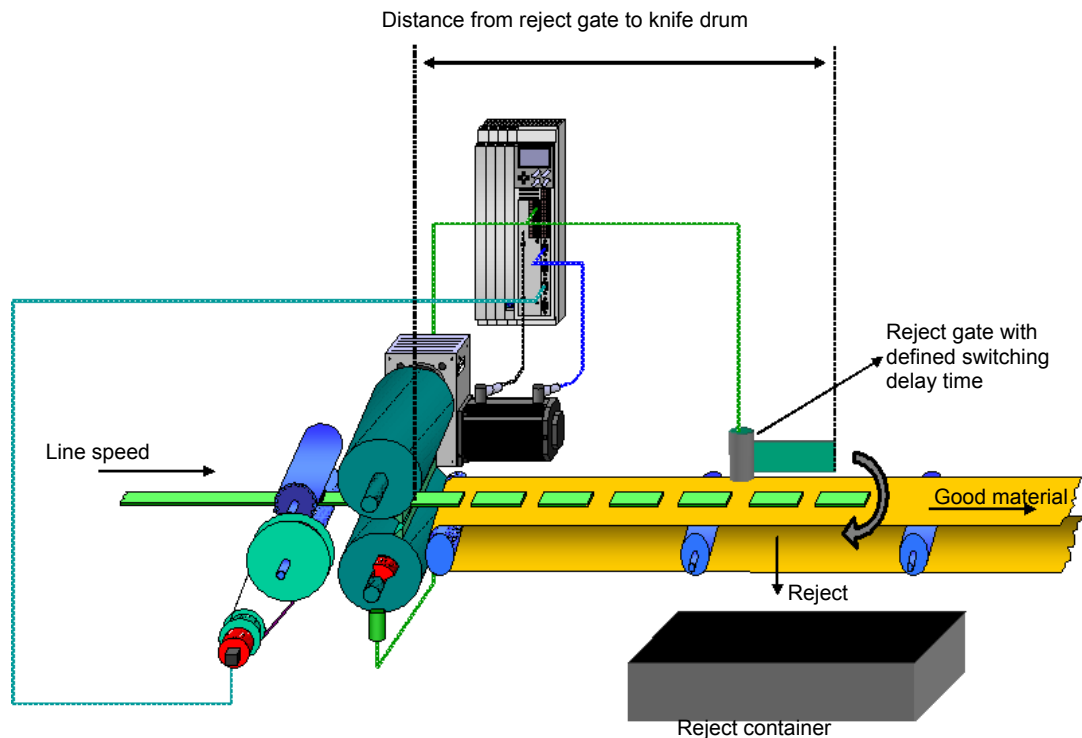


Figure 30: Representation of reject gate control

Cross Cutter

Features of the "Cross Cutter" prepared solution

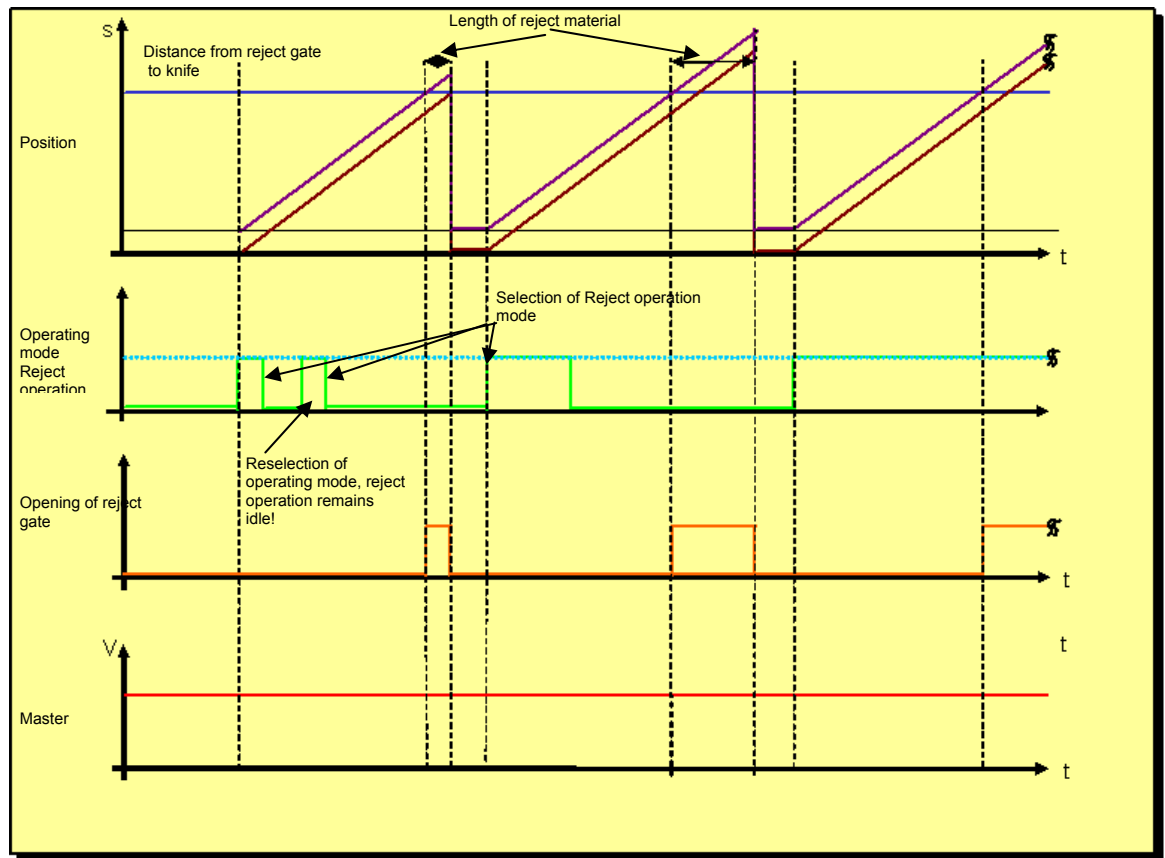


Figure 31: Schematic representation of signal generation for reject gate control

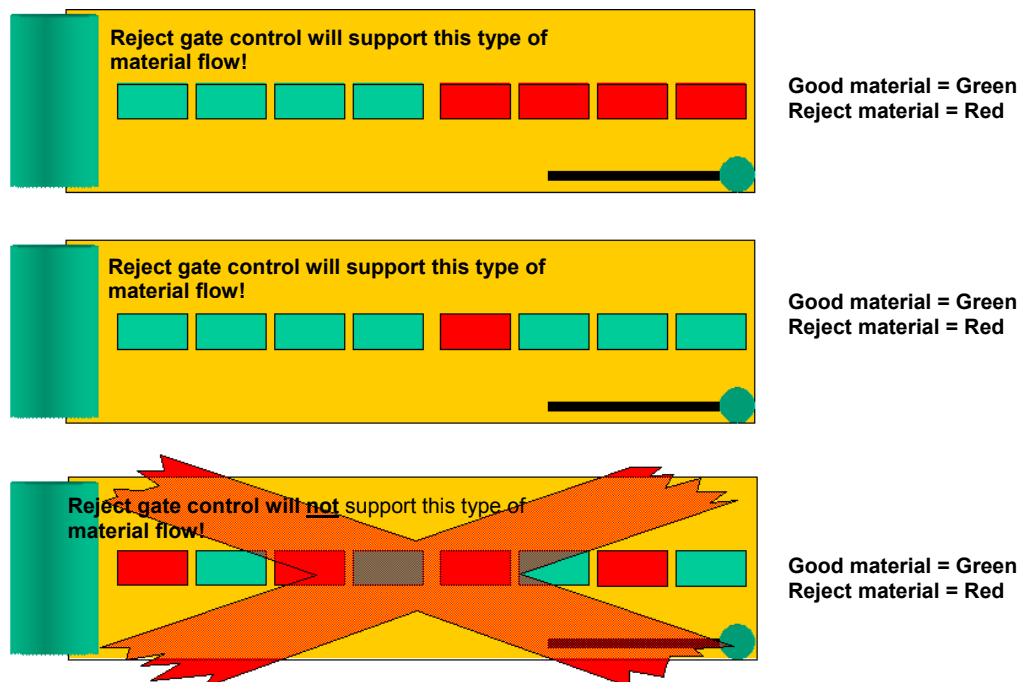


Figure 32: Material flows supported by the reject gate

Cross Cutter

Features of the "Cross Cutter" prepared solution

Variable names

Inputs		
Global variables (Variable type: VAR_GLOBAL)		
Name	Data type	Value/Meaning
g_wRejectGateSwitchingDwelltime	WORD	Selection of the switching dwell time (switching delay time) of the reject gate in ms. This value is displayed in code C3680/000; this variable can be written to this code.
g_dnDistanceRejectGate	DINT	Selection of the distance from the reject gate to the cutting position in 0.0001[m_units]. This value is displayed in code C3679/000; this variable can be written to this code. The value is converted into increments internally. If you use this variable, to avoid inconsistencies, do not write variable g_dnDistanceRejectGate_p!
g_dnDistanceRejectGate_p	DINT	Selection of the distance from the reject gate to the cutting position in [incr]. If you use this variable, to avoid inconsistencies, do not write variable g_dnDistanceRejectGate!

Outputs		
Global variables (Variable type: VAR_GLOBAL)		
Name	Data type	Value/Meaning
g_bOpenRejectGate	BOOL	This output is used to control the reject gate. This value is displayed in code C4150/000 bit 25. The variable is permanently linked to digital output 3 using 9300 Servo PLC or digital output 1 using ECS.
		TRUE Open the reject gate.
		FALSE Do not open the reject gate.

Code	Possible settings:		Comment
	Default	Selection	
C3679/000	0.0000	0.0000... {0.0001 m_units} ... 214000.0000	Distance from reject gate to cutting position
C3680/000	0	0 ... {1 ms} ... 65535	Switching dwell time (switching delay time) of reject gate
C4150/000 Bit 25	-	-	Application status word bit 25: Control of reject gate
			TRUE Open the reject gate.
			FALSE Do not open the reject gate.

Cross Cutter

Features of the "Cross Cutter" prepared solution

2.5.13 Calculating the print-mark-sensor distance

In order to be able to use mark correction, you need to define the exact distance between the mark sensor and the curve zero of the x axis. As illustrated previously in Chapter 2.2.2, the curve zero does not line up with the six o'clock position of the knife. This makes it very difficult to measure the distance between the print-mark sensor and the profile zero crossing. In order to make it easier for the user to determine the print-mark-sensor distance, this is calculated internally. The user simply has to tell the system the distance between the print-mark sensor and the six o'clock position on the knife by entering this value in code C3668/000. The value calculated for the distance between the print-mark sensor and the profile zero crossing is written automatically to global variable `g_dnXTpPos`.

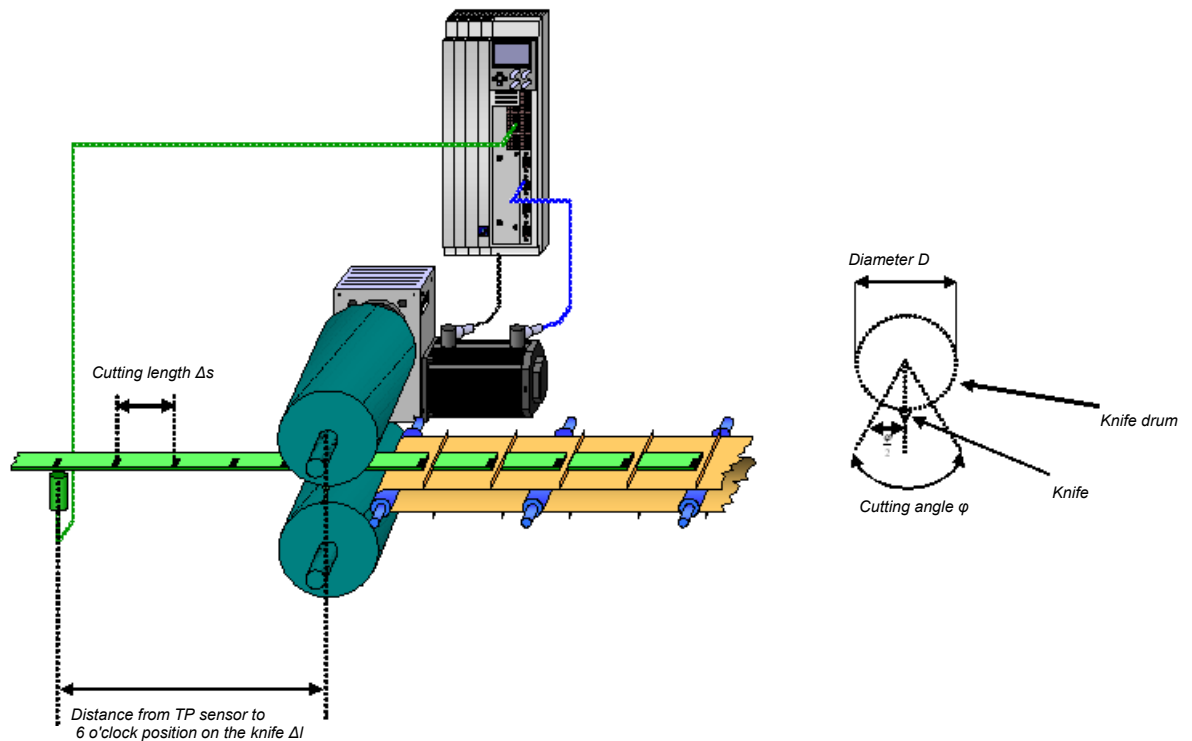


Figure 33: Calculation of the mark sensor distance

Cross Cutter

Features of the "Cross Cutter" prepared solution

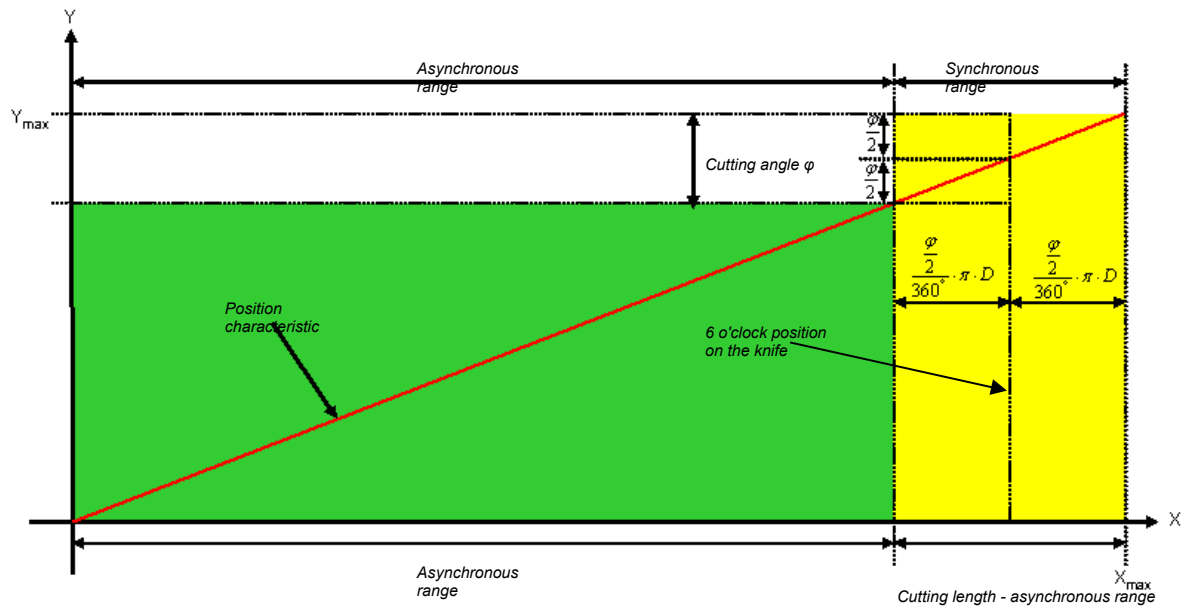


Figure 34: Graph showing how the mark sensor distance is calculated

Cross Cutter

Features of the "Cross Cutter" prepared solution

Variable names

Inputs		
Global variables (Variable type: VAR_GLOBAL)		
Name	Data type	Value/Meaning
g_dnMeasuredTPDistance	DINT	Distance measured from TP sensor to 6 o'clock position on the knife drum. The value is entered in 0.0001[m_units]. This value is displayed in code C3668/000; this variable can be written to this code.

Outputs		
Global variables (Variable type: VAR_GLOBAL)		
Name	Data type	Value/Meaning
g_bUseShiftedTP	BOOL	Use of the print-mark register. This value is displayed in code C4150/000 bit 28.
		TRUE The print-mark register needs to be used.
		FALSE The print-mark register does not need to be used.
g_nNumberOfKnives	INT	Number of knives mounted on the knife drum. This value is displayed in code C3690/000.
g_dnXTpPos	DINT	Print-mark-sensor position calculated. A modulo arithmetic operation is used to adjust this value by multiples of the cutting length so that a virtual touch-probe-sensor position can be calculated as close as possible to the knife. The value is scaled in 0.0001[m_units]. The value is converted into increments internally. If you use this variable, to avoid inconsistencies, do not write variable g_dnXTpPos_p!
g_dnXTpPos_p	DINT	Print-mark-sensor position calculated. A modulo arithmetic operation is used to adjust this value by multiples of the cutting length so that a virtual touch-probe-sensor position can be calculated as close as possible to the knife. The value is scaled in [incr]. The value is converted into increments internally. If you use this variable, to avoid inconsistencies, do not write variable g_dnXTpPos!
g_dnMultipleTPDistance	DINT	Calculated value indicating the distance of the print-mark sensor in integer multiples of the cutting length. The value is scaled in 0.0001[m_units]. You will need this value if you are using a print-mark register. The value is converted into increments internally. If you use this variable, to avoid inconsistencies, do not write variable g_dnMultipleTPDistance_p!
g_dnMultipleTPDistance_p	DINT	Calculated value indicating the distance of the print-mark sensor in integer multiples of the cutting length. The value is scaled in 1[incr]. You will need this value if you are using a print-mark register. If you use this variable, to avoid inconsistencies, do not write variable g_dnMultipleTPDistance!

Cross Cutter

Features of the "Cross Cutter" prepared solution

Code	Possible settings:		Comment
	Default	Selection	
C3668/ 000	370.0000	0.0000... {0.0001 ... 214000.0000 m_units}	Distance measured from TP sensor to 6 o'clock position on the knife drum.
C3690/ 000	-	-	- Number of knives mounted on the knife drum. Up to 6 knives can be mounted on the cross-cutter drum.
C4150/ 000 Bit 28	-	-	Application status word bit 28: Use of the print-mark register.
			TRUE Use the print-mark register.
			FALSE Do not use the print-mark register.

Cross Cutter

Features of the "Cross Cutter" prepared solution

2.5.14 Print-mark register

A print-mark register enables the user to mount the print-mark sensor further away from the knife. As soon as the distance between the location of the print-mark sensor and the end of the cut ($X_{\max}$ value of profile) exceeds the cutting length, the print-mark register will need to be used to make sure that the cut is always corrected to the right print-mark signal. However, the aim should be to mount the print-mark sensor as close as possible to the knife. The greater the distance between the print-mark sensor and the knife, the more changes in the material flow will go unnoticed, leading to cutting inaccuracies. Up to 16 print-mark signals can be managed in the print-mark register; these are then made available to the system at the relevant time as necessary (i.e. they are delayed by a corresponding value). The print-mark register is activated automatically by the prepared solution and receives all necessary information via internal signals. The user does not need to enter any values.

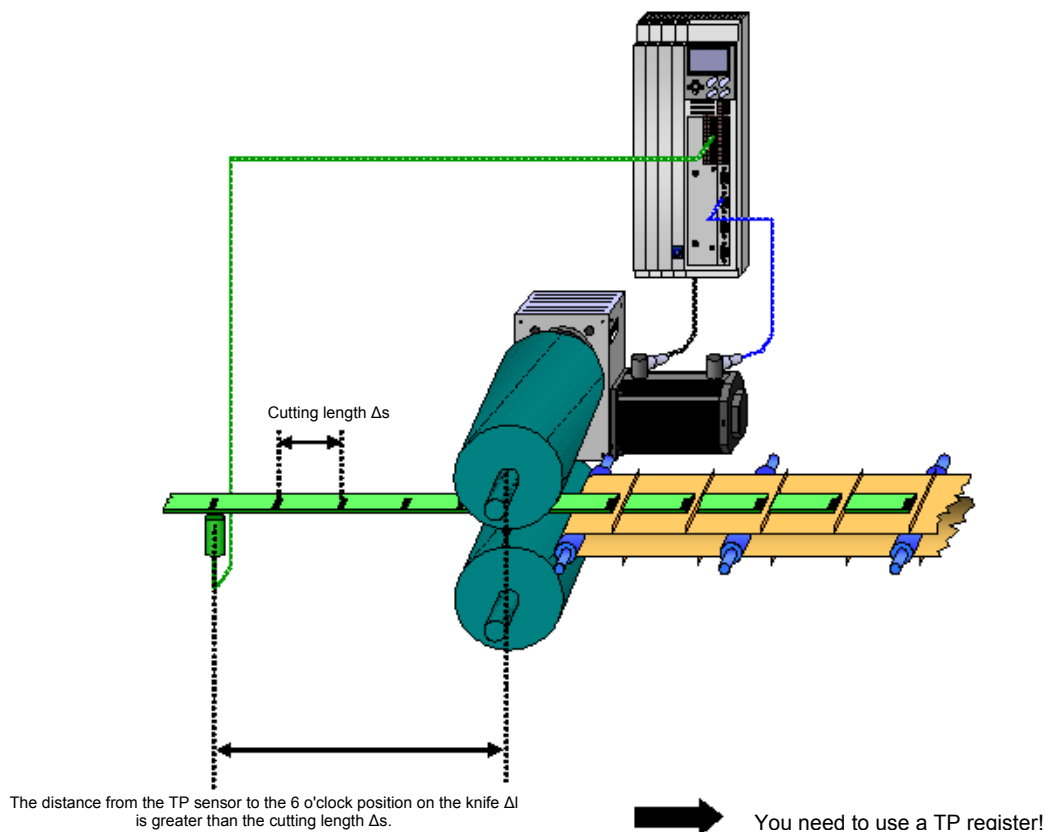


Figure 35: Using the print-mark register

Figure 35 shows that a print-mark register is always used whenever the distance from the print-mark sensor to the cross-cutter knife is greater than the set cutting length.

Cross Cutter

Features of the "Cross Cutter" prepared solution

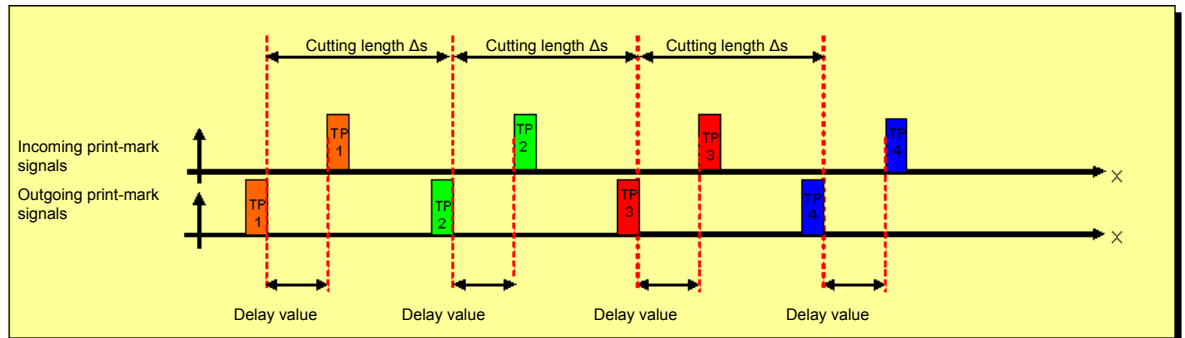


Figure 36: Graph of the print-mark register

Variable names

Inputs		
Global variables (Variable type: VAR_GLOBAL)		
Name	Data type	Value/Meaning
g_bUseShiftedTP	BOOL	Use of the print-mark register. This value is displayed in code C4150/000 bit 28.
		TRUE The print-mark register needs to be used.
		FALSE The print-mark register does not need to be used.
g_dnMultipleTPDistance	DINT	Calculated value indicating the distance of the print-mark sensor in integer multiples of the cutting length. The value is scaled in 0.0001[m_units]. The print-mark signal is delayed by this value. The value is converted into increments internally. If you use this variable, to avoid inconsistencies, do not write variable g_nMultipleTPDistance_p!
g_dnMultipleTPDistance_p	DINT	Calculated value indicating the distance of the print-mark sensor in integer multiples of the cutting length. The value is scaled in 1[incr.]. The print-mark signal is delayed by this value. If you use this variable, to avoid inconsistencies, do not write variable g_dnMultipleTPDistance!

Outputs		
Global variables (Variable type: VAR_GLOBAL)		
Name	Data type	Value/Meaning
g_bOverflowTPRegister	BOOL	This output indicates a print-mark register overflow.
		TRUE Print-mark register overflow.
		FALSE No print-mark register overflow.

Code	Possible settings:		Comment
	Default	Selection	
C4150/ 000 Bit 28	-	-	Application status word bit 28: Use of the print-mark register.
			TRUE Use the print-mark register.
			FALSE Do not use the print-mark register.

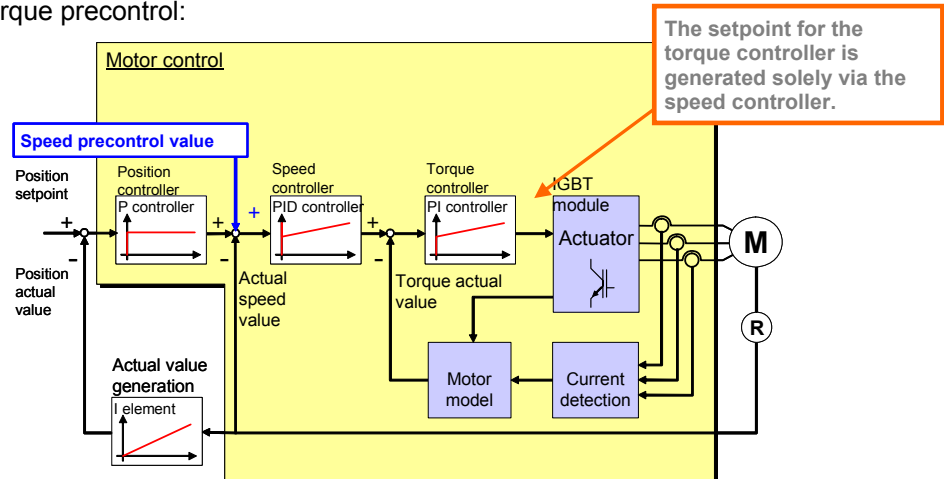
Cross Cutter

Features of the "Cross Cutter" prepared solution

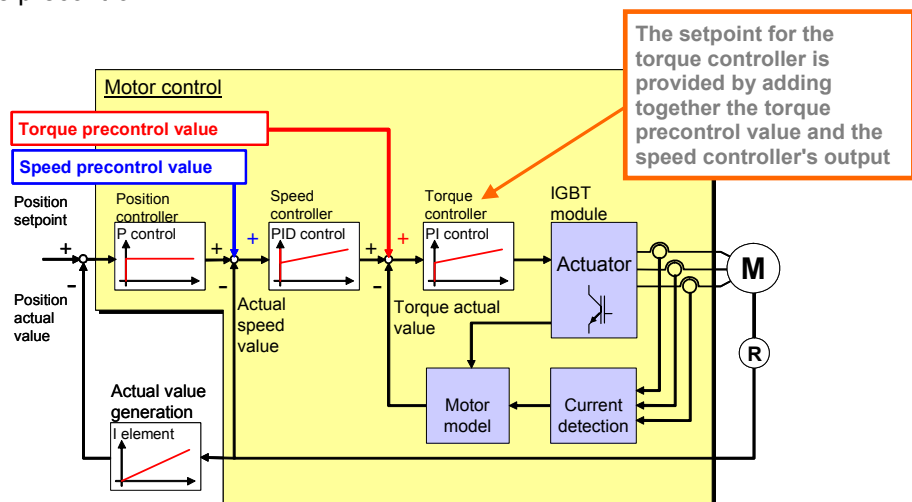
2.5.15 Torque precontrol

In accordance with the signal flow and the dynamic requirements of the drive, in addition to the generation of the speed precontrol value and the following error, torque precontrol can significantly improve the dynamic characteristics of the drive. This is achieved by relieving the load on the speed controller in system block *MCTRL* in accordance with the following principle:

a) Without torque precontrol:



b) With torque precontrol:



The more accurately the torque precontrol signal is able to map the actual torque, the more the load on the speed controller will be relieved and the better the dynamic characteristics of the drive will be.

Cross Cutter

Features of the "Cross Cutter" prepared solution

Generally, torque load on a drive comprises the following components:

$$M(t) = J \cdot d \frac{d\varphi(t)}{dt^2} + D_{\varphi} \cdot \frac{d\varphi(t)}{dt} + c_{\varphi} \cdot \varphi(t) + M_s(t)$$

where:

- $\varphi(t)$ = Angle of rotation of the drive²
- J = Rotary mass inertia
- D_{φ} = Rotary attenuation factor
- c_{φ} = Rotary spring force constant
- $M_s(t)$ = External load torque

The dynamic component in particular affects the accuracy of the motion of the drive. In the cross-cutter application, this component is determined essentially by the rotary acceleration/deceleration of the knife drum and its mass inertia. Therefore, as a first approximation, a signal proportionate to acceleration/deceleration can be used as a suitable torque precontrol signal. The second mathematical derivation of the function $s = f(x)$ produces a signal proportionate to acceleration/deceleration dependent upon the master value position x :

$$a(x) = d \frac{ds}{dx^2}$$

The torque component proportionate to acceleration is the product of the rotary acceleration and the (constant) mass inertia:

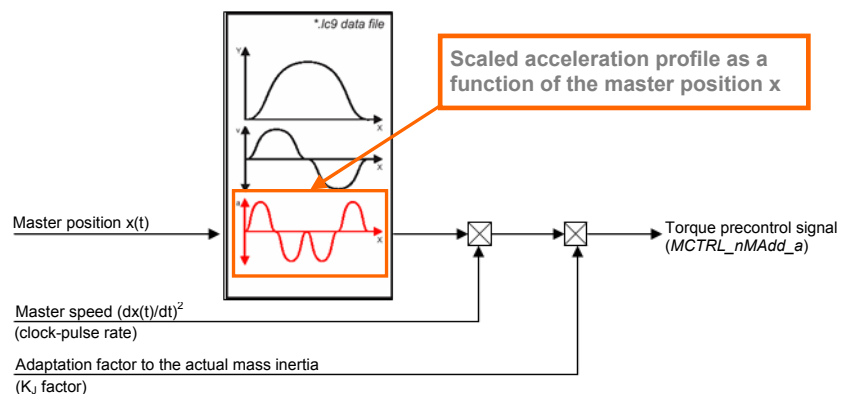
$$M(t) = K_J \cdot d \frac{d\varphi(x(t))}{dx^2} \cdot \left(\frac{dx(t)}{dt} \right)^2$$

Adaptation factor K_J for adaptation to the actual mass inertia of the drive system (usually calculated as an experimental value)

Scaled acceleration characteristic as a function of the master position $x(t)$

Taking into account the likelihood that the master position will change over

At constant master speed, a torque precontrol signal can be generated accordingly by multiplying the square of the master speed by the scaled acceleration curve and an adaptation factor (representing the constant mass inertia of the drive system). The signal flow must therefore look like this:



² The designation $s(t)$ is used generally for translatory and rotary systems. The designation $\varphi(t)$ is used specifically for rotary systems.

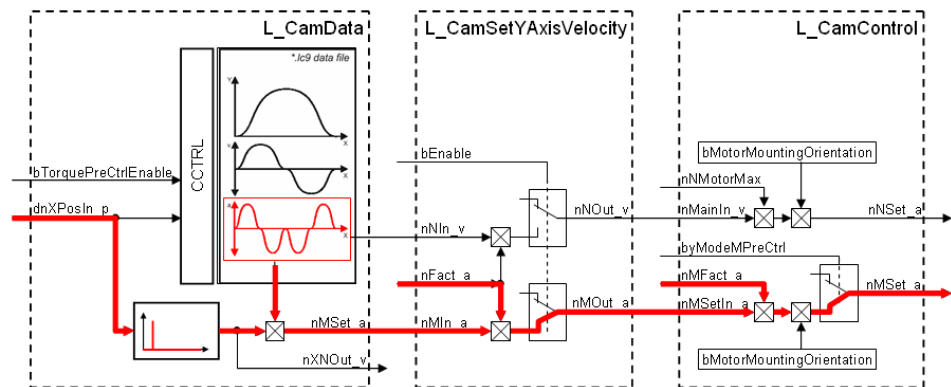
Cross Cutter

Features of the "Cross Cutter" prepared solution

The function blocks for calculating motion profiles for rotary cross cutters support the calculation of suitable scaled acceleration curves for generating the torque precontrol signal:

In addition to the project file, in which setpoint generation, signal flow, control of the *MCTRL* system block for drive control, etc. are defined, where applications use the Cam software package, an additional file (LC9 file) is required. This file contains the motion and scaled acceleration precontrol profiles in the form of tables of values. Scaled acceleration precontrol profiles are calculated automatically by the *CrossCutterMotionCalculation* function block without the user having to make further entries.

Signal flow between function blocks for the generation of the torque precontrol signal looks like this:



Variable names

Inputs		
Global variables (Variable type: VAR_GLOBAL)		
Name	Data type	Value/Meaning
g_bTorquePrecontrolEnable	BOOL	Enable torque precontrol. This value is displayed in code C3666/000; this variable can be written to this code.
		TRUE Torque precontrol is enabled.
		FALSE Torque precontrol is disabled.
g_nFactorInertia_a	INT	Adaptation factor for torque precontrol in % 16384 = 100% This value is displayed in code C3667/000; this variable can be written to this code.
g_nTorquePreAdd_a	INT	Setvalue of the external torque precontrol in % 16384 = 100% This value is displayed in code C3639/000; this variable can be written to this code.

Cross Cutter

Features of the "Cross Cutter" prepared solution

Code	Possible settings:		Comment
	Default	Selection	
C3639/ 000	0.0	-200.0... {1%} ... 200.0	Setvalue of the external torque precontrol
C3640/ 000	1	0 {1} 5	function external torque precontrol
			0 OUT = IN1
			1 OUT = IN1 + IN2
			2 OUT = IN1 - IN2
			3 OUT = IN1 * IN2
			4 OUT = IN1 / IN2
			5 OUT = IN1/(100-IN2)
			IN1= g_nFactorInertia_a (C3667/000) IN2= g_nTorquePreAdd_a (C3639/000)
C3666/ 000	0	0 {1} 1	Enable torque precontrol.
			TRUE Torque precontrol is enabled.
			FALSE Torque precontrol is disabled.
C3667/ 000	10.0	-200.0... {1%} ... 200.0	Adaptation factor for torque precontrol

2.5.16 Evaluation of absolute value encoders at load and drive end

Absolute value encoders can be evaluated for the prepared solution. There are several reasons for this. Firstly, due to sincos evaluation, evaluation on absolute value encoders is very much higher and therefore more accurate than on the majority of incremental encoders. Furthermore, there is an assurance that the position of the knife will always be known, even after mains switching. This means that homing does not have to be performed. However, mounting location is also a decisive factor in respect of the accuracy of the encoder system used. Depending on torsion and gearbox backlash, it can be advantageous to mount the encoder on the load side, as this enables inaccuracies due to torsion and gearbox backlash to be disregarded.

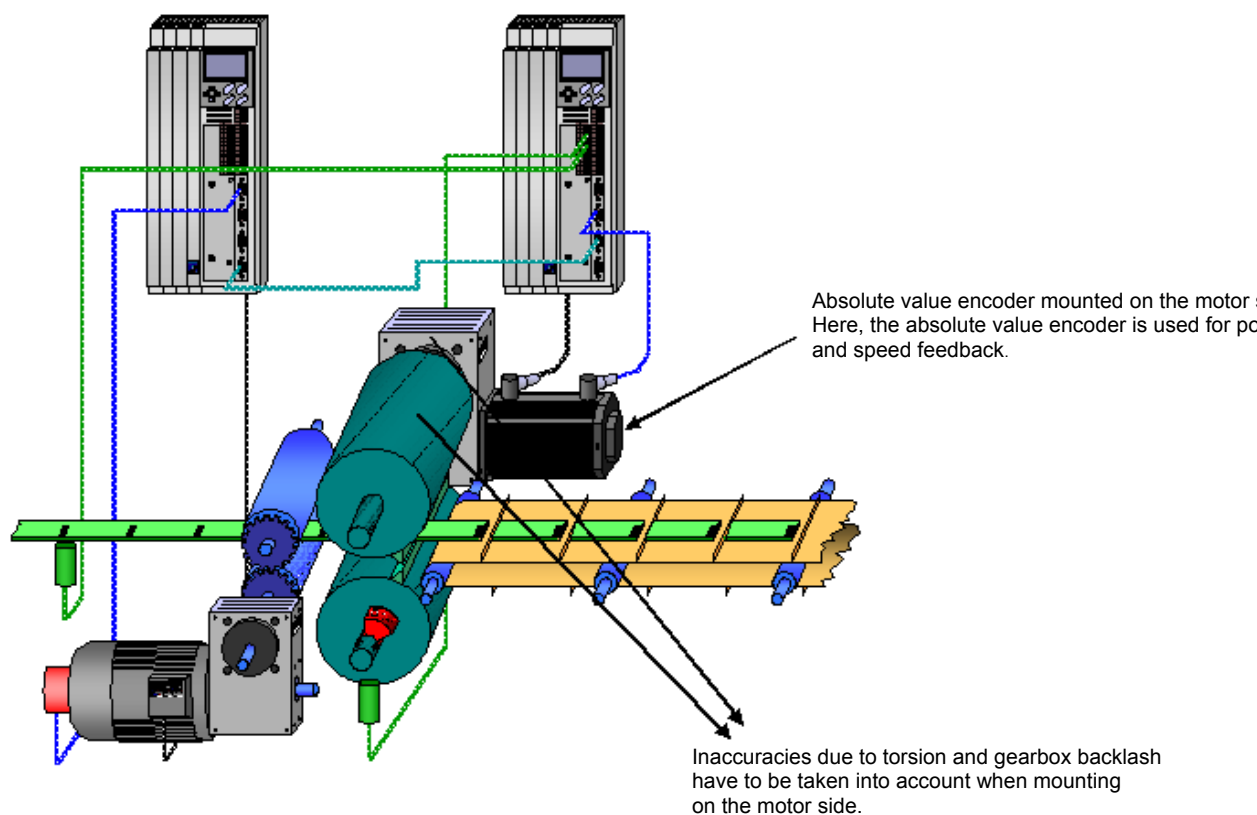


Figure 37: Absolute value encoder mounted on the motor side

Cross Cutter

Features of the "Cross Cutter" prepared solution

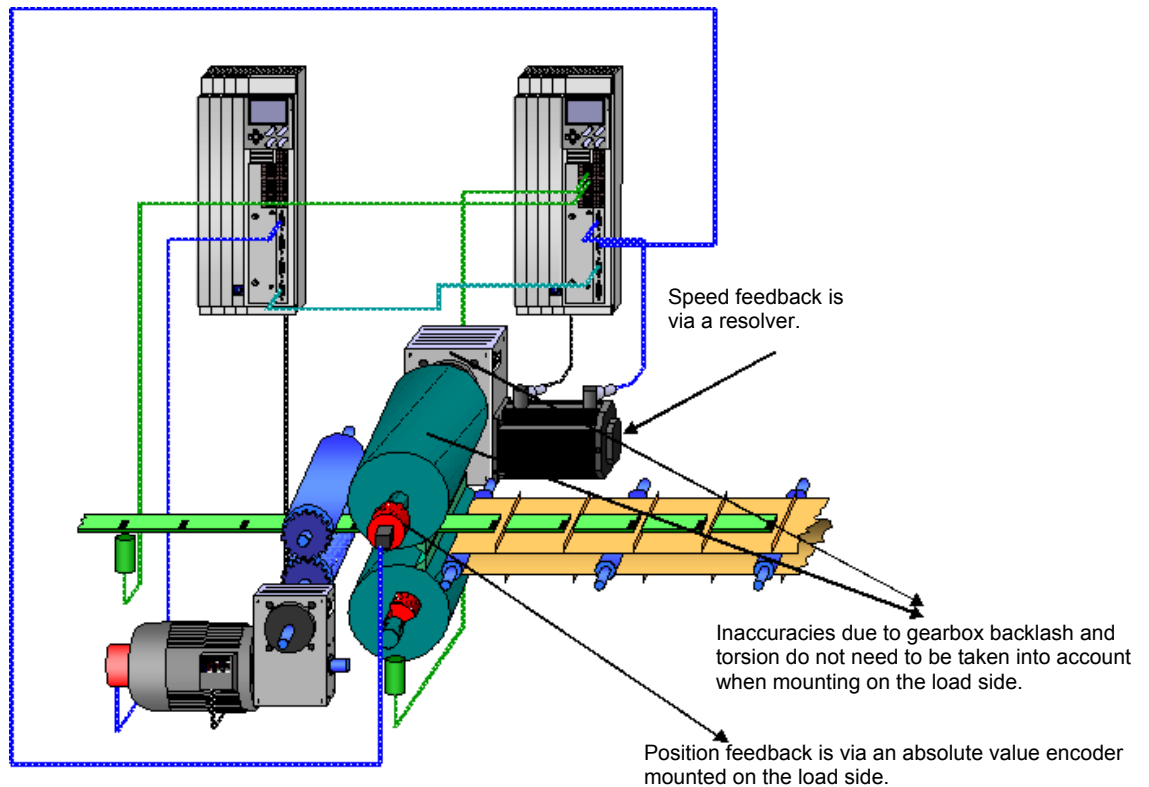


Figure 38: Absolute value encoder mounted on the load side

Determining scaling factors for position feedback mounted on the load side:

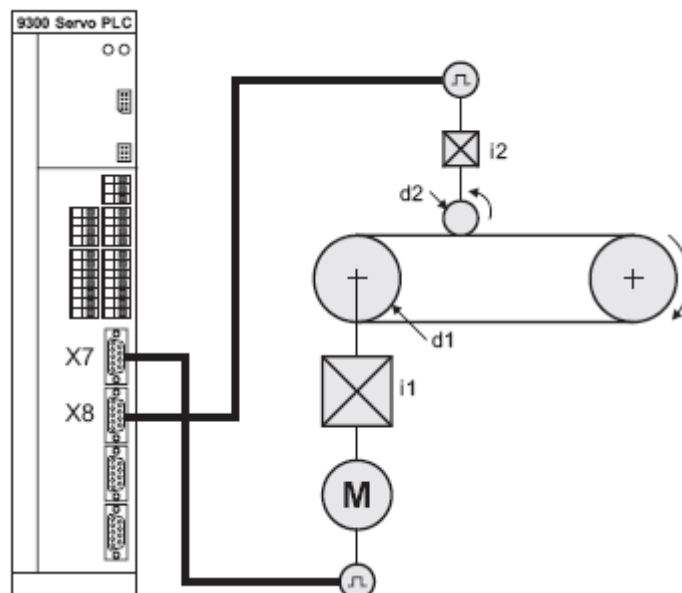


Figure 39: Determining scaling factors

Cross Cutter

Features of the "Cross Cutter" prepared solution

The following formula can be used to calculate scaling factors for load-side mounting of position feedback (see Figure 39):

$$\frac{C3637 / 000}{C3638 / 000} = \frac{vk2 \cdot i1}{vk1 \cdot i2}$$

$vk1 = d1 \cdot \pi$ = Vorschubkonstante Drehzahlrückführung

$vk1 = d2 \cdot \pi$ = Vorschubkonstante Lagerückführung

$i1$ = Getriebefaktor Drehzahlrückführung

$i2$ = Getriebefaktor Lagerückführung



Caution!

For applications with position feedback on the load side, please note that the position encoder must rotate in the same direction as the speed sensor! If there is not the same rotating direction of the feedback systems the rotating direction must be changed with codenumber C3637 by setting a negativ value.



Caution!

If you use the ECS Sytem: With the Parameterisable variant it is not possible to use an absolute encoder because this encoder type uses the same input X8 as the frequency cuppling via DFIN. If an absolute encoder is necessary in the application the user has to use the programmable version of the prepared solution Cross Cutter and have to handle the line feedback for example via CAN on its own.

Variable names

Code	Possible settings:		Comment
	Default	Selection	
C0490/ 000	0	0.... {1}4	Feedback system for the position controller
			0 Resolver at X7
			1 TTL encoder at X8
			2 Sin/cos encoder at X8
			3 ST absolute value encoder at X8
			4 MT absolute value encoder at X8
C0495/ 000	0	0.... {1}4	Feedback system for the speed controller
			0 Resolver at X7
			1 TTL encoder at X8
			2 Sin/cos encoder at X8
			3 ST absolute value encoder at X8
			4 MT absolute value encoder at X8
C3637/ 000	1	-32767 {1} ... 32767	Gearbox (numerator) position feedback It is not allowed to set the value Zero!

Cross Cutter

Features of the "Cross Cutter" prepared solution

Code	Possible settings:		Comment
	Default	Selection	
C3638/ 000	1	1... {1} ... 32767	Gearbox (denominator) position feedback

Cross Cutter

Features of the "Cross Cutter" prepared solution

2.5.17 Evaluation of the master value signal

The incoming master value is expected at the DFIN Input (X9 for Servo PLC or X8 for ECS). The user can invert and even simulate the incoming master value. The simulation mode can be helpful during installation and start-up, in order to test the system. However, the set value in simulation mode is given directly to the system, there is no acceleration or deceleration ramp included. For this reason big jumps in the set value should be avoided because otherwise there is the risk of a following error.



Caution!

The master value applied on the drive controller via the DFIN input must not be negative! If the master value is negative, it must be inverted. The user can check code C0426 to determine whether or not the master value is negative. This code displays the input value of the DFIN input.



Caution!

If you use the ECS Sytem: With the Parameterisable variant it is not possible to use an absolute encoder because this encoder type uses the same input X8 as the frequency coupling via DFIN. If an absolute encoder is necessary in the application the user has to use the programmable version of the prepared solution Cross Cutter and have to handle the line feedback for example via CAN on its own.

Variable names

Inputs		
Global variables (Variable type: VAR_GLOBAL)		
Name	Data type	Value/Meaning
g_bUseTestsignal	BOOL	Activates master value simulation. This value is written in code C4135/000 bit 10. This value is displayed in code C4136/000 bit 10.
		TRUE Master value is simulated via code C3681/000 (g_nTestsignalDFIN_v).
		FALSE Master value is used by DFIN input.
g_bInvertDFINSignal	BOOL	Inversion of DFIN signal This value is written in code C3635/000.
		TRUE Invert DFIN signal.
		FALSE No inversion.
g_nTestsignalDFIN_v	INT	Selection of the simulation value for DFIN simulation. This value is displayed in code C3681/000; this variable can be written to this code.

Cross Cutter

Features of the "Cross Cutter" prepared solution

Code	Possible settings:		Comment
	Default	Selection	
C3635/ 000	0	0 ... {1} ...1	Inversion of DFIN signal
			TRUE Invert DFIN signal.
			FALSE No inversion.
C3636/ 000	0	0 ... {1} ...1	Inversion of motor mounting direction
			TRUE Motor left mounting (inverted)
			FALSE Motor right mounting (not inverted)
C3681/ 000	0	0 ... {1 rpm} ...15000	Selection of the simulation value for DFIN simulation.
C4135/ 000 Bit 10	0	0 {1} 1	Application control word bit 10: Activates master value simulation.
			TRUE Master value is simulated via code C3681/000 (<i>g_nTestsignal/DFIN_v</i>).
			FALSE Master value is used by DFIN input.

Cross Cutter

Installing and starting up the "CrossCutter" prepared solution

3 Installing and starting up the "CrossCutter" prepared solution

3.1 Required components

Lenze hardware

Product	Type designation	Hardware version	Firmware version
PC system bus adapter	EMF2177IB	1.3 or higher	1.7 or higher
Handheld terminal (XT keypad)	EMZ9371BC	-	-
Digital frequency coupling		-	-
9300ET ServoPLC	EVS93xxET	7A or higher	7.0 or higher
or			
Application ECSxA	ECSxA	4B or higher	7.4 or higher

Lenze software

Product	Type designation	Version
Global Drive Loader (GDLoader)	(Freeware)	3.0 or higher
Drive PLC Developer Studio (DDS)	ESP-DDS1-P	2.3 or higher
Global Drive Control (GDC)	ESP-GDC2	4.10 or higher
CamDesigner Basic	ESP-CAM1-B	2.3 or higher
SWP CAM libraries	ESP-SPAC-CAM1	2.0 SP1

Cross Cutter

Installing and starting up the "CrossCutter" prepared solution

3.2 Installing the Cross Cutter prepared solution

A CD containing all necessary data is provided (see Chapter 1.2). An installation wizard installs the data on the PC automatically. The user is guided step for step through the installation by the installation wizard.

Using the programmable variant of the prepared solution cross cutter the user is asked for a password by starting the program master *CrossCutter_SPLC_Vxxxxxx.lpc* or *CrossCutter_ECS_Vxxxxxx.lpc*. The user has permission for Level 2, here you just have to press the OK Button and for Level 1, the password is lenze.

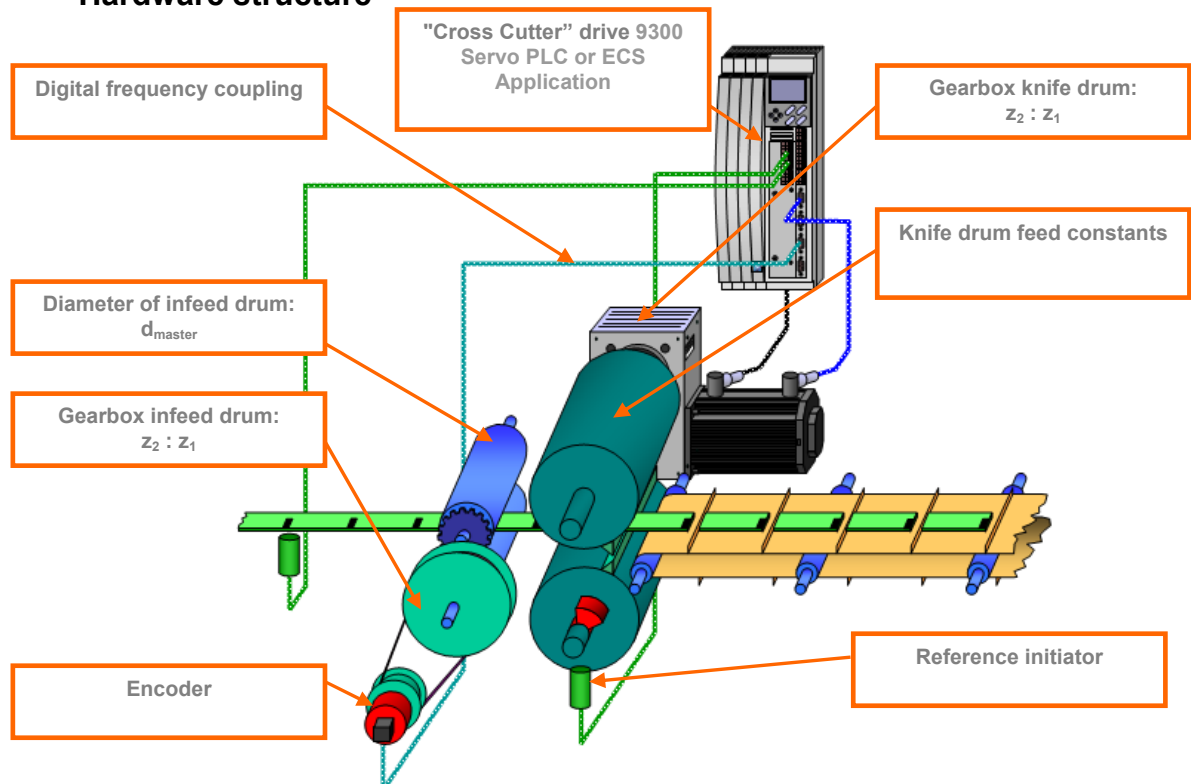


TIP!

Password Level 2: without typing in any password press the OK Button.

Password Level 1: lenze

3.3 Hardware structure



Cross Cutter

Installing and starting up the "CrossCutter" prepared solution

3.3.1 Wiring the control terminals

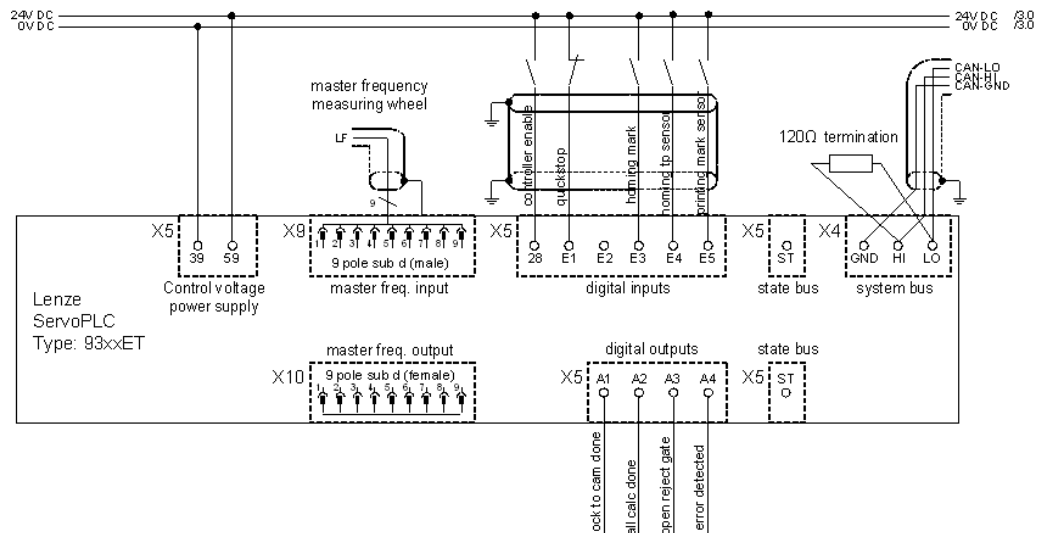


Figure 40: Control terminal wiring plan for 9300 Servo PLC

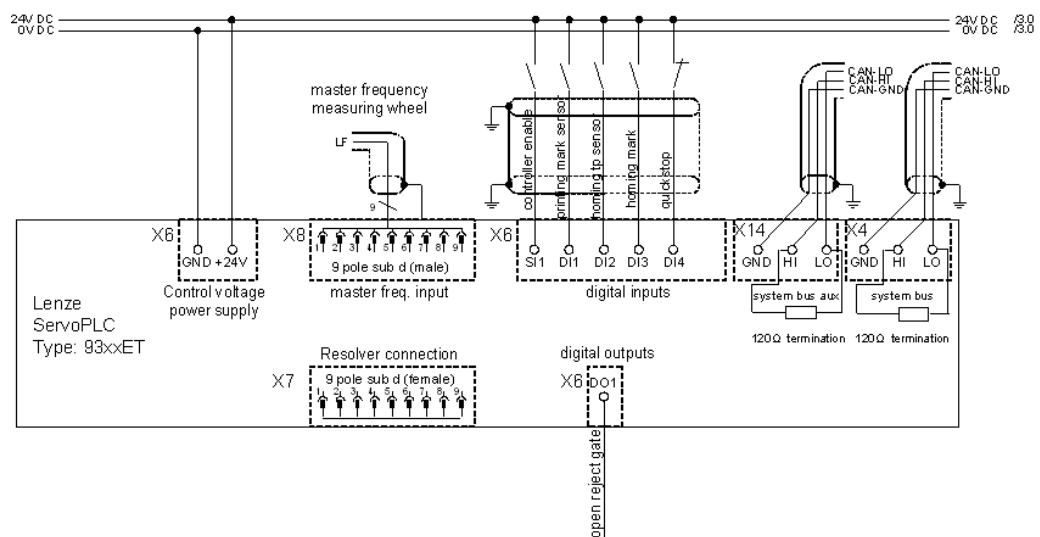


Figure 41: Control terminal wiring plan for ECS



Caution!

Digital input *DIGIN_bln1_b* using 9300 Servo PLC or *DIGIN_bln4_b* using ECS is inverted via the code initialisation values (see Chapter 6.2.2) in order to achieve wire-break protection (0 V = QSP).



Caution!

The connection plan above shows the wiring that is fundamentally required specifically in order for the prepared solution to function correctly.

Cross Cutter

Installing and starting up the "CrossCutter" prepared solution



Caution!

The basic wiring (e.g. connection to the mains and motor, connection of the feedback system) is not illustrated in this chapter. Please refer to Chapter 4 of the Operating Instructions supplied with the device for more information about basic wiring (this document can also be downloaded from Lenze's homepage at www.lenze.de).

Cross Cutter

Installing and starting up the "CrossCutter" prepared solution

3.4 Setting the parameters of the prepared solution

3.4.1 Control/Status interface with master control

In the parameterisable variant, the control and status interface is preconfigured. An overview of signal assignment and the options for reconfiguring (multiplexing) the control/status signals appears below.

Control interface

		Meaning
Word 0	System control word	Bit 00: Dependent on the setting in code C4000/000 Bit 01: Dependent on the setting in code C4000/000 Bit 02: Dependent on the setting in code C4000/000 Bit 03: Dependent on the setting in code C4000/000 Bit 04: Dependent on the setting in code C4000/000 Bit 05: Dependent on the setting in code C4000/000 Bit 06: Dependent on the setting in code C4000/000 Bit 07: Dependent on the setting in code C4000/000 Bit 08: Dependent on the setting in code C4000/000 Bit 09: Dependent on the setting in code C4000/000 Bit 10: Dependent on the setting in code C4000/000 Bit 11: Dependent on the setting in code C4000/000 Bit 12: Dependent on the setting in code C4000/000 Bit 13: Dependent on the setting in code C4000/000 Bit 14: Dependent on the setting in code C4000/000 Bit 15: Dependent on the setting in code C4000/000 C4000/000 = 0: The <i>DCTRL_wAIF1Ctrl</i> and <i>DCTRL_wCAN1Ctrl</i> system variables are always set to zero. C4000/000 = 1, 2, 3: The drive control word is copied to <i>DCTRL_wCAN1Ctrl</i>; <i>DCTRL_wAIF1Ctrl</i> is set to zero. C4000/000 = 4, 5, 6: The drive control word is copied to <i>DCTRL_wAIF1Ctrl</i>; <i>DCTRL_wCAN1Ctrl</i> is set to zero.
Word 1	Data word 1	Control word: Selection of operating mode: 1: Cutting mode 2: Reject mode 3: Test mode 4: Asynch. Mode
Word 2	Application control word	Bit 00: Start/stop homing, only in conjunction with controller enable (<i>g_bHomingStart</i>) Bit 01: Set home position (<i>g_bHomePositionSet</i>) Bit 02: Reset home position (<i>g_bHomePositionReset</i>) Bit 03: Manual jog in positive direction (CW) (<i>g_bManualJogPos</i>) Bit 04: Manual jog in negative direction (CCW) (<i>g_bManualJogNeg</i>) Bit 05: Perform positioning (<i>g_bMotionStart</i>) Bit 06: Activate external user error (<i>g_bExtUserError</i>) Bit 07: Reset error (<i>g_bErrorReset</i>) Bit 08: Inhibit controller (<i>g_bControllerInhibit</i>) Bit 09: QSP (<i>g_bUserQuickstop</i>) Bit 10: Activate the DFIN test signal to simulate the DFIN signal (<i>g_bUseTestSignal</i>) Bit 11: Reset the cutting counter for cutting operation (<i>g_bResetCuttingProfileCounter</i>) Bit 12: Reset the cutting counter for reject operation (<i>g_bResetRejectProfileCounter</i>) Bit 13: Reset the cutting counter for test operation (<i>g_bResetTestProfileCounter</i>) Bit 14: Start and stop selected cross-cutter operating mode (<i>g_bStartStopOperation</i>) Bit 15: Activate the variable to accept a newly calculated cutting length immediately and work with the new cutting length (<i>g_bWorkWithNewCuttingLength</i>) Bit 16: Activate the variable to accept a newly calculated reject length immediately and work with the new cutting length (<i>g_bWorkWithNewRejectLength</i>) Bit 17: Activate the variable to accept a newly calculated test length immediately and work with the new cutting length (<i>g_bWorkWithNewTestLength</i>) Bit 18: Activate X trimming and switch from the calculated mark sensor distance to the mark sensor distance specified by X trimming (<i>g_bUseXPositionTrimming</i>) Bit 19: Key this input to perform X trimming in a positive direction at the set speed (<i>g_bXAxisTrimmingPosDirection</i>) Bit 20: Key this input to perform X trimming in a negative direction at the set speed (<i>g_bXAxisTrimmingNegDirection</i>) Bit 21: Enable print-mark synchronisation (<i>g_bXTPEnable</i>) Bit 22: Enable ramp generator for print-mark synchronisation (<i>g_bRfqXTPEnable</i>) Bit 23: Reset counter for number of print marks located outside the window (<i>g_bResetCounterTPOutOfWindow</i>) Bit 24: Abort cross-cutter operation (<i>g_bAbortCrossCutter</i>) Bit 25: Execute X trimming absolute positioning (<i>g_bLoadSetXAxisTrimmingTPPos</i>) Bit 26: (not assigned) Bit 27: (not assigned) Bit 28: (not assigned) Bit 29: (not assigned) Bit 30: (not assigned) Bit 31: (not assigned)
Word 3		

Cross Cutter

Installing and starting up the "CrossCutter" prepared solution

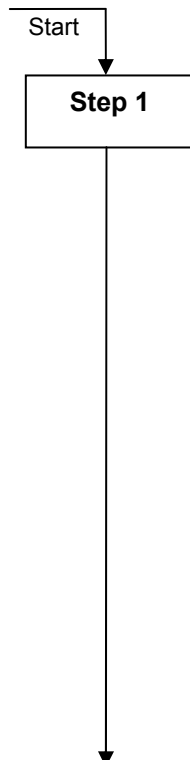
The signal sources for the data words (designated *Word 0*, *Word 1*, *Word 2* and *Word 3*) are dependent on the operating mode in C4000/000:

Signal source	Word 0	Word 1	Word 2	Word 3
C4000/000 = 0	C0135/000	C4137/000	C4135/000	
C4000/000 = 1	CAN1_wDctrlCtrl	CAN1_nlnW1_a	CAN1_nlnW2_a	CAN1_nlnW3_a
C4000/000 = 2	CAN2_nlnW1_a	CAN2_nlnW2_a	CAN2_nlnW3_a	CAN2_nlnW4_a
C4000/000 = 3	CAN3_nlnW1_a	CAN3_nlnW2_a	CAN3_nlnW3_a	CAN3_nlnW4_a
C4000/000 = 4	AIF1_wDctrlCtrl	AIF1_nlnW1_a	AIF1_nlnW2_a	AIF1_nlnW3_a
C4000/000 = 5	AIF2_nlnW1_a	AIF2_nlnW2_a	AIF2_nlnW3_a	AIF2_nlnW4_a
C4000/000 = 6	AIF3_nlnW1_a	AIF3_nlnW2_a	AIF3_nlnW3_a	AIF3_nlnW4_a

The illustration of the 32-bit application control word (*Word 3* and *Word 4*) on the previous page shows the factory-set default configuration. Users can also adapt the configuration of the application control word in accordance with their individual needs by setting the values of codes C4010/001 to C4010/032 accordingly.

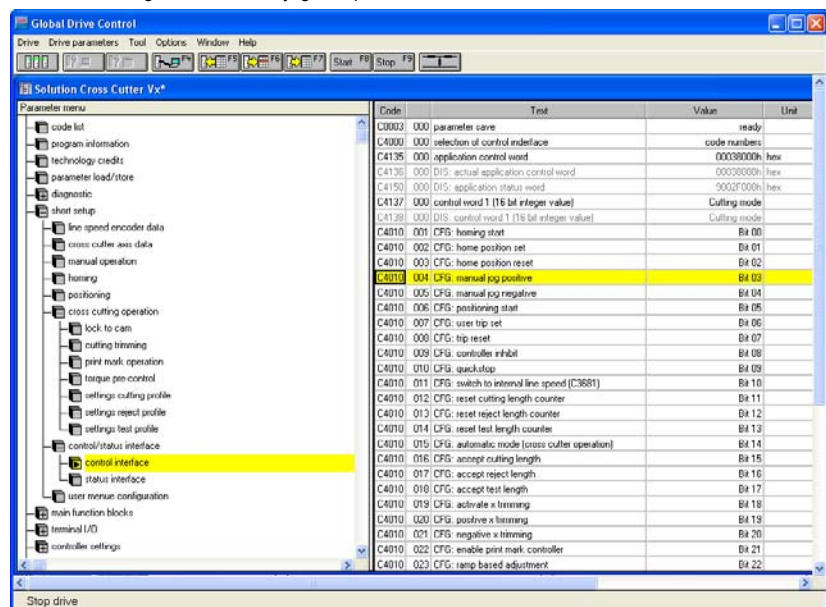
An example of how the **GDC** parameterisation software can be used to reconfigure the application control word bit by bit appears below:

You will find the code in the complete code list or in the menu (Short Setup → Control/Status Interface → Control Interface):



Select code C4010/00x which is assigned to the function to be configured.

Example: The control signal for manual jog in a positive direction is defined in code C4010/004:



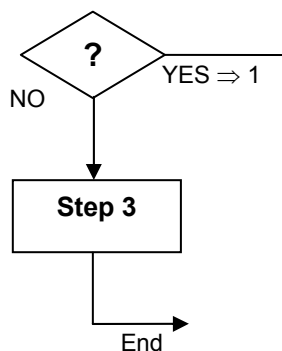
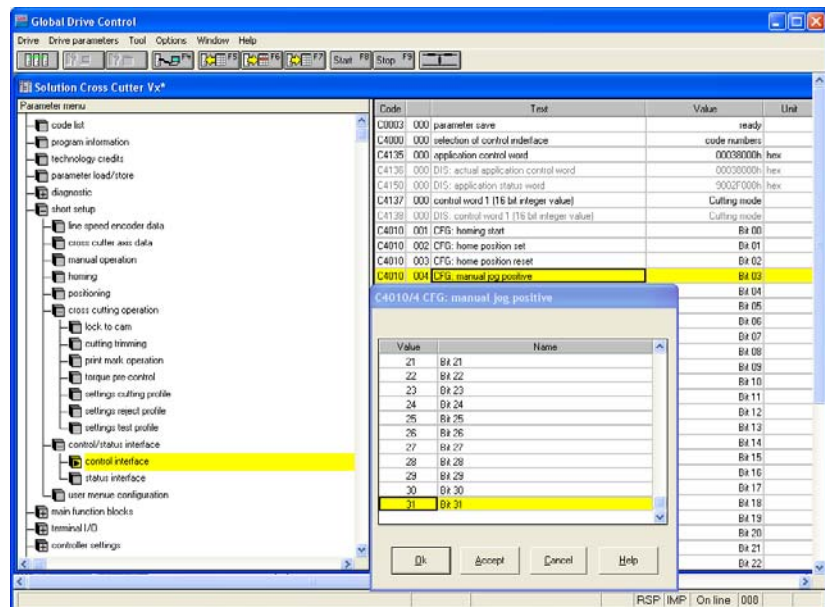
Cross Cutter

Installing and starting up the "CrossCutter" prepared solution

Step 2

Click on the code and select the required control bit.

Example: Bit 31 is to be used to control manual jog in a positive direction on the cross-cutter axis. Set C4010/004 = 31 accordingly and confirm your entry with Ok.



To reconfigure other control bits, proceed as described in Step 1.

Save the modified settings to non-volatile memory with C0003/000 = 1.

Cross Cutter

Installing and starting up the "CrossCutter" prepared solution

Status interface

Meaning		
Bit 00: Dependent on the setting in code C4000/000	System status word	Word 0
Bit 01: Dependent on the setting in code C4000/000		
Bit 02: Dependent on the setting in code C4000/000		
Bit 03: Dependent on the setting in code C4000/000		
Bit 04: Dependent on the setting in code C4000/000		
Bit 05: Dependent on the setting in code C4000/000		
Bit 06: Dependent on the setting in code C4000/000		
Bit 07: Dependent on the setting in code C4000/000		
Bit 08: Dependent on the setting in code C4000/000		
Bit 09: Dependent on the setting in code C4000/000		
Bit 10: Dependent on the setting in code C4000/000		
Bit 11: Dependent on the setting in code C4000/000		
Bit 12: Dependent on the setting in code C4000/000		
Bit 13: Dependent on the setting in code C4000/000		
Bit 14: Dependent on the setting in code C4000/000		
Bit 15: Dependent on the setting in code C4000/000		
Error number (<i>g_wErrorNumber</i>)	Data word 1	Word 1
Bit 00: System error (<i>g_bGlobalError</i>)	Application status word	Word 2
Bit 01: Following error warning (<i>g_bFollowErrorWarn</i>)		
Bit 02: Following error (<i>g_bFollowErrorQSP</i>)		
Bit 03: Positioning in progress (<i>g_bToolPositioningBusy</i>)		
Bit 04: Target position reached (<i>g_bToolPositioningDone</i>)		
Bit 05: System locking to motion profile (<i>g_bLockToCamBusy</i>)		
Bit 06: System locked to motion profile (<i>g_bLockToCamDone</i>)		
Bit 07: Mark correction active (<i>g_bXTpSyncBusy</i>)		
Bit 08: Calculation sequence for new cross-cutter motion profile for a new cutting length in progress (<i>g_bCrossCutterCalcBusyCuttingProfile</i>)		
Bit 09: Calculation sequence for new cross-cutter motion profile for a new reject length in progress (<i>g_bCrossCutterCalcBusyRejectProfile</i>)		
Bit 10: Calculation sequence for new cross-cutter motion profile for a new test length in progress (<i>g_bCrossCutterCalcBusyTestProfile</i>)		
Bit 11: Calculation sequence for new cross-cutter motion profile for a new start/stop length in progress (<i>g_bCrossCutterCalcBusyStartStopProfile</i>)		
Bit 12: Calculation sequence for new cross-cutter motion profile for a new cutting length completed successfully. Starting a new calculation sequence will reset this status output to FALSE (<i>g_bCrossCutterCalcDoneCuttingProfile</i>).		
Bit 13: Calculation sequence for new cross-cutter motion profile for a new reject length completed successfully. Starting a new calculation sequence will reset this status output to FALSE (<i>g_bCrossCutterCalcDoneRejectProfile</i>).		
Bit 14: Calculation sequence for new cross-cutter motion profile for a new test length completed successfully. Starting a new calculation sequence will reset this status output to FALSE (<i>g_bCrossCutterCalcDoneTestProfile</i>).		
Bit 15: Calculation sequence for new cross-cutter motion profile for a new start/stop length completed successfully. Starting a new calculation sequence will reset this status output to FALSE (<i>g_bCrossCutterCalcDoneStartStopProfile</i>).		
Bit 16: Homing activated (<i>g_bHomingBusy</i>)		Word 3
Bit 17: Home position detected (<i>g_bHomingDone</i>)		
Bit 18: "Homing error" status signal (<i>g_bHomingError</i>)		
Bit 19: (not assigned)		
Bit 20: The cross cutter is operating as an asynchronous cross cutter (<i>g_bUseAsynchronousCrossCutter</i>).		
Bit 21: The cross cutter is operating as a synchronous cross cutter (<i>g_bUseSynchronousCrossCutter</i>).		
Bit 22: The cross cutter is operating in "Continuous cutting" mode (<i>g_bUseCuttingProfile</i>).		
Bit 23: The cross cutter is operating in "Reject operation" mode (<i>g_bUseRejectProfile</i>).		
Bit 24: The cross cutter is operating in "Test operation" mode (<i>g_bUseTestProfile</i>).		
Bit 25: Opening reject gate (<i>g_bOpenRejectGate</i>)		
Bit 26: The maximum number of cycles without a mark signal appearing in the mark window has been exceeded (<i>g_bMaxNumberOfTPOutOfWindowIncreased</i>).		
Bit 27: System ready for print-mark control (<i>g_bSystemReadyToUseTP</i>)		
Bit 28: Using print-mark register (<i>g_bUseShiftedTP</i>)		
Bit 29: Knife in material (NOT <i>g_bKnifeNotInMaterial</i>)		
Bit 30: (not assigned)		
Bit 31: All calculation sequences for new cross-cutter motion profiles have been completed successfully. Starting a new calculation sequence will reset this status output to FALSE (<i>g_bAllCalcDone</i>).		

Cross Cutter

Installing and starting up the "CrossCutter" prepared solution

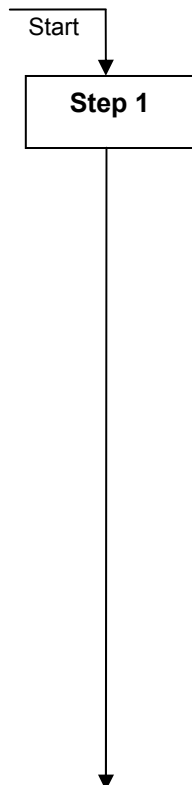
The individual status words are output to system variables (designated *Word 0*, *Word 1*, *Word 2* and *Word 4*) dependent on the operating mode in C4000/000:

Signal source	Word 0	Word 1	Word 2	Word 3
C4000/000 = 0	C0135/000	C4152/000	C4150/000	
C4000/000 = 1	CAN1_wDctrlStat	CAN1_nOutW1_a	CAN1_nOutW2_a	CAN1_nOutW3_a
C4000/000 = 2	CAN2_nOutW1_a	CAN2_nOutW2_a	CAN2_nOutW3_a	CAN2_nOutW4_a
C4000/000 = 3	CAN3_nOutW1_a	CAN3_nOutW2_a	CAN3_nOutW3_a	CAN3_nOutW4_a
C4000/000 = 4	AIF1_wDctrlStat	AIF1_nOutW1_a	AIF1_nOutW2_a	AIF1_nOutW3_a
C4000/000 = 5	AIF2_nOutW1_a	AIF2_nOutW2_a	AIF2_nOutW3_a	AIF2_nOutW4_a
C4000/000 = 6	AIF3_nOutW1_a	AIF3_nOutW2_a	AIF3_nOutW3_a	AIF3_nOutW4_a

The illustration on the previous page shows the factory-set default configuration of the 32-bit application status word. Users can also adapt the configuration of the application status word in accordance with their individual needs by setting the values of codes C4012/001 to C4012/032 accordingly.

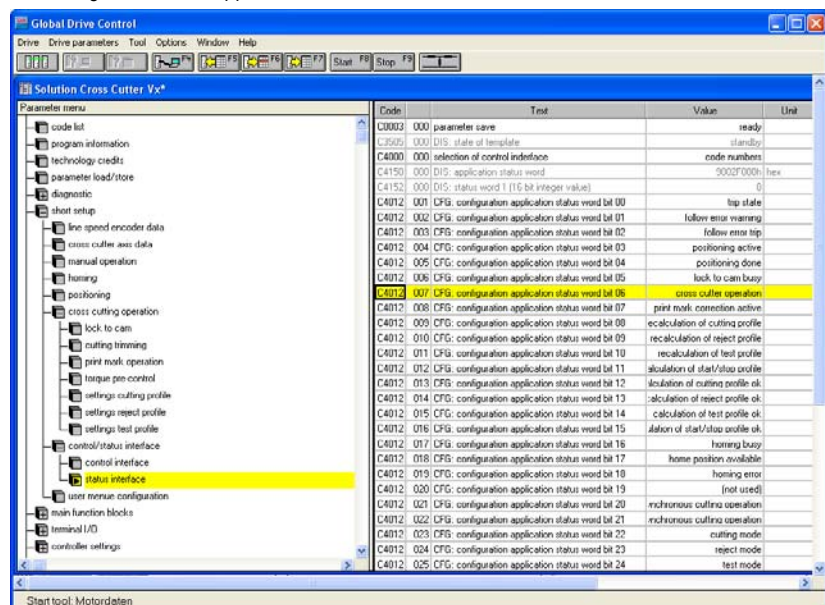
An example of how the **GDC** parameterisation software can be used to reconfigure the application status word bit by bit appears below:

You will find the code in the complete code list or in the menu Short Setup → Control/Status Interface → Status Interface:



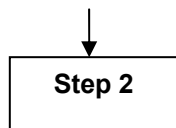
Select the application status word bit you wish to reconfigure.

Example: To assign bit 6 of the application status word, select code C4012/007:



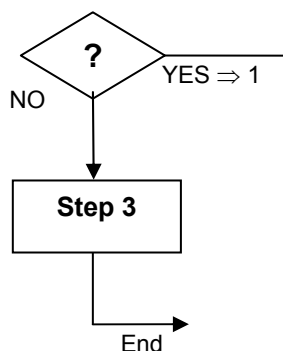
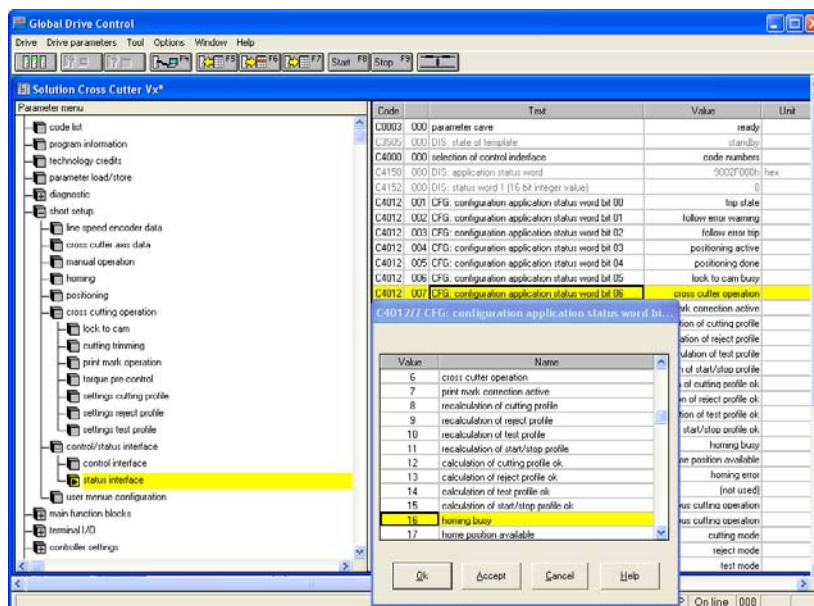
Cross Cutter

Installing and starting up the "CrossCutter" prepared solution



Click on the code and select the status information you wish to assign to this status bit for display.

Example: If you wish to see the "Homing in progress" signal, select C4012/007 = 16 (Homing in progress) and confirm with Ok.



To reconfigure other status bits, proceed as described in Step 1.

Save the modified settings to non-volatile memory with C0003/000 = 1.



TIP!

Regardless of the operating mode you select via C4000/000, the application status word will be displayed via code C4150/000.

Cross Cutter

Installing and starting up the "CrossCutter" prepared solution

3.4.2 Preparing for installation and start-up: Generating profile data

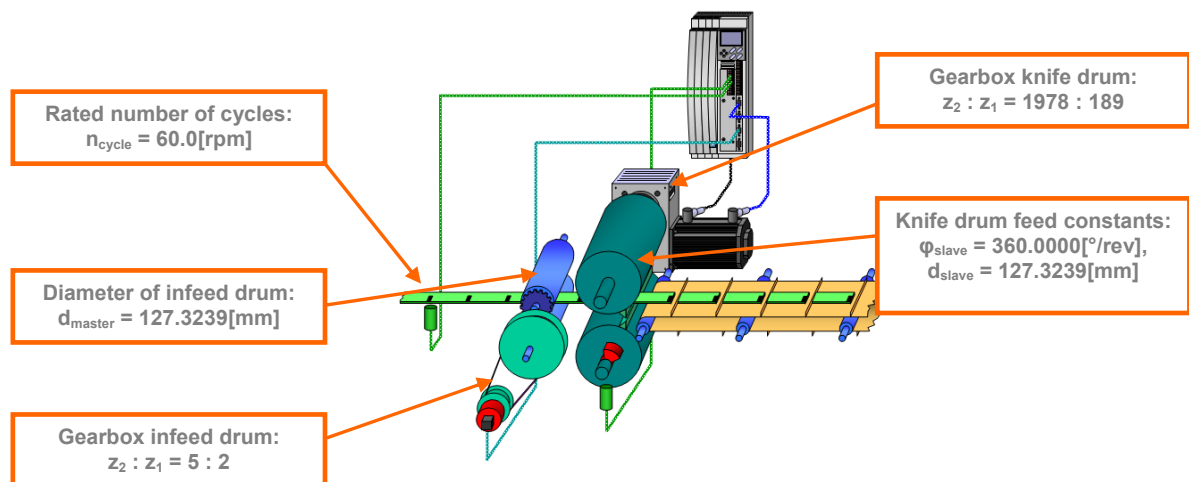
The installation and start-up sequence for the prepared solution described below relates to the parameterisable variant.

The Cross Cutter prepared solution is based on the "Electronic cam" technology function which requires additional data (tables of values for cams). This data is project-specific and cannot be provided by Lenze/can only be provided in restricted format by Lenze (in the form of a ready-to-use file). The following section describes how to generate this additional data.

If you are using the "Electronic cam" technology function on the ServoPLC or ECS servo system you will need an additional file (LC9 file) which is generated using Lenze's **CamDesigner** software tool. **CamDesigner** writes all cam-relevant data to this file. Such data includes:

- Axis data for the master and cam axes (in the file header)
- Number of motion profiles and type of data model (in the file header)
- All motion profiles $y = f(x)$ (motion profiles not defined by the user are predefined automatically with a pause)
- Number of interpolation points for each motion profile (= number of interpolation points for the torque precontrol profile)
- Information about decimal positions for master and cam cycle length (ensures that the drive system will not drift)
- Torque precontrol profiles (profiles not defined by the user contain a zero signal)
- Rated number of cycles in order for the scaled torque precontrol profile to be scaled correctly
- Cam switching points (not relevant for this application)

A basic cam file has to be generated because although the *CrossCutterMotionProfileCalculation* function block is able to modify motion profiles, it cannot generate profile data in the first instance. A step-by-step description of how to generate a basic LC9 file using an example appears below³:



³ You will need to install **CamDesigner Basic** to use the example.

Cross Cutter

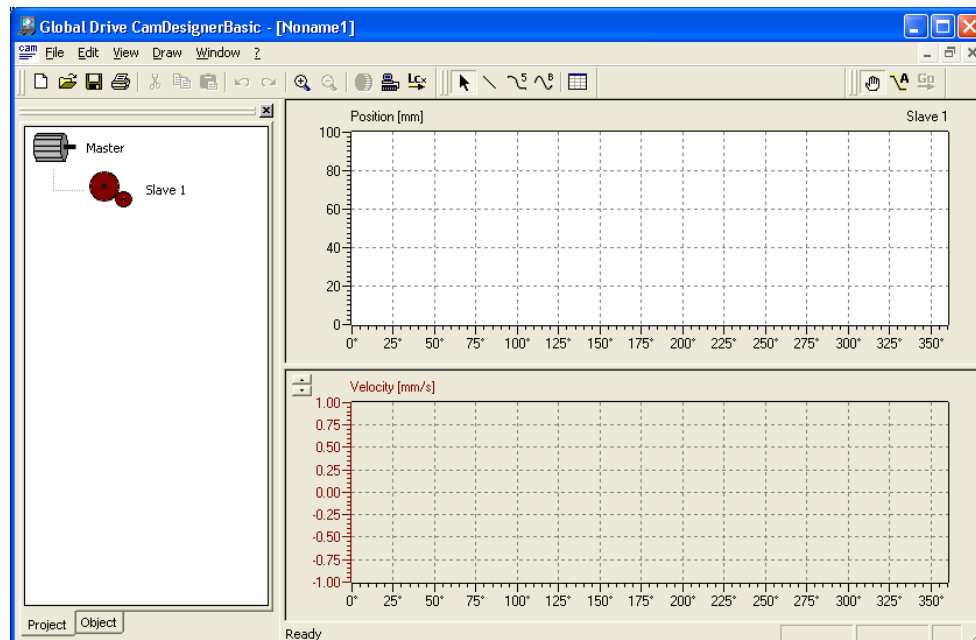
Installing and starting up the "CrossCutter" prepared solution

Start

Step 1

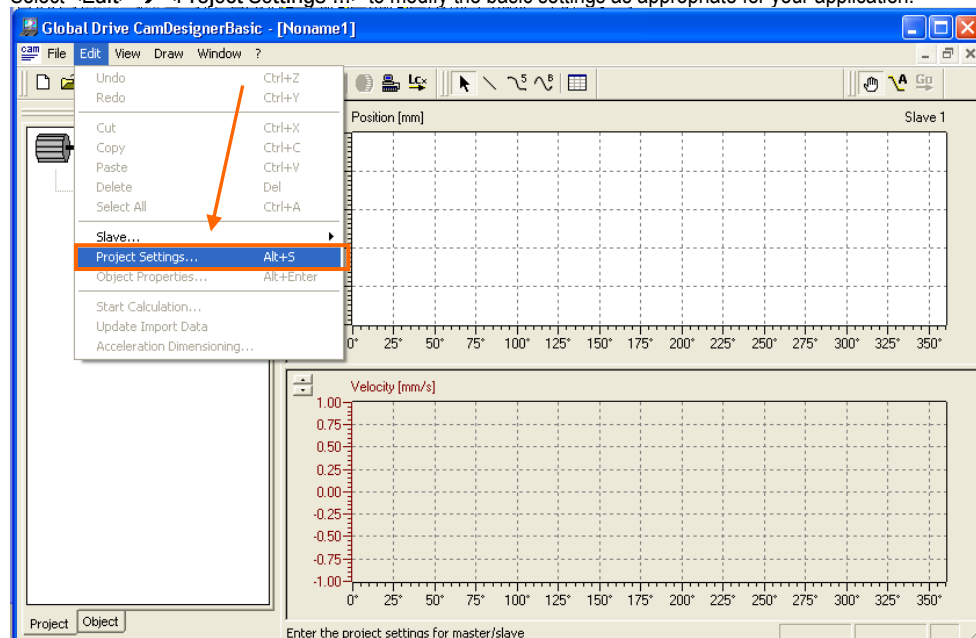
Start Lenze's CamDesigner Basic software tool

(<Start> → <Programs> → <Lenze> → <Global Drive Cam Designer Basic> → <Cam Designer>):



Step 2

Select <Edit> → <Project Settings ...> to modify the basic settings as appropriate for your application:



Installing and starting up the "CrossCutter" prepared solution

Step 3

The Project Settings dialog box will appear. Click the Master tab (master value source parameters):

The screenshot shows the 'Settings' dialog box with the 'Master' tab selected. The 'Name' field is 'Line Speed Encoder'. The 'Scale' section has 'Minimum' set to 0.0000, 'Maximum' set to 400.0000, and 'Unit' set to mm. The 'Cycles' section has '60.00' for 'Cycles/min' and '60.00' for 'Rated cycles/min'. The 'Machine' section has 'Gear factors' with 'z2 = 5' and 'z1 = 2', 'Feed constant' set to 360.0000, and 'Increment/unit' set to 455. An orange arrow points to the 'Master' tab.

- Name: Name of the master value axis (in the example: "Line Speed Encoder")
- Minimum: Must always be set to 0.0000[mm]
- Maximum: Entry indicating the cutting circle circumference of the knife drum on the rotary cross cutter (in the example: 400.0000[mm])
- Unit: Unit in which the cutting length is to be scaled (in the example: 1[m_unit] = 1[mm])
- Gearbox factors: Entry indicating the gearbox factors of the master signal encoder (in the example: $z_2 : z_1 = 5 : 2$)
- Feed constant: Entry indicating the feed constant of the master signal encoder = path feed in the master signal with revolution of the master value at the gearbox output end (in the example: $\pi \cdot 127.3239[\text{mm}] = 400.0000[\text{mm}]$)
- Rated quantity: In this case, entry indicating the maximum number of cycles in [rpm] at which the rotary cross cutter is to run. Initially, this entry refers solely to the synchronous cutting length.

Installing and starting up the "CrossCutter" prepared solution

Step 4

Click the Slave 1 tab to make the settings for the drive axis on the rotary cross cutter:

<u>Name:</u>	Name of the cam axis (in the example: "CrossCutter")
<u>Minimum:</u>	Minimum position in the cross-cutter cycle, must always be set to 0.0000[°]
<u>Maximum:</u>	Maximum position y_{max} in the cross-cutter cycle: This value is dependent on the mechanical engineering of the knife drum and may exhibit the following values: $y_{max} = 360.0000[°]$: 1 knife per knife drum $y_{max} = 180.0000[°]$: 2 knives per knife drum $y_{max} = 120.0000[°]$: 3 knives per knife drum etc. $y_{max} = 60.0000[°]$: 6 knives per knife drum
<u>Unit:</u>	Unit in which the feed motion of the cross cutter is scaled (in the example: 1[s_unit] = 1[°])
<u>Gearbox factors:</u>	Entry indicating the gearbox factors of the cross-cutter drive (with numerator and denominator indicated separately) (in the example: $z_2 : z_1 = 1978 : 189$)
<u>Feed constant:</u>	Entry indicating the feed constant of the cross cutter, must always be set to 360[°]
<u>Data model:</u>	In order to be able to modify interpolation points in motion profiles subsequently, you must select the absolute data model.
<u>Type:</u>	Select the "ServoPLC/ECS servo system" as your target system.
<u>Profiles:</u>	Selection of the number of motion profile (in the example: 10 motion profiles selected)



Caution!

The prepared solution needs 10 motion profiles!

Cross Cutter

Installing and starting up the "CrossCutter" prepared solution

Step 5

Points:

Entry indicating the number of interpolation points for the motion and torque precontrol profiles. The number of interpolation points should be selected so that the x position of the setpoint-generating task moves forward one point per task cycle. An approximate calculation of the number of interpolation points can be arrived at as follows:

$$n_{points} = \frac{6000[points]}{[min]} \cdot \frac{1}{n_{cycle}} \cdot \frac{1}{t_{MC_task}}$$

where

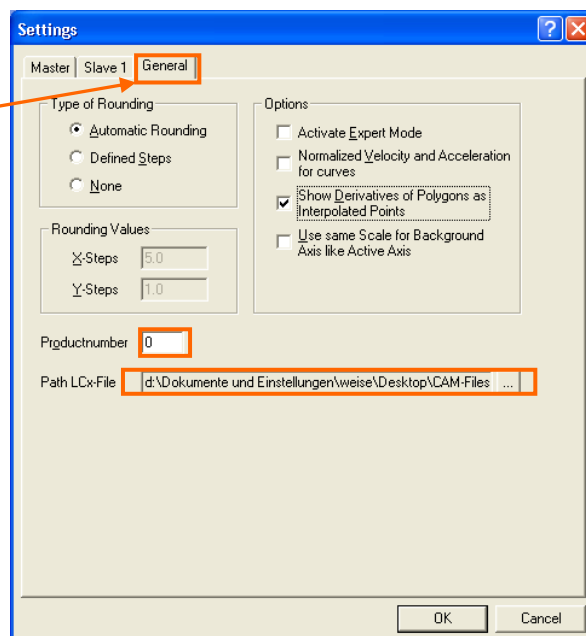
n_{points} : Number of interpolation points to be selected per motion profile
 n_{cycle} : Maximum number of cycles per minute
 t_{MC_task} : Interval time of setpoint-generating task

Example: The setpoint-generating task is processed in the 2 ms cycle. At a maximum clock-pulse rate of 60 cuts/minute a sensible value for the number of interpolation points would be

$$n_{points} = \frac{6000[points]}{[min]} \cdot \frac{1}{60[cuts/min]} \cdot \frac{1}{2[ms]} = 500[points]$$

Tip: Practical Tests have shown, that a number of points from 50 up to 100 are normally enough in most applications and the calculation time will be smaller.

Click the General tab to specify the profile number to be processed for the motion profile of the cross cutter and define a directory path for the LC9 file:



Product number:

You will need to enter a profile number (product number) here in order to enable the motion profile to be addressed subsequently on the ServoPLC/in the ECS servo system. Example: The default setting for the first motion profile for the cross- cutter application is profile number 0.

LCx file path:

Entry indicating the directory path (memory location) for the LC9 file to be generated
 Click Ok to go back to the main CamDesigner window.

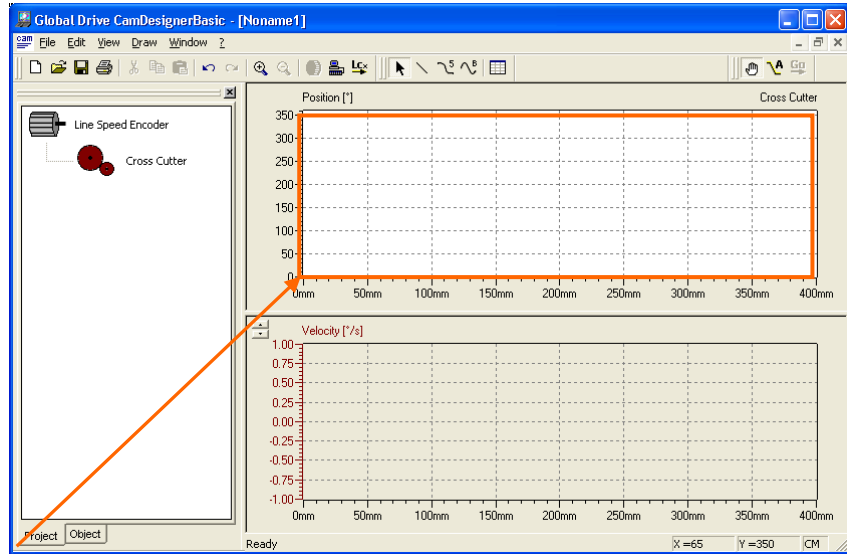
Cross Cutter
Installing and starting up the "CrossCutter" prepared solution

Cross Cutter

Installing and starting up the "CrossCutter" prepared solution

Step 6

The main CamDesigner window will now display the master axis (*Line Speed Encoder*) and the profile axis (cross-cutter axis = *CrossCutter*):



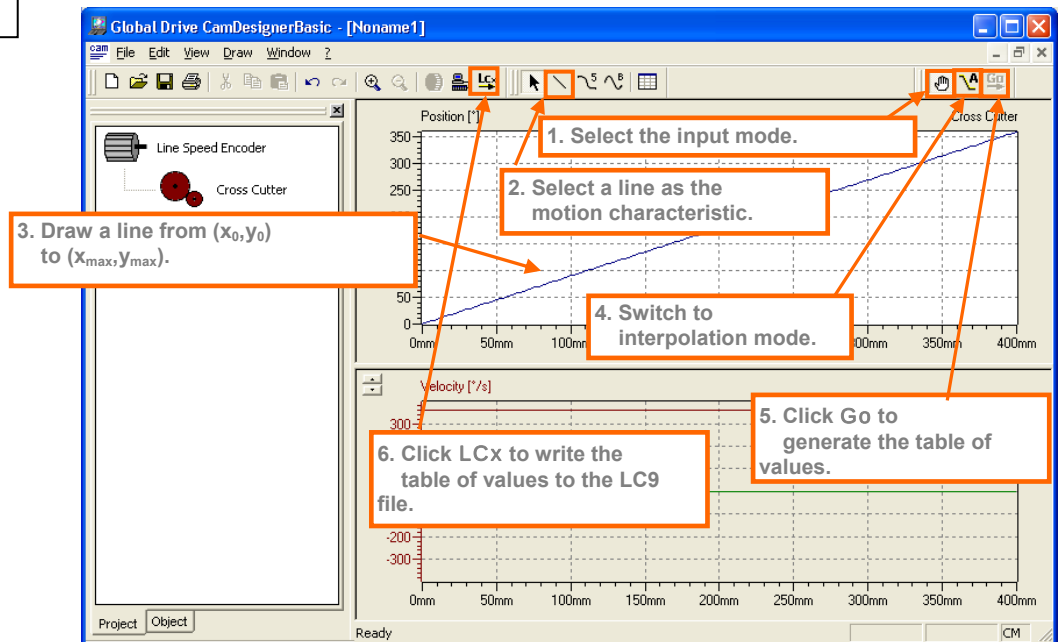
The motion profile characteristic is defined and converted into an interpolation point table in the input window. However, if you are using the *CrossCutterMotionProfileCalculation* function block, the definitive format of the motion profile will not be defined until runtime of the PLC program, although CamDesigner is required at this point to define the number of interpolation points and decimal positions for the Y-cycle (see next step).

Cross Cutter

Installing and starting up the "CrossCutter" prepared solution

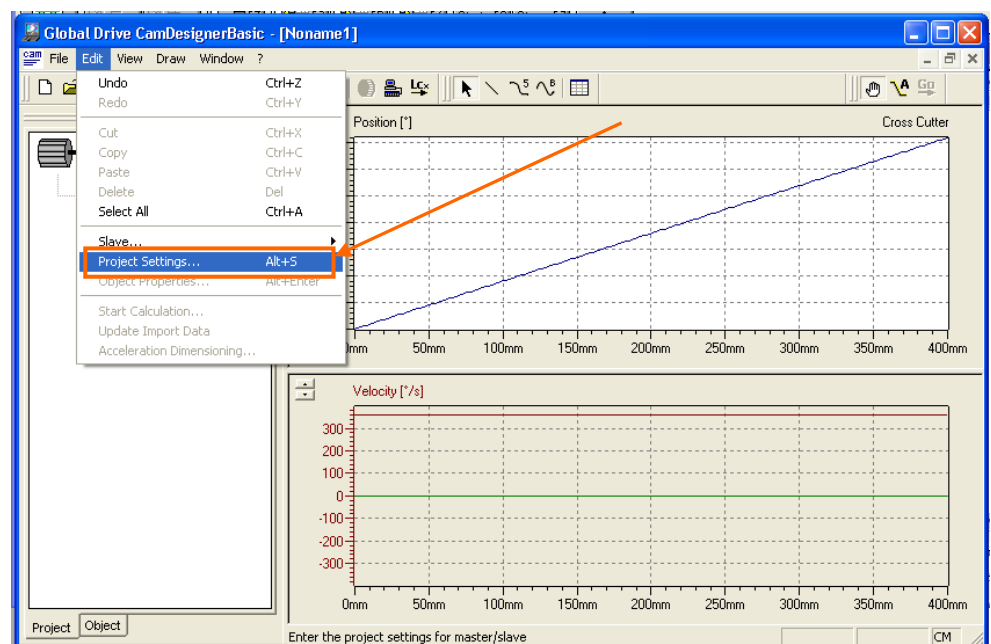
Step 7

Proceed as follows to generate a basic motion profile:



Step 8

Click <Edit> → <Project Settings ...> to modify the project settings for the next motion profile:

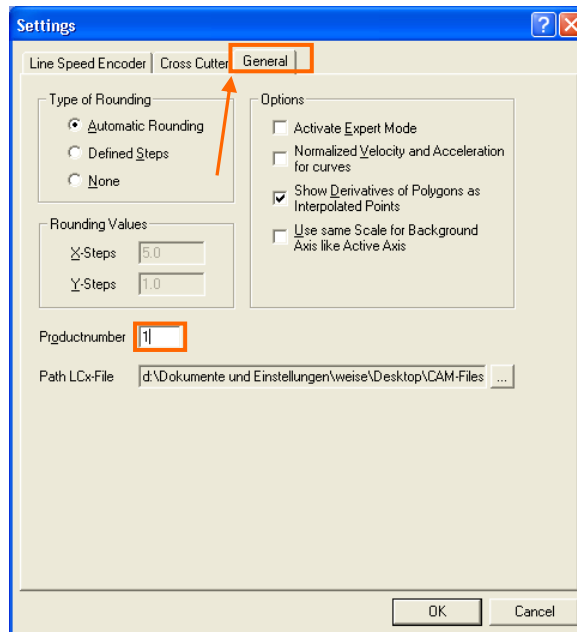


Cross Cutter

Installing and starting up the "CrossCutter" prepared solution

Step 9

Click the General tab to set the number of the next motion profile:



Product number:

Change the product number to the next motion profile (in the example 1).

Click the Ok button to close the dialog window.

Cross Cutter

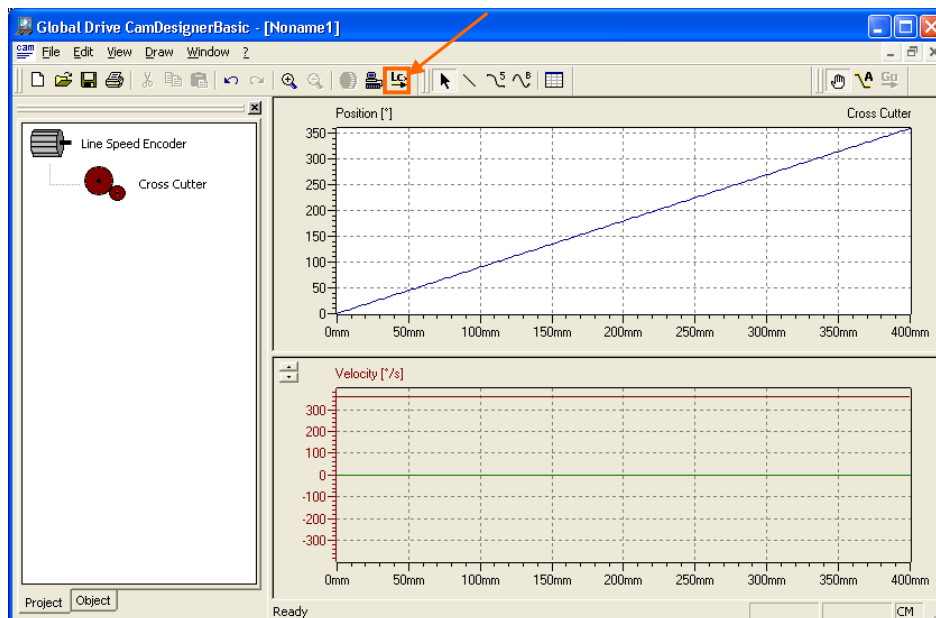
Installing and starting up the "CrossCutter" prepared solution

↓

Step 10

↓

Click LCx to write the table of values to the LC9 file for the next motion profile also:



↓

Step 11

↓

Repeat Steps 9 to 11 until you have created all 10 motion profiles.

End

Exit CamDesigner.

The 10 motion profiles are used as follows in the prepared solution:

- Asynchronous cross-cutter mode uses motion profile 0.
- Motion profiles 1 and 2 are used for Continuous cutting mode.
- Motion profiles 3 and 4 are used for Reject operation mode.
- Motion profiles 5 and 6 are used for Test operation mode.
- Motion profiles 7 and 8 are used to start and stop the cross cutter.
- Motion profile 9 is reserved for future expansions.

Cross Cutter

Installing and starting up the "CrossCutter" prepared solution

3.4.3 Downloading the CrossCutter_SPLC_Vxxxx.bin program and CrossCutter_ECS_Vxxxx.bin

Start

Step 1

Step 2

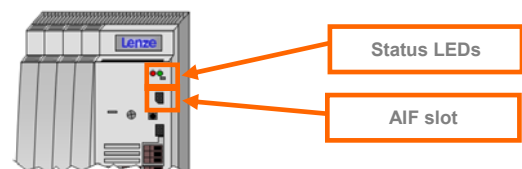
Requirements:

- The servo controller has been set up and wired in accordance with the wiring plan in Chapter 3.3.1 and as per the product documentation.
- The axis has been inhibited via terminal 28 if you are using 9300 Servo PLC or via terminal SI1 if you are using the ECS.
- Control voltage is being supplied to the axis via terminals 59 (+24 V DC) and 39 (GND).
- Use of keypad XT EMZ9371BC to make and modify code parameter settings.
- The prepared solution cross cutter and the required Lenze software tools have been installed on your PC/laptop.
- The EMF2177IB/EMF2173IB system bus adapter has been successfully installed.

Use the keypad XT EMZ9371IB to set the following codes on the cross cutter's servo controller in the sequence indicated:

C2108/000	2 = Stop active PLC program
C0002/000	0 = Apply default setting
C0350/000	Node address of servo axis (each node address can only be used once on the system bus)
C0352/000	1 = Cross-cutter servo controller is bus master on system bus
C0003/000	1 = Saving of system bus settings to non-volatile memory

Disconnect the power supply entirely from the cross-cutter servo controller and wait for the red and green status LEDs on the servo controller to go out, or for at least 20 s.

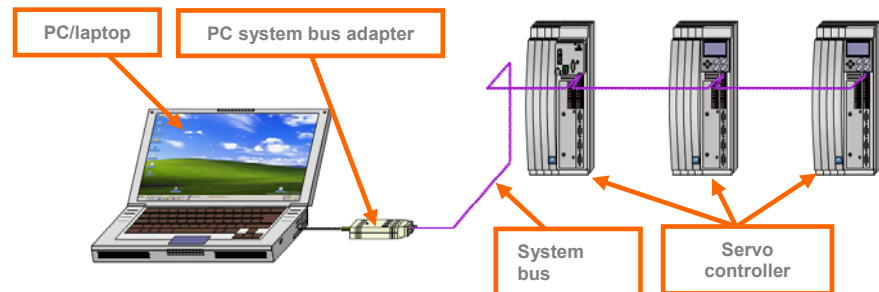


Cross Cutter

Installing and starting up the "CrossCutter" prepared solution

Step 3

Connect your PC/laptop to the servo axis system bus via the EMF2177IB/EMF2173IB PC system bus adapter (you may be using more than one axis in the drive system):



TIP!

To facilitate connection, the EMF2177IB/EMF2173IB PC system bus adapter is supplied with a three-pin system bus connector supporting easy connection of the PC system bus adapter to the system bus.

Step 4

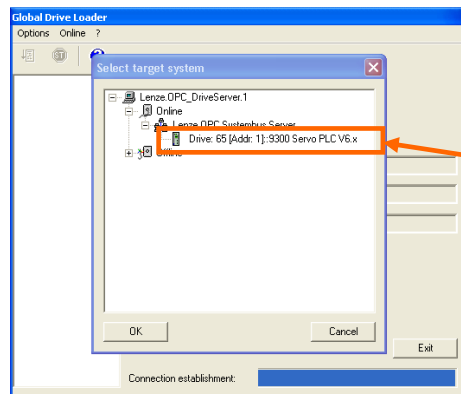
Reconnect the servo controller to the control voltage.

Cross Cutter

Installing and starting up the "CrossCutter" prepared solution

Step 5

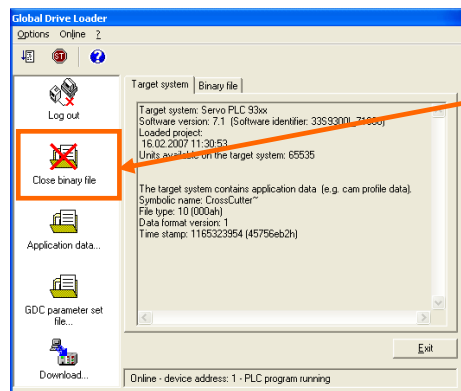
Start GDLoader and run a search for all drives connected to the system bus. GDLoader will scan the system bus and find the servo drive:



Click the required servo controller (in the example, this is the controller with system bus node address 1) and confirm with Ok.

Step 6

Next, assign the files to the required servo axis by clicking the corresponding icons in the left-hand column:



Select the following file as the DDS binary file:

Project:

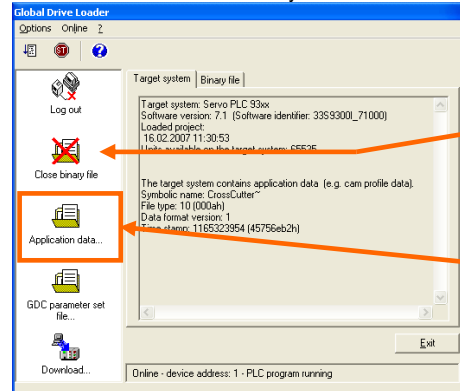
For Servo PLC: *CrossCutter_SPLC_Vxxxxxx.bin*

For ECS: *CrossCutter_ECS_Vxxxxxx.bin*

Installing and starting up the "CrossCutter" prepared solution

Step 7

Assign the prepared cam file (LC9 file) by clicking the Application data... icon in the left-hand column: This step has been included for the operation of the GDLoader. The scope of supply of the prepared solution does not include a LC9 file. They have to be created by the user (see chapter 3.4.2).

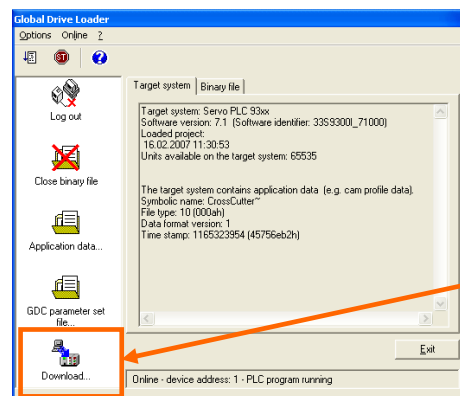


File types you have already assigned will appear crossed through in red. You can undo an assignment by clicking again on the icon crossed through in red.

Assign the prepared cam file (LC9 files).

Step 8

Click the Download icon to start the download to the servo axis:

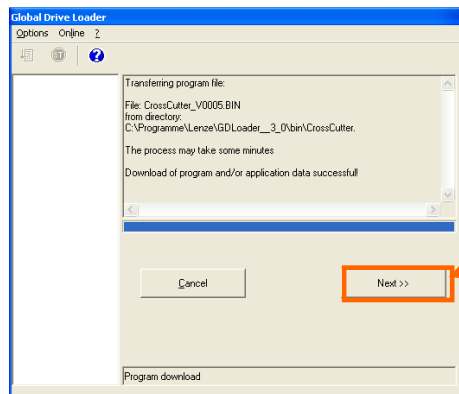


Cross Cutter

Installing and starting up the "CrossCutter" prepared solution

Step 9

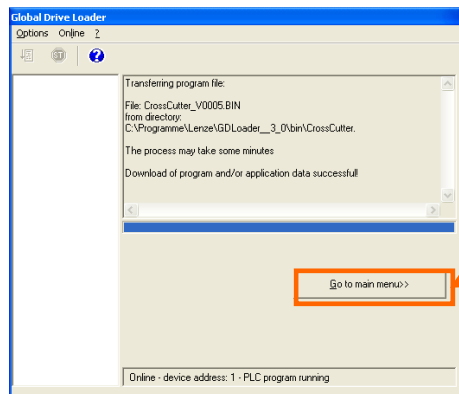
Wait for the download to complete.



Once the download is complete, click Next.

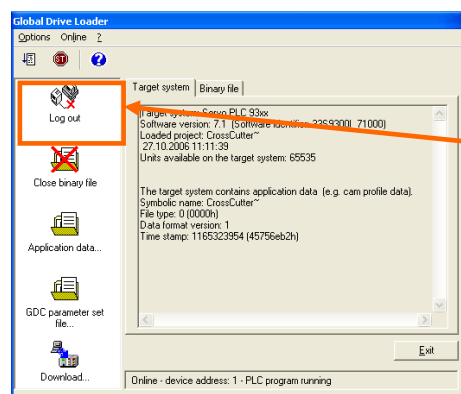
Step 10

Click Go to Main Menu >> to go to the main GDLoader window:



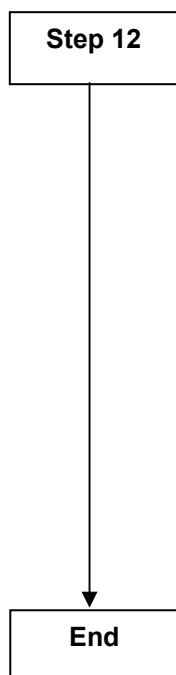
Step 11

Click the Log Out button to disconnect from the servo controller:

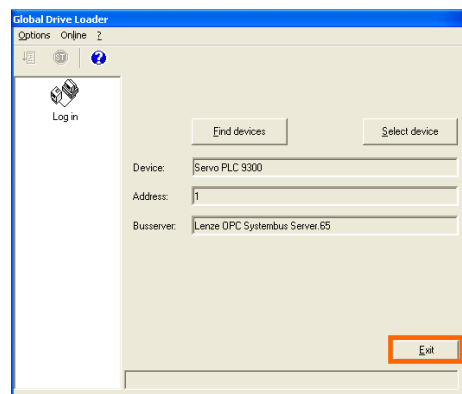


Cross Cutter

Installing and starting up the "CrossCutter" prepared solution



Exit GDLoader:



Program download to the controller is now complete.

Cross Cutter

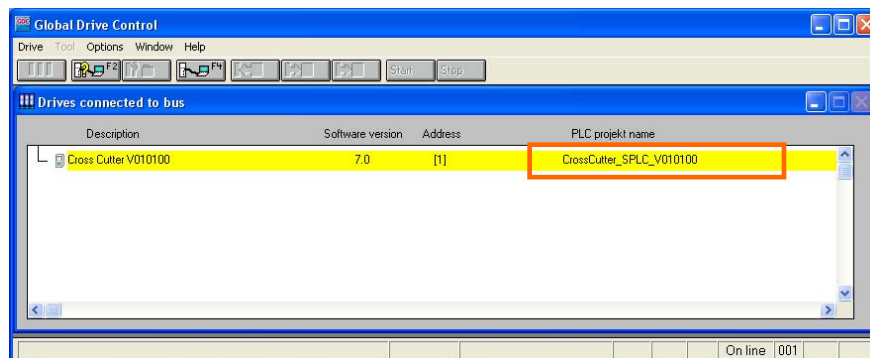
Installing and starting up the "CrossCutter" prepared solution

3.4.4 Online start-up with GDC

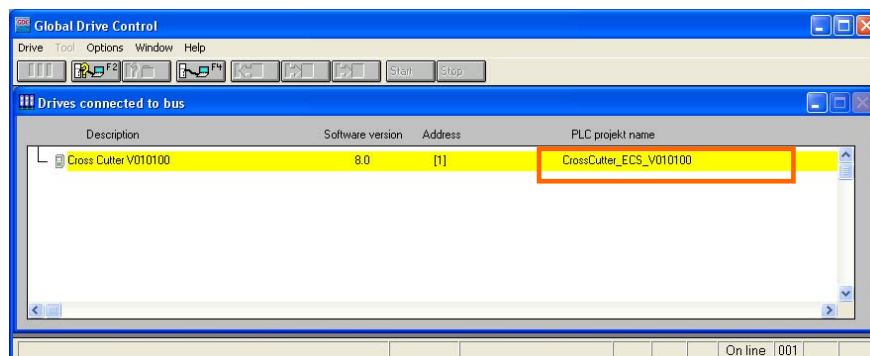
Start

Step 1

Start GDC and run a search for the controllers connected to the system bus. GDC will find the connected controllers. These will include the controller loaded with the *CrossCutter_SPLC_Vxxxxxx* for 9300 Servo PLC and *CrossCutter_ECS_Vxxxxxx* for ECS.



Prepared Solution for 9300 Servo PLC



Prepared Solution for ECS

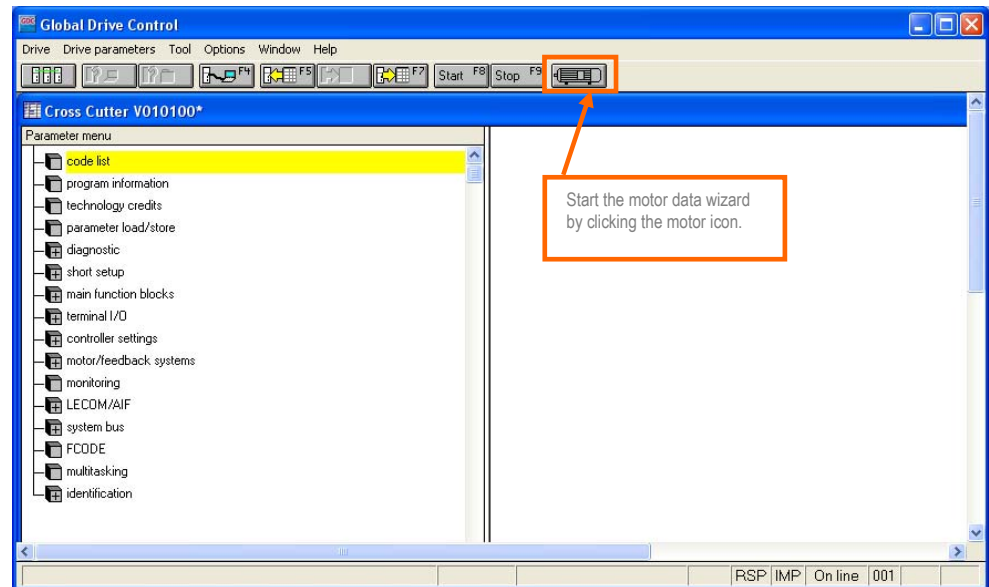
Double-click the cross-cutter axis drive and read back the parameters from the drive.

Cross Cutter

Installing and starting up the "CrossCutter" prepared solution

Step 2

Start the motor data wizard:



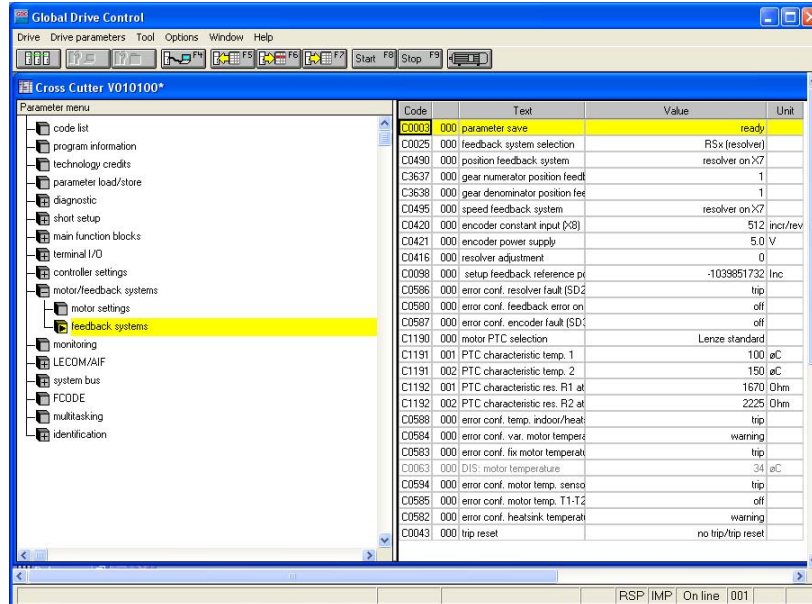
Follow the Instructions issued by the motor data wizard to configure the connected motor correctly.
Exit the motor data wizard once you have completed the motor parameter settings.

Cross Cutter

Installing and starting up the "CrossCutter" prepared solution

Step 3

Call up the parameters for configuring the speed and position feedback system (Motor/Feedback Systems → Feedback Systems):



Make the following code settings:

- C0490/000 Setting feedback system position control
 - C0495/000 Setting feedback system speed control
 - C0420/000 Number of encoder increments when using TTL incremental encoders at X8 or number of periods on sincos encoders at X8
 - C0421/000 Setting of supply voltage of feedback system at X8
- In addition, in the case of separate position and speed feedback:
- C3637/000 Gearbox (numerator) position feedback (see Chapter 2.5.16)
 - C3638/000 Gearbox (denominator) position feedback (see Chapter 2.5.16)



Caution!

For applications with position feedback on the load side, please note that the position encoder must rotate in the same direction as the speed sensor!



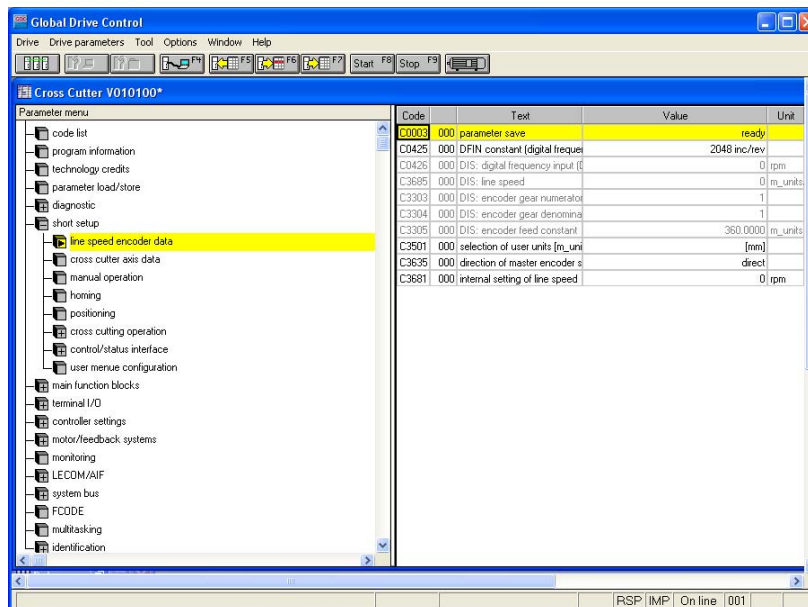
Caution!

If you use the ECS Sytem: With the Parameterisable variant it is not possible to use an absolute encoder because this encoder type uses the same input X8 as the frequency coupling via DFIN. If an absolute encoder is necessary in the application the user has to use the programmable version of the prepared solution Cross Cutter and have to handle the absolute encoder on its own.

Installing and starting up the "CrossCutter" prepared solution

Step 4

Next, set the parameters for the digital frequency input
(Short Setup → Line Speed Encoder Data):



Make the following code settings for 9300 Servo PLC:

- C0425/000 DFIN constant for digital frequency input (X9)
- C3635/000 Direction of rotation master value encoder (X9)

Make the following code settings for ECS:

:

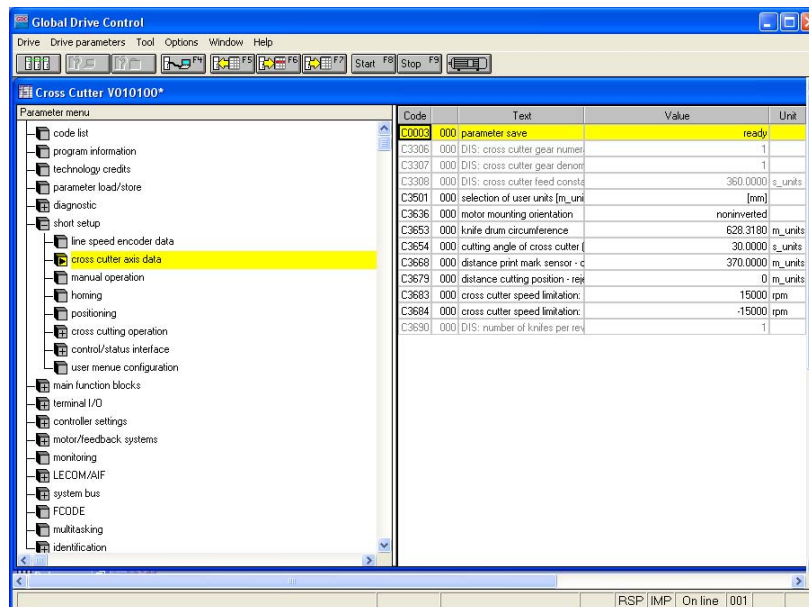
- C0420/000 Encoder number of increments (X8)
- C0421/000 Encoder power supply(X8)
- C3635/000 Direction of rotation master value encoder (X8)

Cross Cutter

Installing and starting up the "CrossCutter" prepared solution

Step 5

Call up the parameters for configuring the cross-cutter axis
(Short Setup → Cross-Cutter Axis Data):



Make the following code settings:

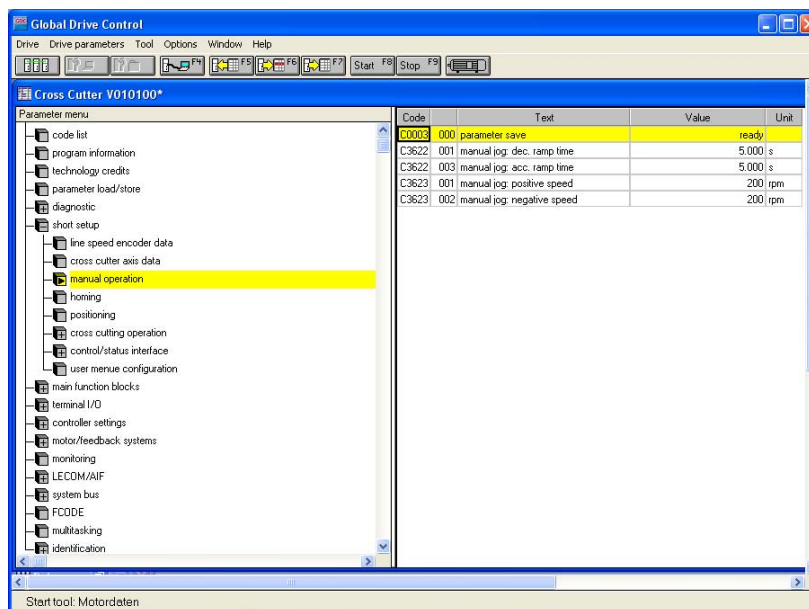
- C3501/000 Selection of application units [m_units]
- C3636/000 Motor mounting direction
- C3653/000 Cutting circle circumference
- C3654/000 Cutting angle (synchronous angle)
- C3668/000 Distance mark sensor - cutting position
- C3679/000 Distance cutting position - reject gate
- C3683/000 Speed limiter: Upper speed limit
- C3684/000 Speed limiter: Lower speed limit

Cross Cutter

Installing and starting up the "CrossCutter" prepared solution

Step 6

Make the parameter settings for manual jog (Short Setup → Manual Operation):



TIP!

See Chapter 2.5.4 for more information about manual jog.

Make the following code settings:

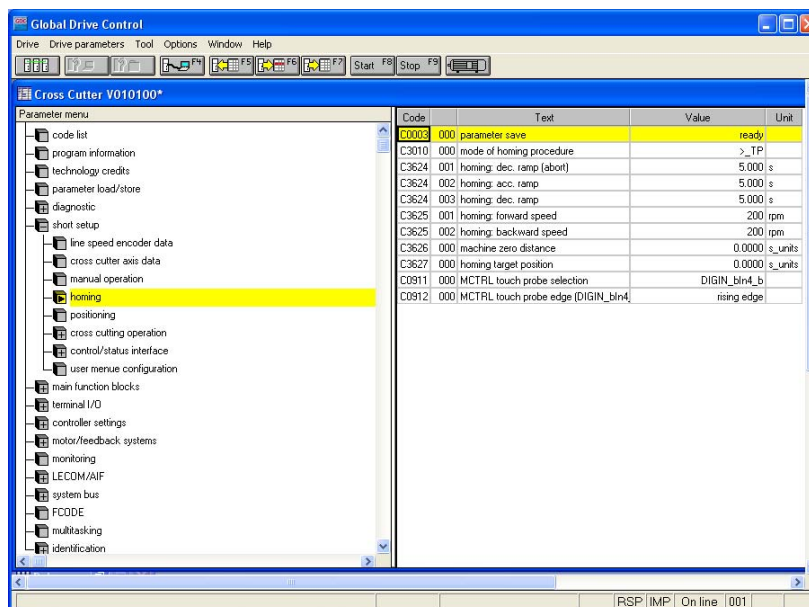
- C3622/001 Inching mode: Deceleration ramp
- C3622/003 Inching mode: Acceleration ramp
- C3623/001 Inching mode: Positive speed
- C3623/002 Inching mode: Negative speed

Cross Cutter

Installing and starting up the "CrossCutter" prepared solution

Step 7

You now need to make the parameter settings for homing (Short Setup → Homing):



TIP!

See Chapter 2.5.6 for more information about homing.

Make the following code settings:

- C3010/000 Homing mode
- C3624/001 Homing: Deceleration time (abort)
- C3624/002 Homing: Acceleration time
- C3624/003 Homing: Deceleration time
- C3625/001 Homing: Positive traversing speed
- C3625/002 Homing: Negative traversing speed
- C3626/000 Position of homing switch
- C3627/000 Target position of axis following homing

If you are using a homing mode which requires a TP sensor signal you will also have to set the TP source:

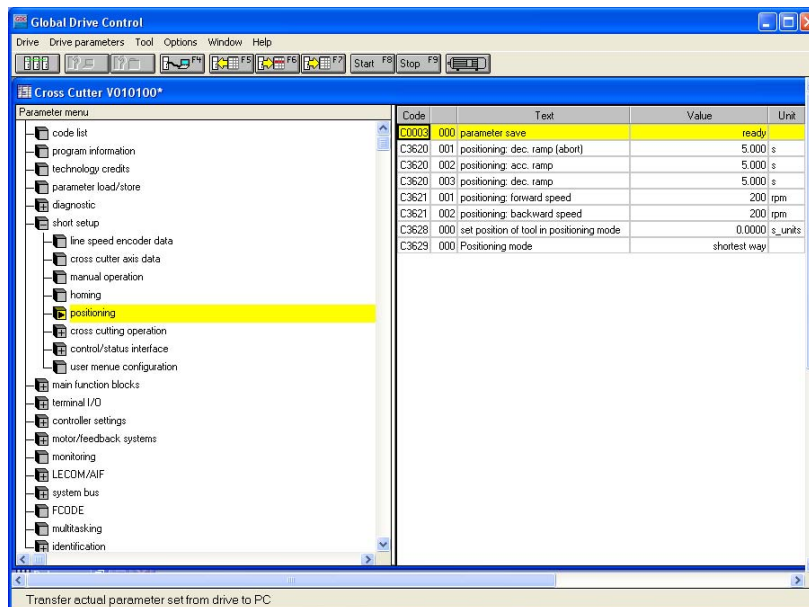
- C0911/000 MCTRL Touch-probe source
- C0912/000 MCTRL Touch-probe edge

Cross Cutter

Installing and starting up the "CrossCutter" prepared solution

Step 8

You now need to make the settings for the positioning parameters (Short Setup → Positioning):



TIP!

See Chapter 2.5.7 for more information about positioning.

Make the following code settings:

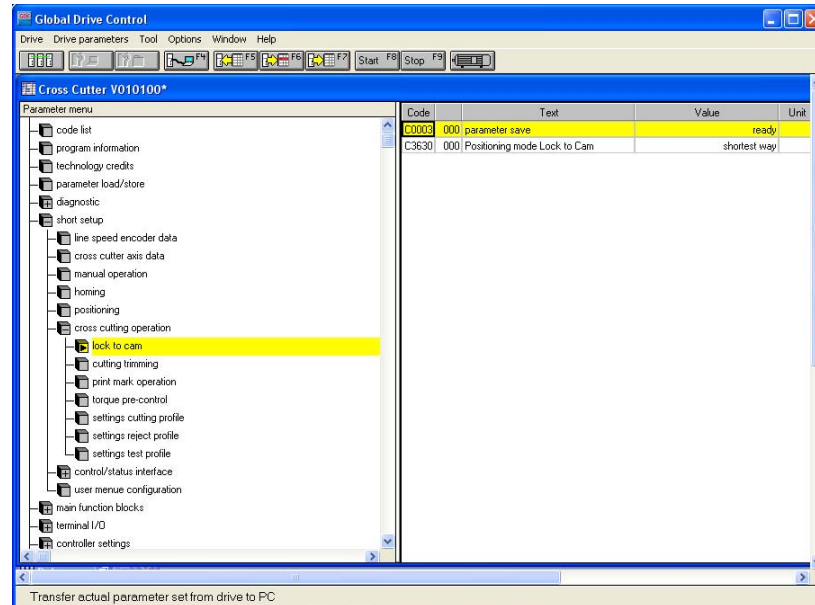
C3620/001	Positioning: Deceleration time (abort)
C3620/002	Positioning: Acceleration time
C3620/003	Positioning: Deceleration time
C3621/001	Positioning: Positive traversing speed
C3621/002	Positioning: Negative traversing speed
C3628/000	Target position (positioning mode)
C3629/000	Positioning mode

Cross Cutter

Installing and starting up the "CrossCutter" prepared solution

Step 9

You now need to make the settings for the Lock To Cam parameters
Short Setup → Cross Cutting Operation → Lock To Cam):



TIPPI!

See Chapter 2.5.1 for more information about Lock To Cam.

Make the following code settings:

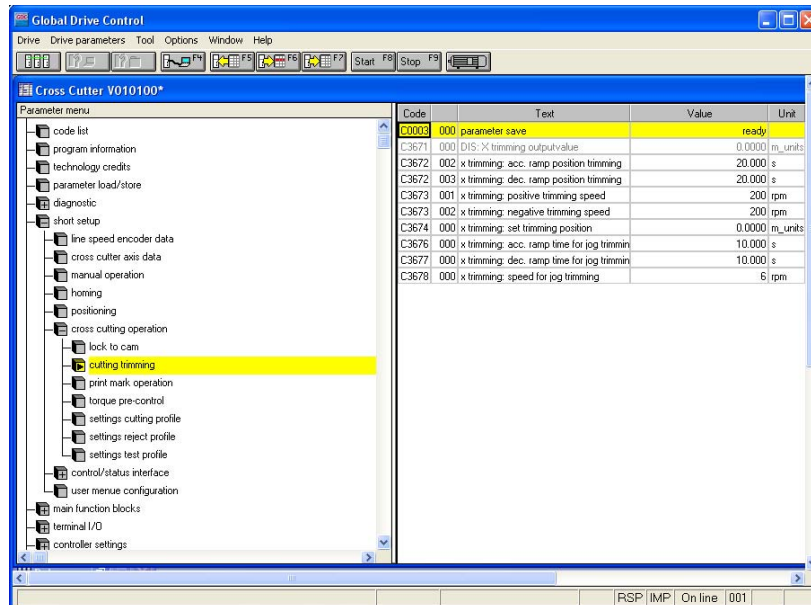
- C3630/000 Positioning mode Lock To Cam
- 0= Move to target position via shortest way.
 - 1 = Move to target position only in positive direction (CW).
 - 2 = Move to target position only in negative direction (CCW).

Cross Cutter

Installing and starting up the "CrossCutter" prepared solution

Step 10

Now make the parameter settings for X trimming
(Short Setup → Cross Cutting Operation → Cutting Trimming):



TIP!

See Chapter 2.5.11 for more information about X trimming.

Make the following code settings:

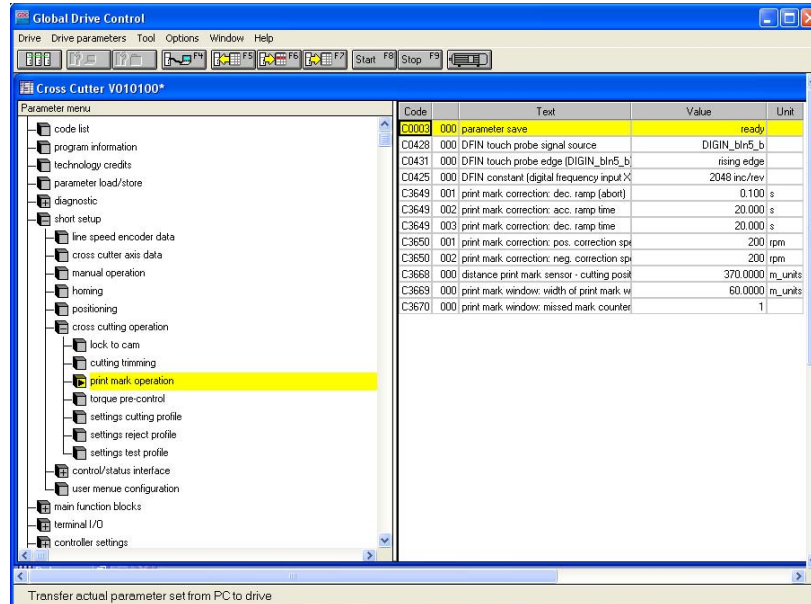
- C3672/002 X trimming: Acceleration ramp position trimming
- C3672/003 X trimming: Deceleration ramp position trimming
- C3673/001 X trimming: Positive positioning speed
- C3673/002 X trimming: Negative positioning speed
- C3674/000 X trimming: Selection of trimming position
- C3676/000 X trimming: Acceleration ramp time inching trimming
- C3677/000 X trimming: Deceleration ramp time inching trimming
- C3678/000 X trimming: Inching speed

Cross Cutter

Installing and starting up the "CrossCutter" prepared solution

Step 11

Now set the specific parameters for mark-controlled operation
(Short Setup → Cross Cutting Operation → Print Mark Operation):



TIP!

See Chapter 2.5.9 for more information about mark-controlled operation.

Make the following code settings:

C0428/000 DFIN Touch-probe source
 C0431/000 DFIN Touch-probe edge (DIGIN_bln5_b)
 C3649/001 Mark correction: Deceleration ramp (abort)
 C3649/002 Mark correction: Acceleration ramp correction profile
 C3649/003 Mark correction: Deceleration ramp correction profile
 C3650/001 Mark correction: Positive correction speed
 C3650/002 Mark correction: Negative correction speed
 C3669/000 Mark window: Width of mark window
 C3670/000 Mark window: Limit value error mark counter



TIP!

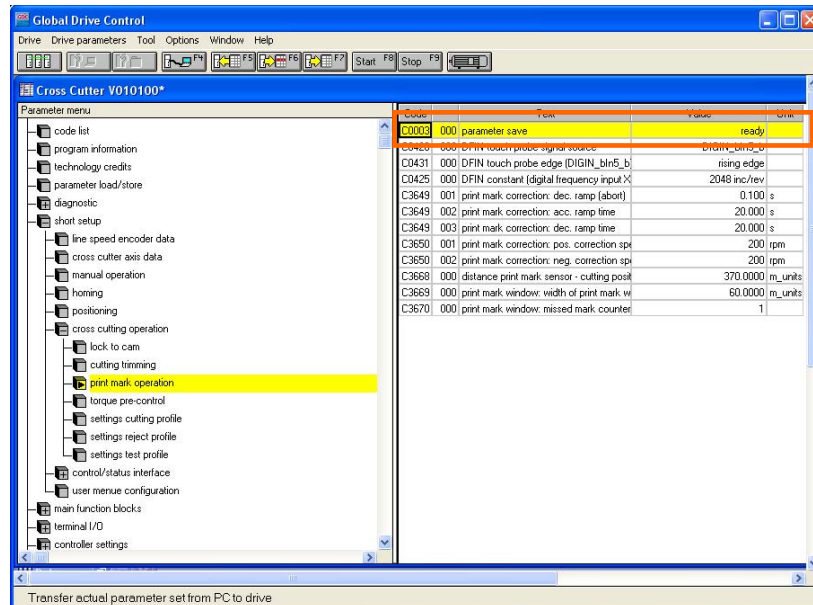
If you have selected mark-controlled operation without a ramp generator (C4135/000 application control word bit 22 deactivated), the speed set in C3650/001 will be used as the correction speed!

Cross Cutter

Installing and starting up the "CrossCutter" prepared solution

Step 12

Now proceed as follows to save the parameter set:

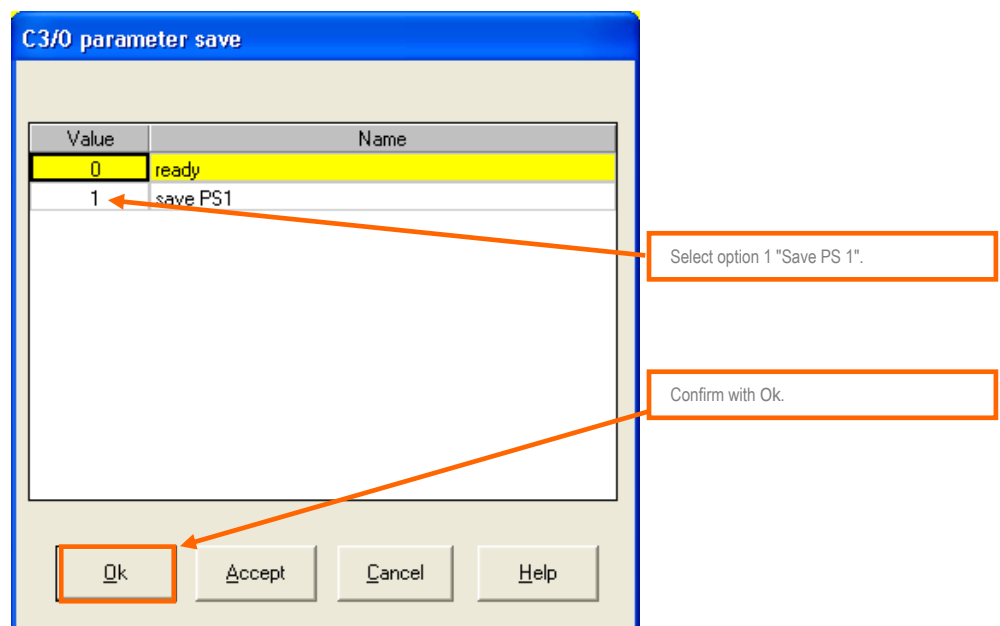


Click code C0003 to save the parameter set:

C0003/000 Save parameter set

Step 12

The input dialog for code C0003 will appear.



The parameter settings have now been saved to non-volatile memory on the controller.

This completes the basic settings for the drive functions.

End

Cross Cutter

Installing and starting up the "CrossCutter" prepared solution

3.4.5 Starting up length-controlled operation



Caution!

The start-up sequence for length-controlled operation of the prepared solution described below relates to the parameterisable variant.

Start

Step 1

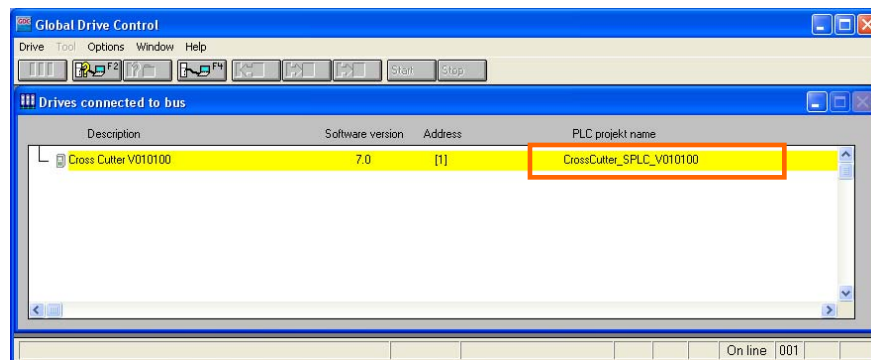
Requirements:

- The cross-cutter axis servo controller has been parameterised in accordance with the installation and start-up instructions as described in Chapter 3.4.4.
- Control voltage is being supplied to the axis via terminals 59 (+24 V DC) and 39 (0 V) and power via terminals L1, L2 and L3.

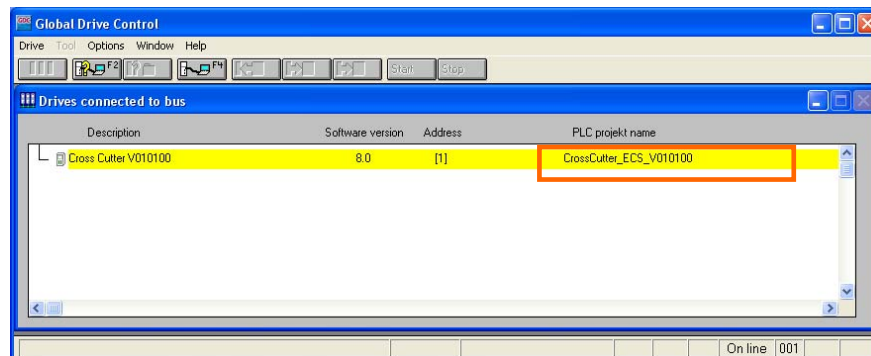
Initial situation:

- No errors are being indicated by the servo axis.
- The servo axis has been enabled.
- The application control and status words have not been customised by amending the values in C4010/0xx and C4012/0xx.
- Axis homing has been completed successfully.

Start GDC and run a search for the controllers connected to the system bus. GDC will find the connected controllers. These will include the controller loaded with the *CrossCutter_SPLC_Vxxxxxx* for 9300 Servo PLC and *CrossCutter_ECS_Vxxxxxx* for ECS.



Prepared Solution for 9300 Servo PLC



Prepared Solution for ECS

Double-click the cross-cutter axis drive and read back the parameters from the drive.

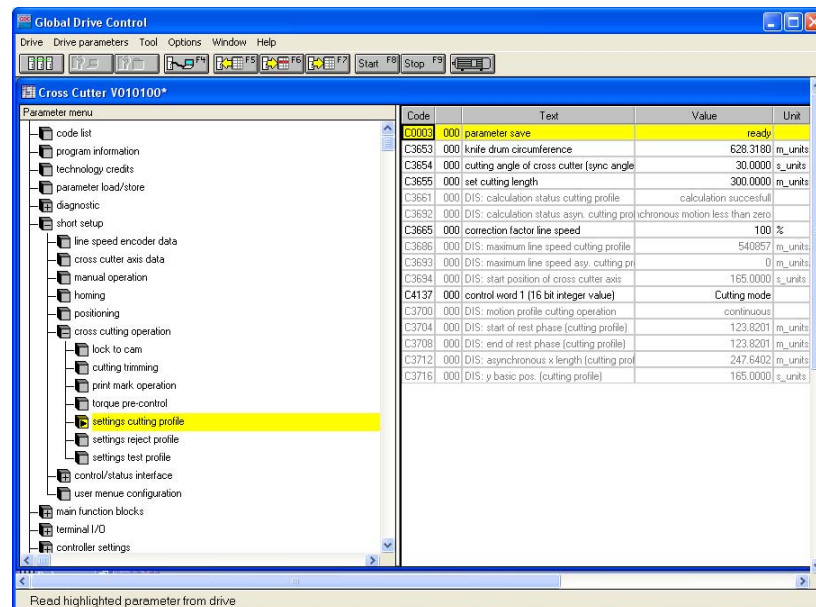
Cross Cutter

Installing and starting up the "CrossCutter" prepared solution

Step 2

Now set the specific parameters for length-controlled operation.

Start with the settings for the cutting profile (Short Setup → Cross Cutting Operation → Settings Cutting Profile):



Make the following code settings:

C3655/000 Setpoint cutting length
Enter the setpoint length here at which you wish the continuous material to be processed.
The value is entered in [m_units].



Caution!

Entering a cutting length of 0 is impermissible and will lead to an error when calculating the motion profile!



TIP!

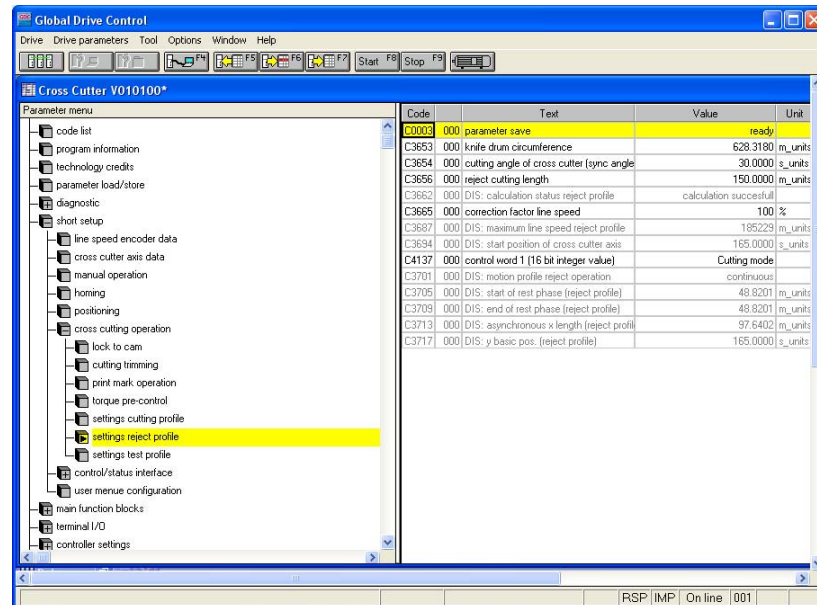
See Chapter 2.5.1 for more information.

Cross Cutter

Installing and starting up the "CrossCutter" prepared solution

Step 3

Continue by making the settings for the reject profile (Short Setup → Cross Cutting Operation → Settings Reject Profile):



Make the following code settings:

C3656/000 Reject length
Enter the setpoint length here to be cut when reject operation is selected. The value is entered in [m_units].



Caution!

Entering a cutting length of 0 is impermissible and will lead to an error when calculating the motion profile!



TIP!

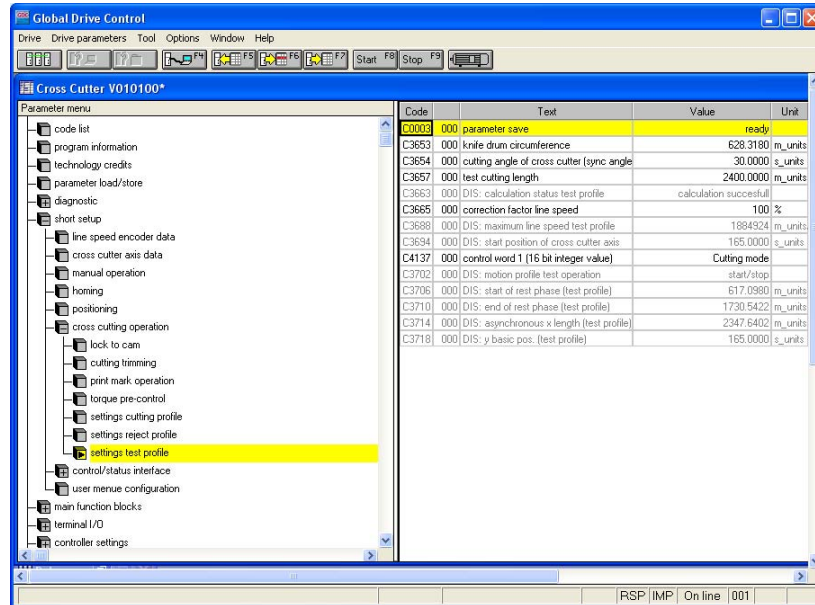
See Chapter 2.5.1 and Chapter 2.5.12 for more information about reject operation.

Cross Cutter

Installing and starting up the "CrossCutter" prepared solution

Step 4

You now need to set the parameters for the test profile (Short Setup → Cross Cutting Operation Settings Test Profile): →



Make the following code settings:

C3657/000 Cutting length test arc
Enter the setpoint length of the test arc here.
The value is entered in [m_units].



Caution!

Entering a cutting length of 0 is impermissible and will lead to an error when calculating the motion profile!



TIP!

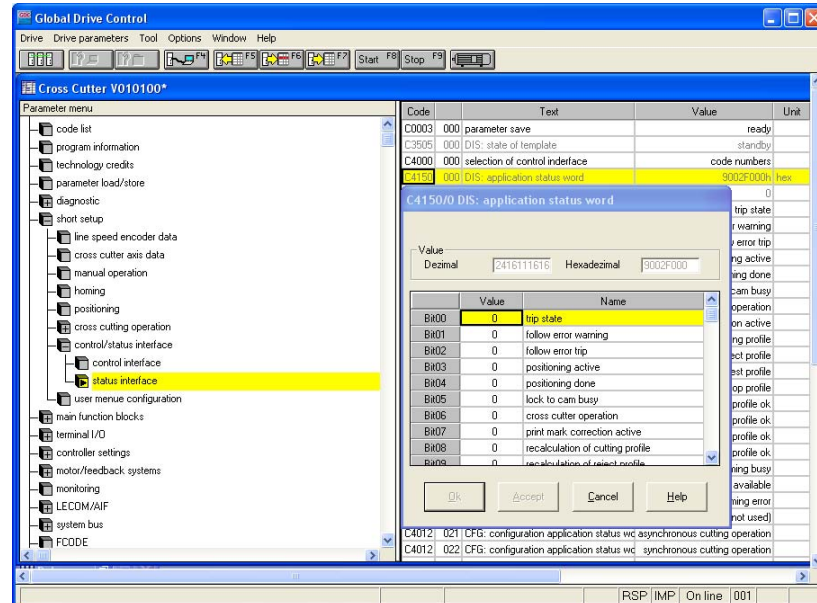
See Chapter 2.5.1 for more information about test operation.

Cross Cutter

Installing and starting up the "CrossCutter" prepared solution

Step 5

You now need to check, in the application status word, that the conditions for starting the cross cutter in length-controlled operation have been met (Short Setup → Control/Status Interface → Status Interface):



Click code C4150/000 to open the application status word.

The following bits have to be set to start length-controlled operation:

- Bit 12 Cutting profile calculation OK
- Bit 13 Reject profile calculation OK
- Bit 14 Test profile calculation OK
- Bit 15 Start/stop profile calculation OK
- Bit 17 Home position detected
- Bit 31 Profile calculation complete

The following bits must not be set to start length-controlled operation:

- Bit 0 Error status
- Bit 1 Following error warning
- Bit 2 Following error disconnection
- Bit 18 Homing error

Check the following code:

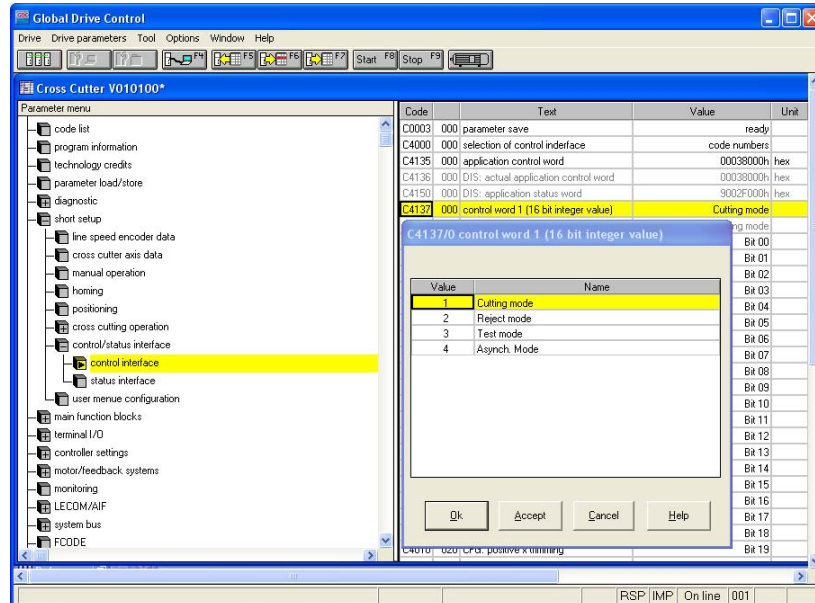
- C3505/000 The system must be in standby (10).

Cross Cutter

Installing and starting up the "CrossCutter" prepared solution

Step 6

You now need to switch to the application control word (Short Setup → Control/Status Interface → Control Interface):



Select the operating mode you wish to use in code C4137/000:

- 1 Cutting mode
- 2 Reject mode
- 3 Test mode
- 4 Asynchr. mode



TIP!

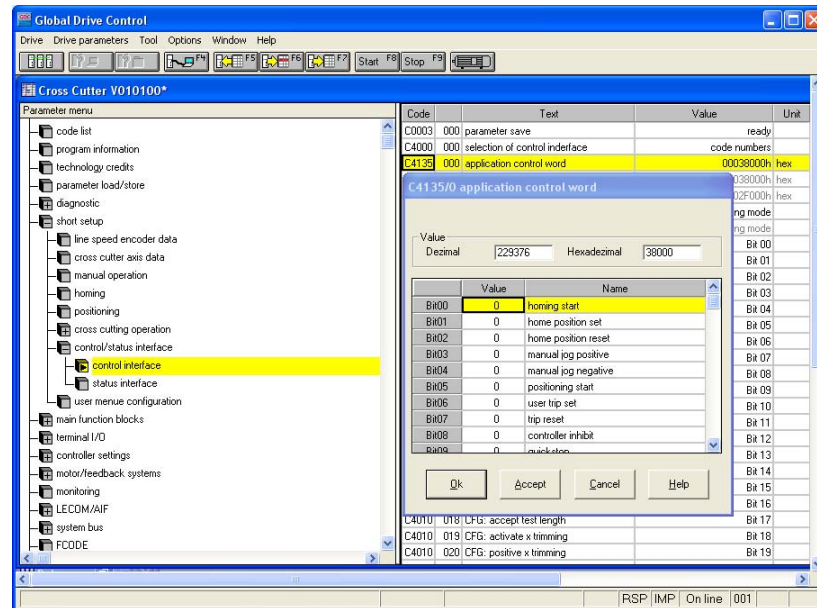
- You can switch between operating modes 1 to 3 even during cross-cutter operation.
- You cannot switch from one of operating modes 1 to 3 to operating mode 4 during operation. In order to switch modes, the system must be in standby (C3505/000=Standby(10)).
- You cannot switch from operating mode 4 to one of operating modes 1 to 3 during operation. In order to switch modes, the system must be in standby (C3505/000=Standby(10)).

Cross Cutter

Installing and starting up the "CrossCutter" prepared solution

Step 7

Click code C4135/000 to open the application control word.



Set the following bits in the order indicated below to start length-controlled operation:

- Bit 15 Accept the newly calculated cutting length
- Bit 16 Accept the newly calculated reject length
- Bit 17 Accept the newly calculated test length
- Bit 14 Start/stop the cross cutter

If the cross cutter is not in the initial position, the cross-cutter drum will execute an additional movement to reach the initial position (starting position). This process is displayed in code C3505/000 with the message "lock to cam busy operation (51)".

C3505/000 lock to cam busy operation (51)

Once the cross cutter reaches the initial position, or if it is already there, the system will switch to cam operation; it is now coupled to the master value. This state is displayed in code C3505/000 with the message cam operation . If a master value is set at this point via the DFIN signal, the cross cutter will follow it.

C3505/000 cam operation (50)

Reset the following bit to stop length-controlled operation:

- Bit 14 Start/stop the cross cutter

Once bit 14 has been reset, the cross-cutter knife will move to its initial position (starting position) and stop there. As soon as the cross cutter stops in the initial position, the system will decouple from the master value. The system will return to standby and this state will be displayed in code C3505/000.

C3505/000 The system is in standby (10).

You can switch from standby to length-controlled operation at any time (see the description in Step 7).

End

Cross Cutter

Installing and starting up the "CrossCutter" prepared solution

3.4.6 Starting up mark-controlled operation



Caution!

The start-up sequence for mark-controlled operation of the prepared solution described below relates to the parameterisable variant.

Start

Requirements:

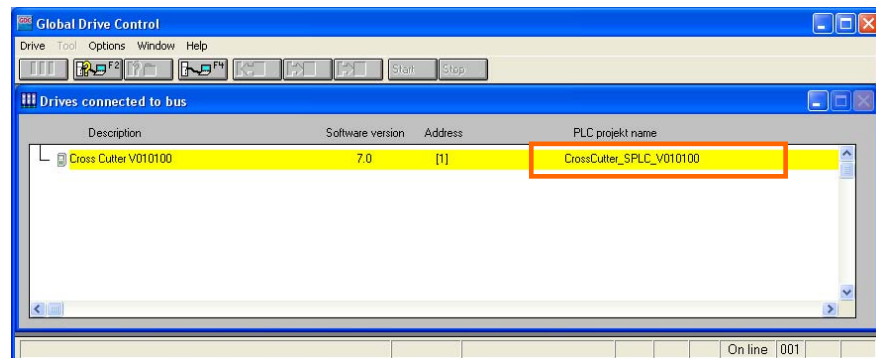
- The cross-cutter axis servo controller has been parameterised in accordance with the installation and start-up instructions as described in Chapter 3.4.4.
- Control voltage is being supplied to the axis via terminals 59 (+24 V DC) and 39 (0 V) and power via terminals L1, L2 and L3.

Initial situation:

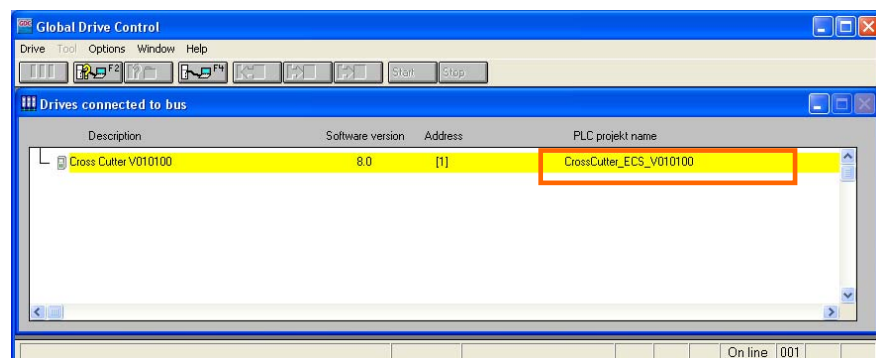
- No errors are being indicated by the servo axis.
- The servo axis has been enabled.
- The application control and status words have not been customised by amending the values in C4010/0xx and C4012/0xx.
- Axis homing has been completed successfully.
- The next print mark detected by the print-mark sensor is a valid print mark.
- The feed drive has been shut down.
- Print-mark sensor not active.

Step 1

Start GDC and run a search for the controllers connected to the system bus. GDC will find the connected controllers. These will include the controller loaded with the *CrossCutter_SPLC_Vxxxxxx* for 9300 Servo PLC and *CrossCutter_ECS_Vxxxxxx* for ECS.



Prepared Solution for 9300 Servo PLC



Prepared Solution for ECS

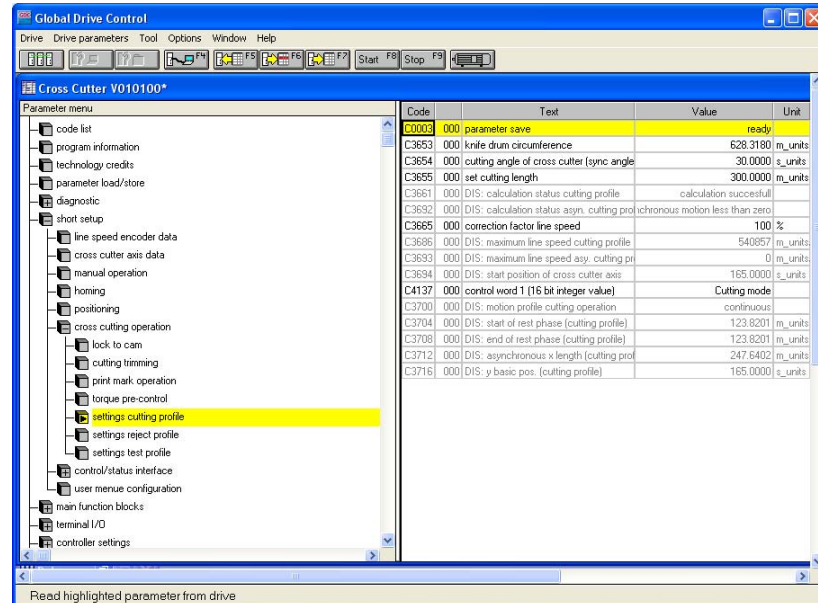
Double-click the cross-cutter axis drive and read back the parameters from the drive.

Cross Cutter

Installing and starting up the "CrossCutter" prepared solution

Step 2

Now set the parameters for the cutting profile (Short Setup → Cross Cutting Operation → Settings Cutting Profile):



Make the following code settings:

C3655/000 Setpoint cutting length
Enter the setpoint length here at which you wish the continuous material to be processed.
The value is entered in [m_units].



Caution!

Entering a cutting length of 0 is impermissible and will lead to an error when calculating the motion profile!



TIP!

See Chapter 2.5.1 and Chapter 2.5.8 for more information.



Caution!

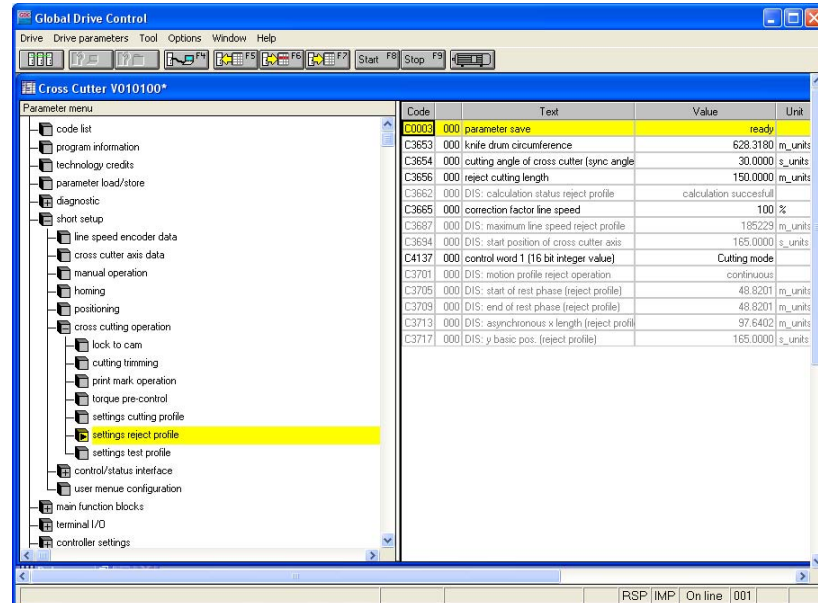
You are not permitted to change the setpoint cutting length during mark-controlled operation because the cutting length is usually equivalent to the print-mark distance or the print-mark distance is a multiple of the cutting length.

Cross Cutter

Installing and starting up the "CrossCutter" prepared solution

Step 3

Continue by making the settings for the reject profile (Short Setup → Cross Cutting Operation → Settings Reject Profile):



Make the following code settings:

C3656/000 Reject length
Enter the setpoint length here to be cut when reject operation is selected. The value is entered in [m_units].



Caution!

Entering a cutting length of 0 is impermissible and will lead to an error when calculating the motion profile!



TIP!

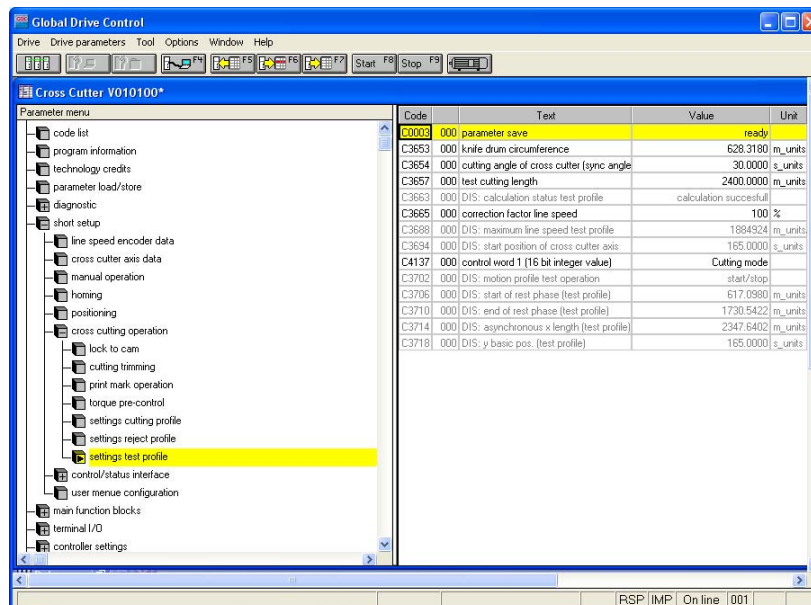
See Chapter 2.5.1 and Chapter 2.5.12 for more information about reject operation.

Cross Cutter

Installing and starting up the "CrossCutter" prepared solution

Step 4

Now set the parameters for the test profile (Short Setup → Cross Cutting Operation → Settings Test Profile):



Make the following code settings:

C3657/000 Cutting length test arc
Enter the setpoint length of the test arc here.
The value is entered in [m_units].



Caution!

Entering a cutting length of 0 is impermissible and will lead to an error when calculating the motion profile!



TIP!

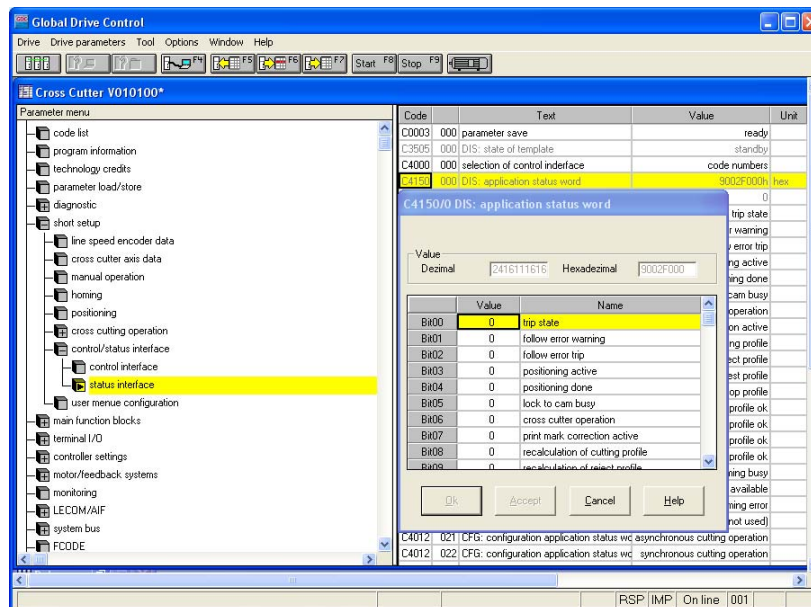
See Chapter 2.5.1 for more information about test operation.

Cross Cutter

Installing and starting up the "CrossCutter" prepared solution

Step 5

You now need to check, in the application status word, that the conditions for starting the cross cutter in mark-controlled operation have been met (Short Setup → Control/Status Interface → Status Interface):



Click code C4150/000 to open the application status word.

The following bits have to be set to start mark-controlled operation:

- Bit 12 Cutting profile calculation OK
- Bit 13 Reject profile calculation OK
- Bit 14 Test profile calculation OK
- Bit 15 Start/stop profile calculation OK
- Bit 17 Home position detected
- Bit 31 Profile calculation complete

The following bits must not be set to start mark-controlled operation:

- Bit 0 Error status
- Bit 1 Following error warning
- Bit 2 Following error disconnection
- Bit 18 Homing error

Check the following code:

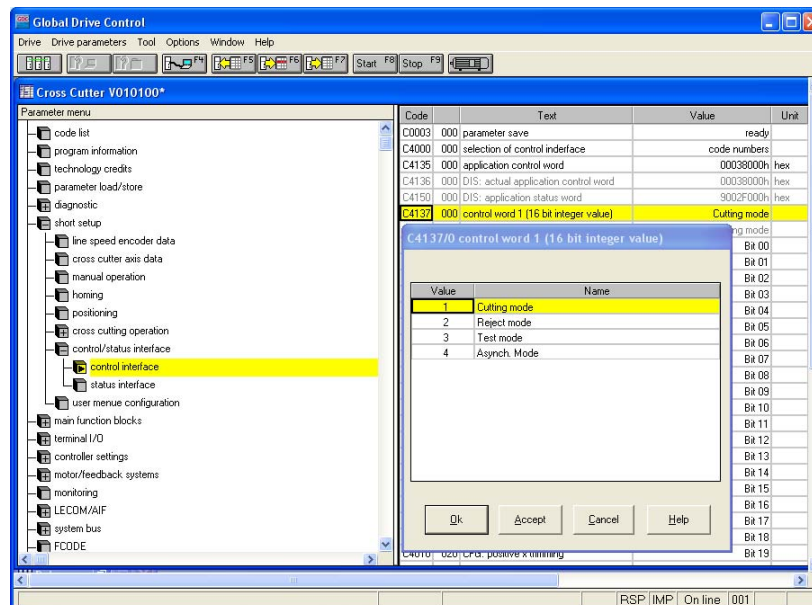
- C3505/000 The system must be in standby (10).

Cross Cutter

Installing and starting up the "CrossCutter" prepared solution

Step 6

You now need to switch to the application control word (Short Setup → Control/Status Interface → Control Interface):



Select the operating mode you wish to use in code C4137/000:

- 1 Cutting mode
- 2 Reject mode
- 3 Test mode
- 4 Asynch. mode



TIP!

- You cannot switch from operating mode 1 to operating mode 4 during operation. In order to switch modes, the system must be in standby (C3505/000=Standby(10)).
- You cannot switch from operating mode 4 to operating mode 1 during operation. In order to switch modes, the system must be in standby (C3505/000=Standby(10)).



Caution!

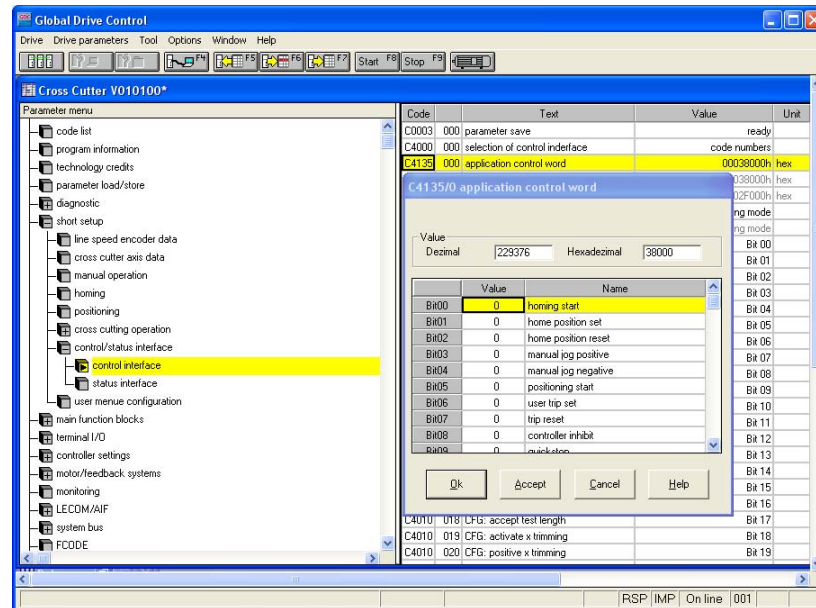
Mark-controlled operation is only supported by the system in operating modes 1 and 4. In all other operating modes, mark-controlled operation is suppressed automatically by the system!

Cross Cutter

Installing and starting up the "CrossCutter" prepared solution

Step 7

Click code C4135/000 to open the application control word.



Set the following bits in the order indicated below to activate mark-controlled operation:

- Bit 15 Accept the newly calculated cutting length
- Bit 16 Accept the newly calculated reject length
- Bit 17 Accept the newly calculated test length
- Bit 21 Activate mark control

Set bit 22 of the application control word (C4135/000) in accordance with whether you wish to use mark-controlled operation with or without a ramp generator (see Chapter 2.5.9).

- Bit 22 0 = Ramp generator deactivated
- Bit 22 1 = Ramp generator activated

You now need to set the following bit to start mark-controlled operation:

- Bit 14 Start/stop the cross cutter

If the cross cutter is not in the initial position, the cross-cutter drum will execute an additional movement to reach the initial position (starting position). This process is displayed in code C3505/000 with the message "lock to cam busy".

C3505/000 lock to cam busy operation (51)

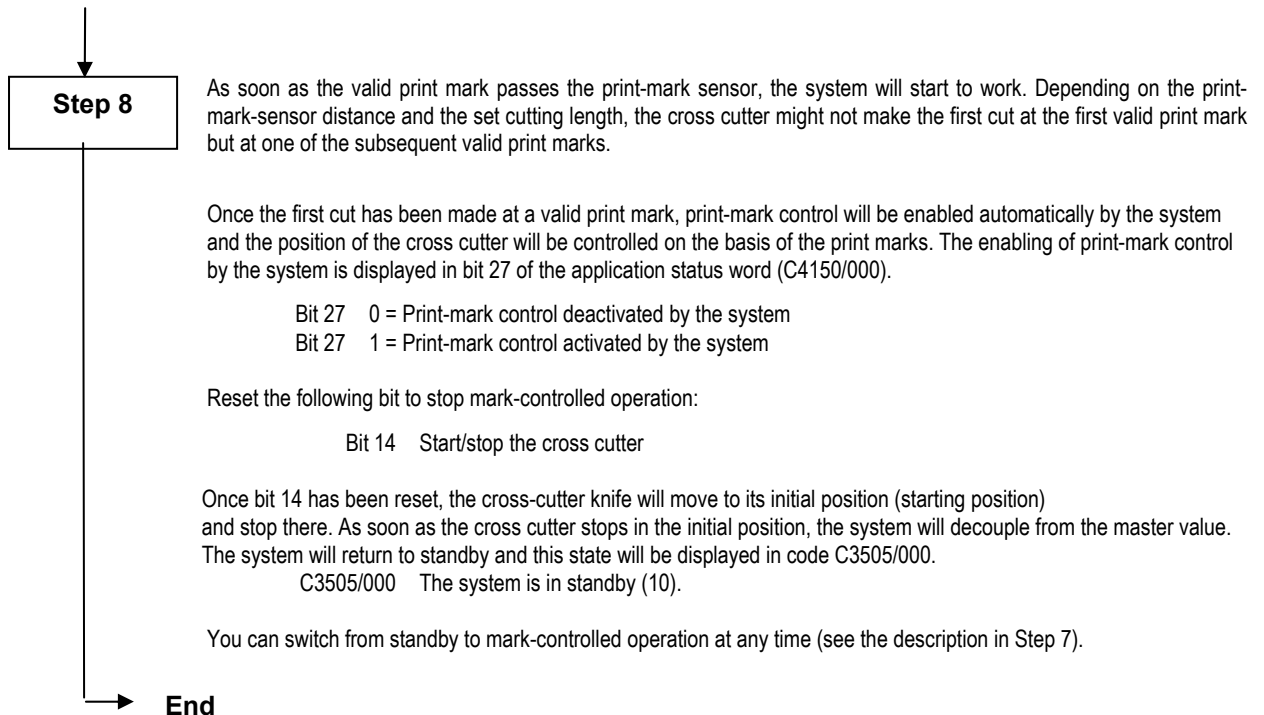
Once the cross-cutter knife reaches the initial position, or if it is already there, the system will switch to "cam operation"; it is now coupled to the master value. This state is displayed in code C3505/000 with the message "cam operation". If a master value is set at this point via the DFIN signal, the cross cutter will follow it.

C3505/000 cam operation (50)

You can now start the feed drive!

Cross Cutter

Installing and starting up the "CrossCutter" prepared solution



4 Functions

4.1 State machine

All system states are mapped on what is known as a state machine. The state machine displays the various states and explains the transitions between the individual states. It supports a total of six states, which you can switch between as follows:

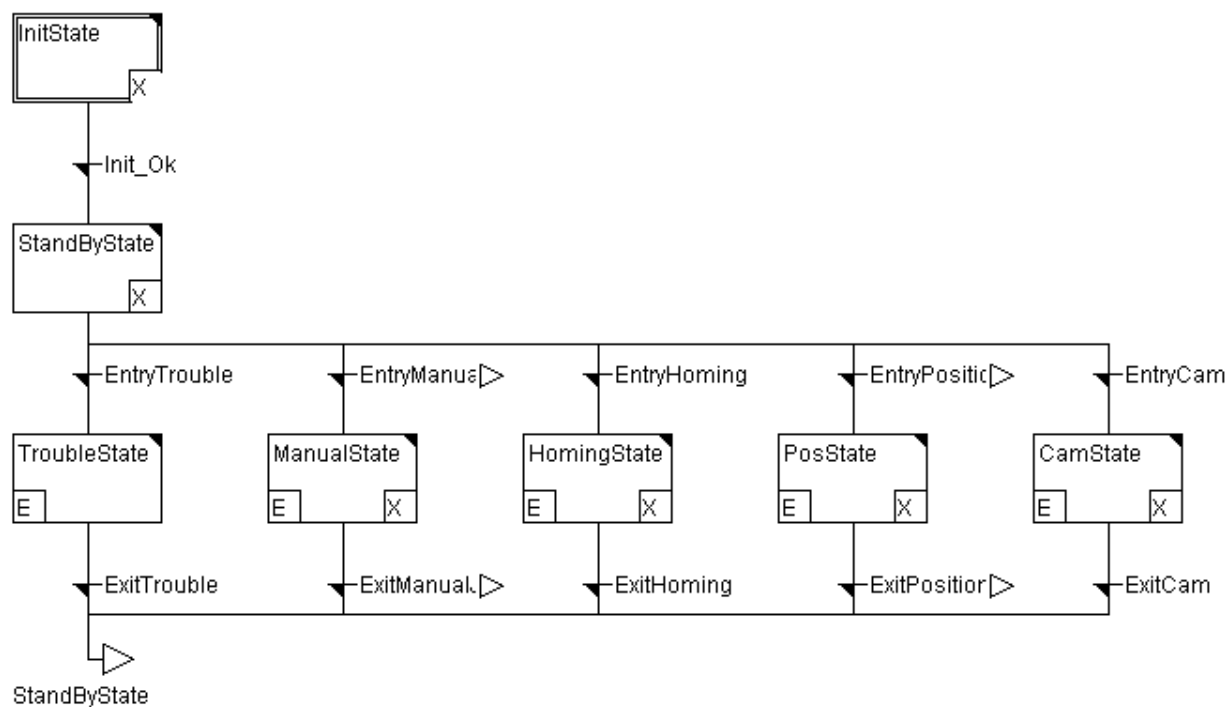


Figure 42: The state machine and its various states

4.2 Conditions for state change

State changes are subject to the following requirements:

"Standby" state:

- One of the other states is being exited or aborted.
- An error occurs. (The system will switch to the "Trouble" state immediately at this point.)

"ManualJog" state:

- The system must be in standby.
- No errors are pending.
- A command is sent to traverse the axis manually (e.g. g_bManualJogPos).

"Homing" state:

- The system must be in standby.
- No errors are pending.
- The global variable for starting homing (g_bHomingStart) is set.

"Positioning" state:

- The system must be in standby.
- No errors are pending.
- The axis must be homed, i.e. homing must have been performed successfully previously in order for the home position to be detected.
- Positioning is started via global variable g_bMotionStart.

"Cam" state:

- The system must be in standby.
- No errors are pending.
- The axis must be homed, i.e. homing must have been performed successfully previously in order for the home position to be detected.
- The "LockToCam" command must be sent with global variable g_bLockToCam.

"Trouble" state:

- An error is pending (g_bGlobalError=TRUE).

4.3 "StandBy" state

The system must be in this state in order to switch to one of the other states, e.g. Positioning. Direct switching from e.g. ManualJog to Cam is not possible. State changes must always be performed via the StandBy state. Whenever a state such as ManualJog or Cam is aborted, the system will revert to StandBy. This is illustrated in Figure 42, where you can see that the system always switches to StandBy.

4.4 "ManualJog" state

In "ManualJog", you can traverse the drive and/or tool by means of manual operator inputs. In this state, the drive moves independently of the motion profile. This type of function is used, for example, to clean the tool or feed in new material.

4.5 "Homing" state

In the "Homing" state, you can perform homing in order to map the position of the tool on the PLC and declare the zero position (home position) to the system. Several homing functions are available to the user for this purpose.

4.6 "Positioning" state

In the "Positioning" state you can move the drive to a defined position independently of the motion profile, e.g. for maintenance work on the tool. The link between the drive and the motion profile can be restored once such work is complete.

4.7 "Cam" state

The "Cam" state is the state in which the axis follows the motion profile. The traversing of the drive on the motion profile is controlled the digital frequency entry. Initially, the motion profile is activated by sending the "Lock To Cam" command. In this state it is possible to switch between individual motion profiles.

4.8 "Trouble" state

The target system switches to the "Trouble" state to report a trip or in other words a FAIL-QSP. The condition for the change to/from the "Trouble" state is not generated in the state machine itself but via a POU working in parallel with the state machine. The user can create further error messages in Program_Task3 in the UserErrors POU. Chapter 6.3.3 explains how to create user errors.

5 Program expansions

Program expansions can be added if you are using the programmable variant of the prepared solution. The program is processed using Lenze's Drive PLC Developer Studio (DDS) software version V2.2 or higher. The following chapters assume basic knowledge of the IEC programming languages and PLC configuration.

5.1 Password of the Prepared Solution Cross Cutter

Using the programmable variant of the prepared solution cross cutter the user is asked for a password by starting the programm master *CrossCutter_SPLC_Vxxxxxx.lpc* for 9300 Servo PLC and *CrossCutter_ECS_Vxxxxxx.lpc*. The user has permission for Level 2, here you just have to press the OK Button and for Level 1, the password is lenze.



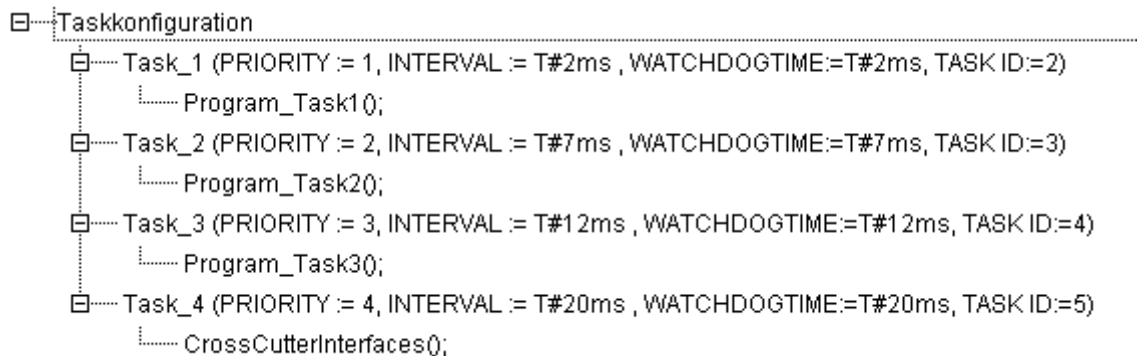
TIP!

Password Level 2: without typing in any password press the OK Button.

Password Level 1: lenze

5.2 Task management

In the programmable variant of the prepared solution, the various functions are called in different tasks on the basis of execution priority. The following tasks are standard tasks supplied ready for use:



Task 1 contains the main core of the application. Even if the user expands the program, this must always have the highest execution priority and is called in the 2 ms interval. All drive-based time-critical functions are implemented in the main core.

Task 2 calls program sections which work independently of the motion profile. These include manual jog, positioning and homing. The state machine and error management are also executed in this task.

Task 3 calls the section of the program for creating user-defined errors.

Task 4 is reserved for the multiplexer which controls the control and status interface.

Cyclic task: Calculations are made in the cyclic task (PLC PRG).



TIP!

The user can add more tasks and call these in user-defined programs. The following conditions must be observed in this context:

- No task may assume a higher priority and a shorter interval time than tasks 1 and 2.
 - Additional networks can be added to existing program sections without modification as long as the processing times of the calling task do not lead to task overflows.
-

5.3 *UserErrors* PRG

The *UserErrors* POU allows users to generate their own error messages. The exact procedure is described in Chapter 6.3.3.

5.4 *CrossCutterInterfaces* PRG

Whereas in the parameterisable variant of the prepared solution interfaces are predefined by fixed assignments of the control/status signals, in the programmable variant, the interfaces between the core functionality and the user layer can be freely defined. To achieve this, the predefined signal combinations can be modified, deleted or expanded in the *CrossCutterInterfaces* PRG.

6 Appendix

6.1 Global variables

6.1.1 VarCrossCutterBasics

Inputs		
Global variables (Variable type: VAR_GLOBAL)		
Name	Data type	Value/Meaning
g_bUseTestsignal	BOOL	Activate the DFIN simulation signal g_nTestsignalDFIN_v for test purposes.
		TRUE Test signal in use. Signals from DFIN will be ignored.
		FALSE Test signal not in use. Signals from DFIN used for master value generation.
g_nTestsignalDFIN_v	INT	Speed signal for the simulation of the DFIN signal for test purposes. This value is only used if the g_bUseTestsignal global variable = TRUE.
g_bImperialUnits	BOOL	Switch-over between metric and imperial measuring systems
		TRUE DDS visualisation is working in [inch].
		FALSE DDS visualisation is working in [mm].
g_bInvertDFINSignal	BOOL	Invert DFIN signal.
		TRUE Inverting incoming signals from the DFIN signal.
		FALSE Not inverting incoming signals from the DFIN signal.
g_dnCamPositionSet	DINT	Current position of motion profile of cross-cutter drum in 0.0001[s_units].
g_dnToolPositionAct	DINT	Current position of cross-cutter drum in 0.0001[s_units].
g_dnVertShaftPosAct	DINT	Current position of master value in 0.0001[m_units].
g_dnFollowError	DINT	Current following error in 0.0001[s_units].

6.1.2 VarCrossCutterMotionProfileCalculation

Inputs			
Global variables (Variable type: VAR_GLOBAL)			
Name	Data type	Value/Meaning	
g_dnSetCuttingLength	DINT	Cutting length in 0.0001[m_units]	
g_dnSetRejectLength	DINT	Reject length in 0.0001[m_units]	
g_dnSetTestLength	DINT	Test length in 0.0001[m_units]	
g_dnSetStartStopLength	DINT	Start/stop length in 0.0001[m_units]	
g_dnCircumferenceSlave	DINT	Cutting circle circumference of knife drum in 0.0001[m_units]: <u>Example:</u> The cutting circle diameter of the knife drum is 127.3240[mm]. The circumference is calculated by multiplying the diameter by π (= 3.141592654 ...) and results in a value of 400.0000[m_units]. You should therefore assign a value of 4000000 to variable g_dnCircumferenceSlave. <u>Note:</u> Do not enter the synchronous cutting length here (this may not be the same as the cutting circle circumference on knife drums with multiple knives). Always enter the full circumference of the knife cutting circle here.	
g_dnSyncAngle	DINT	Cutting angle for which the knife has to run in synchronism with the material in 0.0001[s_units] <u>Example:</u> A dnSyncAngle value = 300000 corresponds to a cutting angle of 30.0000[s_units] = 30.0000 [°]. - If you use this variable, to avoid inconsistencies, do not write variable g_dnSyncAngle_p!	
g_dnSyncAngle_p	DINT	Cutting angle for which the knife has to run in synchronism with the material in [incr.] If you use this variable, to avoid inconsistencies, do not write variable g_dnSyncAngle!	
g_byProfileTypeCalculation	BYTE	Selection of type of motion for the asynchronous motion phase:	
		0	Second order polynomial (linear speed ramps)
		1	Fifth order polynomial
		2	Sloping sine line
		The input values are limited to this value range!	
g_nAdjustCuttingSpeed	INT	Adaptation of the cutting speed. The value is entered in %. 100% = 16384	

Outputs		
Global variables (Variable type: VAR_GLOBAL)		
Name	Data type	Value/Meaning
g_wSetCamCuttingProfileNoForCalc	WORD	Display indicating the motion profile which is being used/will be used to calculate the cutting length.
g_wSetCamRejectProfileNoForCalc	WORD	Display indicating the motion profile which is being used/will be used to calculate the reject length.
g_wSetCamTestProfileNoForCalc	WORD	Display indicating the motion profile which is being used/will be used to calculate the test length.
g_wSetCamStartStopProfileNoForCalc	WORD	Display indicating the motion profile which is being used/will be used to calculate the start/stop length.
g_wActCamCuttingProfileNoCalc	WORD	Display indicating the last motion profile used to calculate the cutting length.
g_wActCamRejectProfileNoCalc	WORD	Display indicating the last motion profile used to calculate the reject length.
g_wActCamTestProfileNoCalc	WORD	Display indicating the last motion profile used to calculate the test length.
g_wActCamStartStopProfileNoCalc	WORD	Display indicating the last motion profile used to calculate the start/stop length.
g_dnActCuttingLength	DINT	Display indicating the cutting length used to calculate the current motion profile (g_wActCamCuttingProfileNoCalc). The value is displayed in 0.0001[m_units].
g_dnActCuttingLength_p	DINT	Display indicating the cutting length used to calculate the current motion profile (g_wActCamCuttingProfileNoCalc). The value is displayed in [incr.].

Cross Cutter Appendix

Outputs		
Global variables		(Variable type: VAR_GLOBAL)
Name	Data type	Value/Meaning
g_dnActRejectLength	DINT	Display indicating the reject length used to calculate the current motion profile (<i>g_wActCamRejectProfileNoCalc</i>). The value is displayed in 0.0001[m_units].
g_dnActTestLength	DINT	Display indicating the test length used to calculate the current motion profile (<i>g_wActCamTestProfileNoCalc</i>). The value is displayed in 0.0001[m_units].
g_dnActStartStopLength	DINT	Display indicating the start/stop length used to calculate the current motion profile (<i>g_wActCamStartStopProfileNoCalc</i>). The value is displayed in 0.0001[m_units].
g_dnCuttingLengthLimit	DINT	Output indicating the maximum cutting length in 0.0001[m_units] possible when the cross cutter is in continuous duty. Selecting a longer cutting length will lead to a motion profile with a break in the asynchronous motion phase. Note: The value of this output signal is only recalculated during a calculation sequence (<i>g_bCrossCutterCalcBusyCuttingProfile</i> = TRUE).
g_dnRejectLengthLimit	DINT	Output indicating the maximum cutting length in 0.0001[m_units] possible when the cross cutter is in continuous duty. Selecting a longer reject length will lead to a motion profile with a break in the asynchronous motion phase. Note: The value of this output signal is only recalculated during a calculation sequence (<i>g_bCrossCutterCalcBusyRejectProfile</i> = TRUE).
g_dnTestLengthLimit	DINT	Output indicating the maximum cutting length in 0.0001[m_units] possible when the cross cutter is in continuous duty. Selecting a longer test length will lead to a motion profile with a break in the asynchronous motion phase. Note: The value of this output signal is only recalculated during a calculation sequence (<i>g_bCrossCutterCalcBusyTestProfile</i> = TRUE).
g_dnStartStopLengthLimit	DINT	Output indicating the maximum cutting length in 0.0001[m_units] possible when the cross cutter is in continuous duty. Selecting a longer start/stop length will lead to a motion profile with a break in the asynchronous motion phase. Note: The value of this output signal is only recalculated during a calculation sequence (<i>g_bCrossCutterCalcBusyStartStopProfile</i> = TRUE).
g_dnXAsynchronousCuttingLength	DINT	Output indicating the material length in 0.0001[m_units] to which the cross cutter is set in the asynchronous motion phase. The value is the difference between the cutting length and the synchronous X length. Note: The value of this output signal is only recalculated during a calculation sequence (<i>g_bCrossCutterCalcBusyCuttingProfile</i> = TRUE).
g_dnXAsynchronousRejectLength	DINT	Output indicating the material length in 0.0001[m_units] to which the cross cutter is set in the asynchronous motion phase. The value is the difference between the reject length and the synchronous X length. Note: The value of this output signal is only recalculated during a calculation sequence (<i>g_bCrossCutterCalcBusyRejectProfile</i> = TRUE).
g_dnXAsynchronousTestLength	DINT	Output indicating the material length in 0.0001[m_units] to which the cross cutter is set in the asynchronous motion phase. The value is the difference between the test length and the synchronous X length. Note: The value of this output signal is only recalculated during a calculation sequence (<i>g_bCrossCutterCalcBusyTestProfile</i> = TRUE).
g_dnXAsynchronousStartStopLength	DINT	Output indicating the material length in 0.0001[m_units] to which the cross cutter is set in the asynchronous motion phase. The value is the difference between the start/stop length and the synchronous X length. Note: The value of this output signal is only recalculated during a calculation sequence (<i>g_bCrossCutterCalcBusyStartStopProfile</i> = TRUE).

Cross Cutter

Appendix

Outputs		
Global variables		(Variable type: VAR_GLOBAL)
Name	Data type	Value/Meaning
g_dnTotalYFeedAngleCuttingLength	DINT	Display indicating the Y final value of the motion profile selected via <i>g_wActCamCuttingProfileNoCalc</i> in 0.0001[s_units]. The value is taken from the Y final value of the basic profile defined with the CamDesigner. <u>Note:</u> The value of this output signal is only recalculated during a calculation sequence (<i>g_bCrossCutterCalcBusyCuttingProfile</i> = TRUE).
g_dnTotalYFeedAngleCuttingLength_p	DINT	Display indicating the Y final value of the motion profile selected via <i>g_wActCamCuttingProfileNoCalc</i> in 0.0001[s_units]. The value is taken from the Y final value of the basic profile defined with the CamDesigner. <u>Note:</u> The value of this output signal is only recalculated during a calculation sequence (<i>g_bCrossCutterCalcBusyCuttingProfile</i> = TRUE).
g_dnTotalYFeedAngleRejectLength	DINT	Display indicating the Y final value of the motion profile selected via <i>g_wActCamRejectProfileNoCalc</i> in 0.0001[s_units]. The value is taken from the Y final value of the basic profile defined with the CamDesigner. <u>Note:</u> The value of this output signal is only recalculated during a calculation sequence (<i>g_bCrossCutterCalcBusyRejectProfile</i> = TRUE).
g_dnTotalYFeedAngleTestLength	DINT	Display indicating the Y final value of the motion profile selected via <i>g_wActCamTestProfileNoCalc</i> in 0.0001[s_units]. The value is taken from the Y final value of the basic profile defined with the CamDesigner. <u>Note:</u> The value of this output signal is only recalculated during a calculation sequence (<i>g_bCrossCutterCalcBusyTestProfile</i> = TRUE).
g_dnTotalYFeedAngleStartStopLength	DINT	Display indicating the Y final value of the motion profile selected via <i>g_wActCamStartStopProfileNoCalc</i> in 0.0001[s_units]. The value is taken from the Y final value of the basic profile defined with the CamDesigner. <u>Note:</u> The value of this output signal is only recalculated during a calculation sequence (<i>g_bCrossCutterCalcBusyStartStopProfile</i> = TRUE).
g_dnX1CalcCuttingProfile	DINT	Display indicating the starting point of the pause in 0.0001[m_units] only if <i>g_bStartStopOperationCuttingProfile</i> = TRUE.
g_dnX1CalcRejectProfile	DINT	Display indicating the starting point of the pause in 0.0001[m_units] only if <i>g_bStartStopOperationRejectProfile</i> = TRUE.
g_dnX1CalcTestProfile	DINT	Display indicating the starting point of the pause in 0.0001[m_units] only if <i>g_bStartStopOperationTestProfile</i> = TRUE.
g_dnX1CalcStartStopProfile	DINT	Display indicating the starting point of the pause in 0.0001[m_units] only if <i>g_bStartStopOperationStartStopProfile</i> = TRUE.
g_dnX1CalcStartStopProfile_p	DINT	Display indicating the starting point of the pause in [incr.] only if <i>g_bStartStopOperationStartStopProfile</i> = TRUE.
g_dnX2CalcCuttingProfile	DINT	Display indicating the finishing point of the pause in 0.0001[m_units] only if <i>g_bStartStopOperationCuttingProfile</i> = TRUE.
g_dnX2CalcRejectProfile	DINT	Display indicating the finishing point of the pause in 0.0001[m_units] only if <i>g_bStartStopOperationRejectProfile</i> = TRUE.
g_dnX2CalcTestProfile	DINT	Display indicating the finishing point of the pause in 0.0001[m_units] only if <i>g_bStartStopOperationTestProfile</i> = TRUE.
g_dnX2CalcStartStopProfile	DINT	Display indicating the finishing point of the pause in 0.0001[m_units] only if <i>g_bStartStopOperationStartStopProfile</i> = TRUE.
g_dnY1Value	DINT	Display indicating the starting point of the cross cutter in 0.0001[s_units].
g_bStartStopOperationCuttingProfile	BOOL	Status signal: Cutting length induces start/stop operation of cross cutter.
		TRUE The selected cutting length exceeds the limit cutting length for start/stop operation => The pause will be extended accordingly in the asynchronous motion phase.
		FALSE The selected cutting length undershoots the limit cutting length for start/stop operation.

Cross Cutter Appendix

Outputs		
Global variables		(Variable type: VAR_GLOBAL)
Name	Data type	Value/Meaning
g_bStartStopOperationRejectProfile	BOOL	Status signal: Reject length induces start/stop operation of cross cutter.
		TRUE The selected reject length exceeds the limit cutting length for start/stop operation => The pause will be extended accordingly in the asynchronous motion phase.
		FALSE The selected reject length undershoots the limit cutting length for start/stop operation.
g_bStartStopOperationTestProfile	BOOL	Status signal: Test length induces start/stop operation of cross cutter.
		TRUE The selected test length exceeds the limit cutting length for start/stop operation => The pause will be extended accordingly in the asynchronous motion phase.
		FALSE The selected test length undershoots the limit cutting length for start/stop operation.
g_bStartStopOperationStartStopProfile	BOOL	Status signal: Start/stop length induces start/stop operation of cross cutter.
		TRUE The selected start/stop length exceeds the limit cutting length for start/stop operation => The pause will be extended accordingly in the asynchronous motion phase.
		FALSE The selected start/stop length undershoots the limit cutting length for start/stop operation.
g_wTotalNumberOfPointsCuttingProfile	WORD	Display indicating the number of interpolation points on the motion profile selected via g_wActCamCuttingProfileNoCalc. Note: The value of this output signal is only recalculated during a calculation sequence (g_bCrossCutterCalcBusyCuttingProfile = TRUE).
g_wTotalNumberOfPointsRejectProfile	WORD	Display indicating the number of interpolation points on the motion profile selected via g_wActCamRejectProfileNoCalc. Note: The value of this output signal is only recalculated during a calculation sequence (g_bCrossCutterCalcBusyRejectProfile = TRUE).
g_wTotalNumberOfPointsTestProfile	WORD	Display indicating the number of interpolation points on the motion profile selected via g_wActCamTestProfileNoCalc. Note: The value of this output signal is only recalculated during a calculation sequence (g_bCrossCutterCalcBusyTestProfile = TRUE).
g_wTotalNumberOfPointsStartStopProfile	WORD	Display indicating the number of interpolation points on the motion profile selected via g_wActCamStartStopProfileNoCalc. Note: The value of this output signal is only recalculated during a calculation sequence (g_bCrossCutterCalcBusyStartStopProfile = TRUE).
g_wTotalNumberOfPointsAsyCuttingProfile	WORD	Display indicating the number of interpolation points on the motion profile for the asynchronous cutting. Note: The value of this output signal is only recalculated during a calculation sequence (g_bCrossCutterCalcBusyAsyCuttingProfile = TRUE).
g_wActNumberOfPointsCuttingProfile	WORD	Display indicating the most recently calculated interpolation point on the motion profile selected via g_wActCamCuttingProfileNoCalc; this output can be used as a progress indicator, for example. The value range is 0 ... (g_wTotalNumberOfPointsCuttingProfile - 1).
g_wActNumberOfPointsRejectProfile	WORD	Display indicating the most recently calculated interpolation point on the motion profile selected via g_wActCamRejectProfileNoCalc; this output can be used as a progress indicator, for example. The value range is 0 ... (g_wTotalNumberOfPointsRejectProfile - 1).

Cross Cutter Appendix

Outputs		
Global variables		(Variable type: VAR_GLOBAL)
Name	Data type	Value/Meaning
g_wActNumberOfPointsTestProfile	WORD	Display indicating the most recently calculated interpolation point on the motion profile selected via <i>g_wActCamTestProfileNoCalc</i> ; this output can be used as a progress indicator, for example. The value range is 0 ... (<i>g_wTotalNumberOfPointsTestProfile</i> - 1).
g_wActNumberOfPointsStartStopProfile	WORD	Display indicating the most recently calculated interpolation point on the motion profile selected via <i>g_wActCamStartStopProfileNoCalc</i> ; this output can be used as a progress indicator, for example. The value range is 0 ... (<i>g_wTotalNumberOfPointsStartStopProfile</i> - 1).
g_wActNumberOfPointsAsyCuttingProfile	WORD	Display indicating the most recently calculated interpolation point on the motion profile for the asynchronous cutting, this output can be used as a progress indicator, for example. The value range is 0 ... (<i>g_wTotalNumberOfPointsAsyCuttingProfile</i> - 1).
g_bCrossCutterCalcBusyCuttingProfile	BOOL	Status signal: Calculation sequence for new cross-cutter motion profile for a new cutting length in progress.
		TRUE Calculation sequence for cross-cutter motion profile for a new cutting length in progress.
		FALSE Calculation sequence for new cross-cutter motion profile for a new cutting length completed successfully or aborted.
g_bCrossCutterCalcBusyRejectProfile	BOOL	Status signal: Calculation sequence for new cross-cutter motion profile for a new reject length in progress.
		TRUE Calculation sequence for cross-cutter motion profile for a new reject length in progress.
		FALSE Calculation sequence for new cross-cutter motion profile for a new reject length completed successfully or aborted.
g_bCrossCutterCalcBusyTestProfile	BOOL	Status signal: Calculation sequence for new cross-cutter motion profile for a new test length in progress.
		TRUE Calculation sequence for cross-cutter motion profile for a new test length in progress.
		FALSE Calculation sequence for new cross-cutter motion profile for a new test length completed successfully or aborted.
g_bCrossCutterCalcBusyStartStopProfile	BOOL	Status signal: Calculation sequence for new cross-cutter motion profile for a new start/stop length in progress.
		TRUE Calculation sequence for cross-cutter motion profile for a new start/stop length in progress.
		FALSE Calculation sequence for new cross-cutter motion profile for a new start/stop length completed successfully or aborted.
g_bCrossCutterCalcBusyAsyCuttingProfile	BOOL	Status signal: Calculation sequence for new cross-cutter motion profile for a new asynchronous length in progress.
		TRUE Calculation sequence for cross-cutter motion profile for a new asynchronous length in progress.
		FALSE Calculation sequence for new cross-cutter motion profile for a new asynchronous length completed successfully or aborted.

Cross Cutter Appendix

Outputs		
Global variables		(Variable type: VAR_GLOBAL)
Name	Data type	Value/Meaning
g_bCrossCutterCalcDoneCuttingProfile	BOOL	Status signal: Calculation sequence for new cross-cutter motion profile for a new cutting length completed successfully.
		TRUE Calculation sequence for new cross-cutter motion profile for a new cutting length completed successfully. Starting a new calculation sequence will reset this status output to FALSE.
		FALSE Calculation sequence for new cross-cutter motion profile for a new cutting length not completed.
g_bCrossCutterCalcDoneRejectProfile	BOOL	Status signal: Calculation sequence for new cross-cutter motion profile for a new reject length completed successfully.
		TRUE Calculation sequence for new cross-cutter motion profile for a new reject length completed successfully. Starting a new calculation sequence will reset this status output to FALSE.
		FALSE Calculation sequence for new cross-cutter motion profile for a new reject length not completed.
g_bCrossCutterCalcDoneTestProfile	BOOL	Status signal: Calculation sequence for new cross-cutter motion profile for a new test length completed successfully.
		TRUE Calculation sequence for new cross-cutter motion profile for a new test length completed successfully. Starting a new calculation sequence will reset this status output to FALSE.
		FALSE Calculation sequence for new cross-cutter motion profile for a new test length not completed.
g_bCrossCutterCalcDoneStartStopProfile	BOOL	Status signal: Calculation sequence for new cross-cutter motion profile for a new start/stop length completed successfully.
		TRUE Calculation sequence for new cross-cutter motion profile for a new start/stop length completed successfully. Starting a new calculation sequence will reset this status output to FALSE.
		FALSE Calculation sequence for new cross-cutter motion profile for a new start/stop length not completed.
g_bCrossCutterCalcDoneAsyCuttingProfile	BOOL	Status signal: Calculation sequence for new cross-cutter motion profile for a new asynchronous length completed successfully.
		TRUE Calculation sequence for new cross-cutter motion profile for a new asynchronous length completed successfully. Starting a new calculation sequence will reset this status output to FALSE.
		FALSE Calculation sequence for new cross-cutter motion profile for a new asynchronous length not completed.

Cross Cutter Appendix

Outputs		
Global variables		(Variable type: VAR_GLOBAL)
Name	Data type	Value/Meaning
g_nCrossCutterCalcStateCuttingProfile	INT	Status of calculation sequence for new cutting length.
		0 Calculation ok.
		1 - No LC9 file found. - Profile data not found in absolute data model. - Read error when accessing profile data.
		2 Selected motion profile not found.
		4 The angle of motion of the knife drum for the asynchronous motion phase is less than or equal to zero (check y final value of basic profile and cutting angle!).
		8 A value of zero has been entered for the knife drum's cutting circle circumference.
		16 The preset cutting length is less than the material length corresponding to the cutting angle.
		32 The selected motion profile contains fewer than 20 interpolation points.
		64 The selected motion profile does not start at zero (0.0000/0.0000).
		128 Cutting angle too small.
		256 The y final value of the basic profile does not correspond to value $360.0000[^\circ]/n_{\text{outs/rev}}$. ($n_{\text{outs/rev}}$ = Number of knives on knife drum, values of 1, 2, 3, 4, 5 or 6 are possible)
		512 Internal read access to code C0011 failed.
		If the nState status variable is exhibiting non-zero values, the calculated variables may exhibit invalid values, in which case it will not be possible to use them subsequently in the control program!

Cross Cutter Appendix

Outputs		
Global variables		(Variable type: VAR_GLOBAL)
Name	Data type	Value/Meaning
g_nCrossCutterCalcStateRejectProfile	INT	Status of calculation sequence for new reject length.
		0 Calculation ok.
		1 - No LC9 file found. - Profile data not found in absolute data model. - Read error when accessing profile data.
		2 Selected motion profile not found.
		4 The angle of motion of the knife drum for the asynchronous motion phase is less than or equal to zero (check y final value of basic profile and cutting angle!).
		8 A value of zero has been entered for the knife drum's cutting circle circumference.
		16 The preset cutting length is less than the material length corresponding to the cutting angle.
		32 The selected motion profile contains fewer than 20 interpolation points.
		64 The selected motion profile does not start at zero (0.0000/0.0000).
		128 Cutting angle too small.
		256 The y final value of the basic profile does not correspond to value $360.0000[^\circ]/n_{\text{outs/rev}}$. ($n_{\text{outs/rev}}$ = Number of knives on knife drum, values of 1, 2, 3, 4, 5 or 6 are possible)
		512 Internal read access to code C0011 failed.
		If the nState status variable is exhibiting non-zero values, the calculated variables may exhibit invalid values, in which case it will not be possible to use them subsequently in the control program!

Cross Cutter Appendix

Outputs		
Global variables		(Variable type: VAR_GLOBAL)
Name	Data type	Value/Meaning
g_nCrossCutterCalcStateTestProfile	INT	Status of calculation sequence for new test length.
		0 Calculation ok.
		1 - No LC9 file found. - Profile data not found in absolute data model. - Read error when accessing profile data.
		2 Selected motion profile not found.
		4 The angle of motion of the knife drum for the asynchronous motion phase is less than or equal to zero (check y final value of basic profile and cutting angle!).
		8 A value of zero has been entered for the knife drum's cutting circle circumference.
		16 The preset cutting length is less than the material length corresponding to the cutting angle.
		32 The selected motion profile contains fewer than 20 interpolation points.
		64 The selected motion profile does not start at zero (0.0000/0.0000).
		128 Cutting angle too small.
		256 The y final value of the basic profile does not correspond to value $360.0000[^\circ]/n_{\text{outs/rev}}$. ($n_{\text{outs/rev}}$ = Number of knives on knife drum, values of 1, 2, 3, 4, 5 or 6 are possible)
		512 Internal read access to code C0011 failed.
		If the nState status variable is exhibiting non-zero values, the calculated variables may exhibit invalid values, in which case it will not be possible to use them subsequently in the control program!

Cross Cutter Appendix

Outputs		
Global variables		(Variable type: VAR_GLOBAL)
Name	Data type	Value/Meaning
g_nCrossCutterCalcStateStartStopProfile	INT	Status of calculation sequence for new start/stop length.
		0 Calculation ok.
		1 - No LC9 file found. - Profile data not found in absolute data model. - Read error when accessing profile data.
		2 Selected motion profile not found.
		4 The angle of motion of the knife drum for the asynchronous motion phase is less than or equal to zero (check y final value of basic profile and cutting angle!).
		8 A value of zero has been entered for the knife drum's cutting circle circumference.
		16 The preset cutting length is less than the material length corresponding to the cutting angle.
		32 The selected motion profile contains fewer than 20 interpolation points.
		64 The selected motion profile does not start at zero (0.0000/0.0000).
		128 Cutting angle too small.
		256 The y final value of the basic profile does not correspond to value $360.0000[^\circ]/n_{\text{outs/rev}}$. ($n_{\text{outs/rev}}$ = Number of knives on knife drum, values of 1, 2, 3, 4, 5 or 6 are possible)
		512 Internal read access to code C0011 failed.
		If the nState status variable is exhibiting non-zero values, the calculated variables may exhibit invalid values, in which case it will not be possible to use them subsequently in the control program!

Cross Cutter Appendix

Outputs		
Global variables		(Variable type: VAR_GLOBAL)
Name	Data type	Value/Meaning
g_nCrossCutterCalcStateAsyCuttingProfile	INT	Status of calculation sequence for new asynchronous cutting length.
		0 Calculation ok.
		1 - No LC9 file found. - Profile data not found in absolute data model. - Read error when accessing profile data.
		2 Selected motion profile not found.
		4 The angle of motion of the knife drum for the asynchronous motion phase is less than or equal to zero (check y final value of basic profile and cutting angle!).
		8 A value of zero has been entered for the knife drum's cutting circle circumference.
		16 The preset cutting length is less than the material length corresponding to the cutting angle.
		32 The selected motion profile contains fewer than 20 interpolation points.
		64 The selected motion profile does not start at zero (0.0000/0.0000).
		128 Cutting angle too small.
		256 The y final value of the basic profile does not correspond to value $360.0000[^\circ]/n_{\text{outs/rev}}$. ($n_{\text{outs/rev}}$ = Number of knives on knife drum, values of 1, 2, 3, 4, 5 or 6 are possible)
		512 Internal read access to code C0011 failed.
		If the nState status variable is exhibiting non-zero values, the calculated variables may exhibit invalid values, in which case it will not be possible to use them subsequently in the control program!

Cross Cutter Appendix

Outputs		
Global variables (Variable type: VAR_GLOBAL)		
Name	Data type	Value/Meaning
g_bAllCalcDone	BOOL	Status signal: All calculation sequences for new cross-cutter motion profiles have been completed successfully.
		TRUE All calculation sequences for new cross-cutter motion profiles have been completed successfully. Starting a new calculation sequence will reset this status output to FALSE.
		FALSE All calculation sequences for new cross-cutter motion profiles are still in progress.
g_bAnotherRoundCalcCuttingProfile	BOOL	Status signal: The motion profile for the cutting length needs to be recalculated due to changes to input values for the calculation.
		TRUE Need to recalculate.
		FALSE Do not need to recalculate.
g_bAnotherRoundCalcRejectProfile	BOOL	Status signal: The motion profile for the reject length needs to be recalculated due to changes to input values for the calculation.
		TRUE Need to recalculate.
		FALSE Do not need to recalculate.
g_bAnotherRoundCalcTestProfile	BOOL	Status signal: The motion profile for the test length needs to be recalculated due to changes to input values for the calculation.
		TRUE Need to recalculate.
		FALSE Do not need to recalculate.
g_bAnotherRoundCalcStartStopProfile	BOOL	Status signal: The motion profile for the start/stop length needs to be recalculated due to changes to input values for the calculation.
		TRUE Need to recalculate.
		FALSE Do not need to recalculate.
g_nMaxMasterVelForActCuttingProfile_v	INT	Entry indicating the maximum possible line speed for the set cutting length in [incr./ms].
g_nMaxMasterVelForActRejectProfile_v	INT	Entry indicating the maximum possible line speed for the set reject length in [incr./ms].
g_nMaxMasterVelForActTestProfile_v	INT	Entry indicating the maximum possible line speed for the set test length in [incr./ms].
g_nMaxMasterVelForActStartStopProfile_v	INT	Entry indicating the maximum possible line speed for the set start/stop length in [incr./ms].
g_nMaxMasterVelForActAsyCuttingProfile_v	INT	Entry indicating the maximum possible line speed for the set asynchronous cutting length in [incr./ms].
g_dnDistanceToXMax	DINT	Distance between finishing point of motion profile and 6 o'clock position on knife in 0.0001[m_units]. If you use this variable, to avoid inconsistencies, do not write variable g_dnDistanceToXMax_p!
g_dnDistanceToXMax_p	DINT	Distance between finishing point of motion profile and 6 o'clock position on knife in 0.0001[incr.]. If you use this variable, to avoid inconsistencies, do not write variable g_dnDistanceToXMax!
g_nNumberOfKnives	INT	Number of knives mounted on the knife drum.

6.1.3 VarCrossCutterMultiplexer

Inputs		
Global variables (Variable type: VAR_GLOBAL)		
Name	Data type	Value/Meaning
g_byCtrlMode	BYTE	Selection of control interface the user wishes to use
		0 Control via code
		1 Control via CAN 1
		2 Control via CAN 2
		3 Control via CAN 3
		4 Control via AIF 1
		5 Control via AIF 2
		6 Control via AIF 3

6.1.4 VarCrossCutterPieceCounter

Inputs		
Global variables (Variable type: VAR_GLOBAL)		
Name	Data type	Value/Meaning
g_bResetCuttingProfileCounter	BOOL	Reset cutting counter for cutting operation.
		TRUE Reset counter.
		FALSE Counter will not be reset.
g_bResetRejectProfileCounter	BOOL	Reset cutting counter for reject operation.
		TRUE Reset counter.
		FALSE Counter will not be reset.
g_bResetTestProfileCounter	BOOL	Reset cutting counter for test operation.
		TRUE Reset counter.
		FALSE Counter will not be reset.
g_wSetValuePiecesOfCuttingProfile	WORD	Setpoint cutting quantity for cutting operation
g_wSetValuePiecesOfTestProfile	WORD	Setpoint cutting quantity for reject operation
g_wSetValuePiecesOfRejectProfile	WORD	Setpoint cutting quantity for test operation

Outputs		
Global variables (Variable type: VAR_GLOBAL)		
Name	Data type	Value/Meaning
g_bSetValueOfCuttingPiesesCuttingProfileReached	BOOL	Status signal: The set cutting quantity has been reached for cutting operation.
		TRUE Set cutting quantity reached.
		FALSE Set cutting quantity not reached.
g_bSetValueOfCuttingPiesesRejectProfileReached	BOOL	Status signal: The set cutting quantity has been reached for reject operation.
		TRUE Set cutting quantity reached.
		FALSE Set cutting quantity not reached.
g_bSetValueOfCuttingPiesesTestProfileReached	BOOL	Status signal: The set cutting quantity has been reached for test operation.
		TRUE Set cutting quantity reached.
		FALSE Set cutting quantity not reached.
g_wCounterValueCuttingProfile	WORD	Cutting counter for cutting operation The counter is limited to a maximum value of 65535; on reaching this value it will overflow to 0.
g_wCounterValueRejectProfile	WORD	Cutting counter for reject operation The counter is limited to a maximum value of 65535; on reaching this value it will overflow to 0.
g_wCounterValueTestProfile	WORD	Cutting counter for test operation The counter is limited to a maximum value of 65535; on reaching this value it will overflow to 0.

6.1.5 VarCrossCutterProfileHandling

Inputs		
Global variables (Variable type: VAR_GLOBAL)		
Name	Data type	Value/Meaning
g_bStartStopOperation	BOOL	Start and stop the selected cross-cutter operating mode.
		TRUE Start operation.
		FALSE Stop operation.
g_bAbordCrossCutter	BOOL	Activate the variable to abort cross-cutter operation.
		TRUE Abort cross-cutter operation.
		FALSE Normal operation
g_wProfileSelector	WORD	Select cross-cutter operating mode. Note: You can switch between modes 1 to 3 during operation. To switch to mode 4 from one of modes 1 to 3, you will first need to stop the cross cutter via <i>g_bStartStopOperation</i> . To switch from mode 4 to one of modes 1 to 3, you will first need to stop the cross cutter via <i>g_bStartStopOperation</i> .
		1 Synchronous cross cutter, continuous cutting
		2 Synchronous cross cutter, reject operation
		3 Synchronous cross cutter, test operation
		4 Asynchronous cross cutter, continuous cutting
g_bWorkWithNewCuttingLength	BOOL	Activate the variable to immediately apply a new cutting length you have calculated and work with the new cutting length. Note: The variable is only valid for the synchronous cross cutter! A new cutting length for the asynchronous cross cutter is always applied immediately!
		TRUE Apply new cutting length calculated.
		FALSE Do not apply new cutting length calculated, continue to work with the last cutting length calculated.
g_bWorkWithNewRejectLength	BOOL	Activate the variable to immediately apply a new reject length you have calculated and work with the new reject length.
		TRUE Apply new reject length calculated.
		FALSE Do not apply new reject length calculated, continue to work with the last reject length calculated.
g_bWorkWithNewTestLength	BOOL	Activate the variable to immediately apply a new test length you have calculated and work with the new test length.
		TRUE Apply new test length calculated.
		FALSE Do not apply new test length calculated, continue to work with the last test length calculated.
g_wCuttingProfileMain	WORD	Selection of motion profile 1 as main motion profile for cutting operation. If the cutting length changes, the system will switch between the main and standby motion profiles.
g_wCuttingProfileStandby	WORD	Selection of motion profile 2 as standby motion profile for cutting operation. If the cutting length changes, the system will switch between the main and standby motion profiles.
g_wRejectProfileMain	WORD	Selection of motion profile 3 as main motion profile for reject operation. If the reject length changes, the system will switch between the main and standby motion profiles.

Cross Cutter Appendix

Inputs		
Global variables		(Variable type: VAR_GLOBAL)
Name	Data type	Value/Meaning
g_wRejectProfileStandby	WORD	Selection of motion profile 4 as standby motion profile for reject operation. If the reject length changes, the system will switch between the main and standby motion profiles.
g_wTestProfileMain	WORD	Selection of motion profile 5 as main motion profile for test operation. If the test length changes, the system will switch between the main and standby motion profiles.
g_wTestProfileStandby	WORD	Selection of motion profile 6 as standby motion profile for test operation. If the test length changes, the system will switch between the main and standby motion profiles.
g_wStartStopProfileMain	WORD	Selection of motion profile 7 as main motion profile for start/stop operation. If the start/stop length changes, the system will switch between the main and standby motion profiles.
g_wStartStopProfileStandby	WORD	Selection of motion profile 8 as standby motion profile for start/stop operation. If the start/stop length changes, the system will switch between the main and standby motion profiles.
g_bStopDFIN	BOOL	Status signal: This signal indicates that the current position value of the master value axis is equivalent to or in excess of the start/stop position (g_dnX1CalcStartStopProfile_p) of the cross cutter.
		TRUE Start/stop position reached.
		FALSE Start/stop position not reached.

Outputs		
Global variables		(Variable type: VAR_GLOBAL)
Name	Data type	Value/Meaning
g_bUseSynchronusCrossCutter	BOOL	Status signal: The cross cutter is operating as a synchronous cross cutter.
		TRUE The cross cutter is operating as a synchronous cross cutter.
		FALSE The cross cutter is not operating as a synchronous cross cutter.
g_bUseAsynchronusCrossCutter	BOOL	Status signal: The cross cutter is operating as an asynchronous cross cutter.
		TRUE The cross cutter is operating as an asynchronous cross cutter.
		FALSE The cross cutter is not operating as an asynchronous cross cutter.
g_bUseCuttingProfile	BOOL	Status signal: The cross cutter is operating in "Continuous cutting" mode.
		TRUE The cross cutter is operating in "Continuous cutting" mode.
		FALSE The cross cutter is not operating in "Continuous cutting" mode.
g_bUseRejectProfile	BOOL	Status signal: The cross cutter is operating in "Reject operation" mode.
		TRUE The cross cutter is operating in "Reject operation" mode.
		FALSE The cross cutter is not operating in "Reject operation" mode.

Cross Cutter Appendix

Outputs			
Global variables		(Variable type: VAR_GLOBAL)	
Name	Data type	Value/Meaning	
g_bUseTestProfile	BOOL	Status signal: The cross cutter is operating in "Test operation" mode.	
		TRUE	The cross cutter is operating in "Test operation" mode.
		FALSE	The cross cutter is not operating in "Test operation" mode.
g_bStartDFIN	BOOL	Activate and deactivate speed value for master value.	
		TRUE	Send speed value.
		FALSE	Set speed to zero.
g_bResetStopDFIN	BOOL	Reset <i>g_bStopDFIN</i> variable.	
		TRUE	<i>g_bStopDFIN</i> reset.
		FALSE	<i>g_bStopDFIN</i> not reset.

6.1.6 VarCrossCutterRejectGate

Inputs		
Global variables		(Variable type: VAR_GLOBAL)
Name	Data type	Value/Meaning
g_wRejectGateSwitchingDwelltime	WORD	Selection of the switching dwell time (switching delay time) of the reject gate in ms.
g_dnDistanceRejectGate	DINT	Selection of the distance from the reject gate to the cutting position in 0.0001[m_units]. The value is converted into increments internally. If you use this variable, to avoid inconsistencies, do not write variable <i>g_dnDistanceRejectGate_p</i> !
g_dnDistanceRejectGate_p	DINT	Selection of the distance from the reject gate to the cutting position in [incr]. If you use this variable, to avoid inconsistencies, do not write variable <i>g_dnDistanceRejectGate</i> !

Outputs		
Global variables		(Variable type: VAR_GLOBAL)
Name	Data type	Value/Meaning
g_bOpenRejectGate	BOOL	This output is used to control the reject gate.
		TRUE Open the reject gate.
		FALSE Do not open the reject gate.

6.1.7 VarCrossCutterTPHandling

Inputs		
Global variables (Variable type: VAR_GLOBAL)		
Name	Data type	Value/Meaning
g_bResetCounterTPOutOfWindow	BOOL	Variable to reset the counter for mark signals located outside the mark window.
		TRUE Reset counter.
		FALSE Do not reset counter.
g_dnMeasuredTPDistance	DINT	Distance measured from mark sensor to 6 o'clock position on knife in [m_unit]. The value is entered in fixed point format with 4 decimal positions (1 [unit] = 10000). The value is converted into increments internally. If you use this variable, to avoid inconsistencies, do not write variable g_dnMeasuredTPDistance_p!
g_dnMeasuredTPDistance_p	DINT	Distance measured from mark sensor to 6 o'clock position of knife in [incr.]. If you use this variable, to avoid inconsistencies, do not write variable g_dnMeasuredTPDistance!
g_DFIN_nIn_v	INT	Speed signal from DFIN in incr./ms.
g_DFIN_dnIncLastScan_p	DINT	Δ inc between mark sensor signal and start of task. This global variable is linked directly to SB DFIN_IO_DigitalFrequency.
g_DFIN_bTPReceived_b	BOOL	Trigger input for calculating position deviation. This signal is used for incoming print-mark-sensor signals. This global variable is linked directly to SB DFIN_IO_DigitalFrequency. Digital input I5 using 9300 Servo PLC and I1 using ECS is reserved for the mark sensor!
		FALSE => TRUE A print-mark signal has been received.
		FALSE No print-mark signal detected.
g_dnTPPositionWindow	DINT	Size of mark window in 0.0001[m_units]. The value is converted into increments internally. If you use this variable, to avoid inconsistencies, do not write variable g_dnTPPositionWindow_p!
g_dnTPPositionWindow_p	DINT	Size of mark window in [incr.]. If you use this variable, to avoid inconsistencies, do not write variable g_dnTPPositionWindow!
g_wMaxNumberOfTpOutOfWindow	WORD	Number of cycles permitted without a mark signal appearing in the mark window. A warning signal is generated if this number is exceeded.

Outputs		
Global variables (Variable type: VAR_GLOBAL)		
Name	Data type	Value/Meaning
g_bTpWindowOpen	BOOL	This output indicates the time at which the window is open.
		TRUE The mark window is open. Incoming mark signals are output at g_bXTPReceived.
		FALSE The mark window is closed. Incoming mark signals are not output at g_bXTPReceived.
g_bTPWindowActive	BOOL	This output indicates that the mark window is active and the first print-mark signal has been received.
		TRUE The mark window is active.
		FALSE The mark window is not active. The first print mark has not yet been detected.

Cross Cutter Appendix

Outputs		
Global variables (Variable type: VAR_GLOBAL)		
Name	Data type	Value/Meaning
g_bOverflowTPRegister	BOOL	This output indicates a print-mark register overflow.
		TRUE Print-mark register overflow.
		FALSE No print-mark register overflow.
g_bMaxNumberOfTpOutOfWindow Increased	BOOL	This output signal indicates that the maximum number of cycles completed without a mark signal appearing in the mark window has been exceeded.
		TRUE The maximum number of cycles without a mark signal appearing in the mark window has been reached.
		FALSE The maximum number of cycles without a mark signal appearing in the mark window has not been reached.
g_bResetCounterTPOutOfWindow	BOOL	Reset counter for print marks appearing outside the mark window.
		TRUE Reset counter.
		FALSE Do not reset counter.
g_dnTPDistance_p	DINT	This output signal indicates the distance between two print marks in increments.
g_wNumberOfTpOutOfWindow	WORD	This output signal indicates the number of cycles completed without a mark signal appearing in the mark window.
g_bKnifeNotInMaterial	BOOL	"Knife not in material" status signal
		TRUE = Knife is not in material. FALSE = Knife is in material.
g_bUseShiftedTP	BOOL	This output indicates when the print-mark register needs to be used.
		TRUE The print-mark register needs to be used.
		FALSE The print-mark register does not need to be used.
g_dnMultipleTPDistance	DINT	Calculated value indicating the distance of the print-mark sensor in integer multiples of the cutting length. The value is scaled in 0.0001[m_units]. You will need this value if you are using a print-mark register. The value is converted into increments internally. If you use this variable, to avoid inconsistencies, do not write variable g_dnMultipleTPDistance_p!
g_dnMultipleTPDistance_p	DINT	Calculated value indicating the distance of the print-mark sensor in integer multiples of the cutting length. The value is scaled in [incr]. You will need this value if you are using a print-mark register. If you use this variable, to avoid inconsistencies, do not write variable g_dnMultipleTPDistance!
g_nNumberOfMissedOutTpMarks	INT	This output signal indicates the number of print-mark signals which pass the cross-cutter knife when the cross cutter starts up before the first cut is made at a print mark.
g_dnOffsetTP	INT	Position value by which the incoming mark signal to start the cross cutter must be delayed in order to ensure that the first cut is made at a valid print mark. The value is scaled in 0.0001[m_units]. The value is converted into increments internally. If you use this variable, to avoid inconsistencies, do not write variable g_dnOffsetTP_p!
g_dnOffsetTP_p	INT	Position value by which the incoming mark signal to start the cross cutter must be delayed in order to ensure that the first cut is made at a valid print mark. The value is scaled in [incr]. If you use this variable, to avoid inconsistencies, do not write variable g_dnOffsetTP!

Outputs			
Global variables		(Variable type: VAR_GLOBAL)	
Name	Data type	Value/Meaning	
g_bSystemReadyToUseTP	BOOL	This output indicates the enabling of print-mark synchronisation by the system.	
		TRUE	Print-mark synchronisation can be used.
		FALSE	Print-mark synchronisation cannot be used.

6.1.8 VarCrossCutterTrouble

Inputs			
Global variables		(Variable type: VAR_GLOBAL)	
Name	Data type	Value/Meaning	
g_bExtUserError	BOOL	User error triggered externally	
		TRUE	An external user error has been triggered.
		FALSE	No errors.

Outputs			
Global variables		(Variable type: VAR_GLOBAL)	
Name	Data type	Value/Meaning	
g_bWrongSetCamProfileNoForCalc	BOOL	Status signal: A motion profile currently being used by the application has been calculated.	
		TRUE	A motion profile currently being used by the application has been calculated.
		FALSE	The calculated motion profile is not currently being used by the application.
g_bErrorCrossCutterCuttingProfileCalcBlock	BOOL	Status signal: An error occurred whilst calculating the motion profile for the new cutting length.	
		TRUE	An error has occurred during calculation.
		FALSE	Calculation completed without errors.
g_bErrorCrossCutterRejectProfileCalcBlock	BOOL	Status signal: An error occurred whilst calculating the motion profile for the new reject length.	
		TRUE	An error has occurred during calculation.
		FALSE	Calculation completed without errors.
g_bErrorCrossCutterTestProfileCalcBlock	BOOL	Status signal: An error occurred whilst calculating the motion profile for the new test length.	
		TRUE	An error has occurred during calculation.
		FALSE	Calculation completed without errors.
g_bErrorCrossCutterStartStopProfileCalcBlock	BOOL	Status signal: An error occurred whilst calculating the motion profile for the new start/stop length.	
		TRUE	An error has occurred during calculation.
		FALSE	Calculation completed without errors.

6.1.9 VarCrossCutterXAxisTrimming

Inputs		
Global variables (Variable type: VAR_GLOBAL)		
Name	Data type	Value/Meaning
g_bUseXPositionTrimming	BOOL	Activates X trimming and switches from the calculated mark sensor distance to the mark sensor distance specified by X trimming.
		TRUE X trimming activated.
		FALSE X trimming deactivated.
g_bXAxisTrimmingPosDirection	BOOL	Key this variable to perform X trimming in a positive direction at the set speed.
		TRUE Perform X trimming in a positive direction.
		FALSE Do not perform X trimming.
g_bXAxisTrimmingNegDirection	BOOL	Key this variable to perform X trimming in a negative direction at the set speed.
		TRUE Perform X trimming in a negative direction.
		FALSE Do not perform X trimming.
g_bLoadSetXAxisTrimmingTPPos	BOOL	Perform absolute positioning at the set speed.
		TRUE Perform absolute positioning in accordance with the value of variable <i>g_dnSetXAxisTrimmingTPPos</i> .
		FALSE Abort absolute positioning.
g_nXAxisTrimmingFactor_v	INT	Selection of the inching speed for X trimming in a positive and negative direction.
g_dnSetXAxisTrimmingTPPos	DINT	Target position of absolute positioning for X trimming in 0.0001[m_units]. The value is converted into increments internally. If you use this variable, to avoid inconsistencies, do not write variable <i>g_dnSetXAxisTrimmingTPPos_p</i> !
g_dnSetXAxisTrimmingTPPos_p	DINT	Target position of absolute positioning for X trimming in [incr]. If you use this variable, to avoid inconsistencies, do not write variable <i>g_dnSetXAxisTrimmingTPPos</i> !

Outputs		
Global variables (Variable type: VAR_GLOBAL)		
Name	Data type	Value/Meaning
g_dnXAxisTrimmingTPPosOut	DINT	Displays the actual X trimming position in 0.0001[m_units]. This output indicates the mark sensor distance set by X trimming. Whenever X trimming is active, this value will serve as the mark sensor distance for mark correction. This value is displayed in code C3671/000.
g_nXAxisVelTrimmingFactor_v	INT	This variable contains the speed value generated for X trimming. It is added to the DFIN signal in order to effect an immediate change in the position value when using X trimming.

Inputs/Outputs		
Global variables (Variable type: VAR_GLOBAL)		
Name	Data type	Value/Meaning
g_dnXAxisTrimmingTPPosOut_p	DINT	This input/output indicates the mark sensor distance set by X trimming. Whenever X trimming is active, this value will serve as the mark sensor distance for mark correction. The variable is a retain variable (this means that once an X trimming value has been calculated, it will remain accessible even after mains switching).

6.1.10 VarBasicOperation

Inputs			
Global variables (Variable type: VAR_GLOBAL)			
Name	Data type	Value/Meaning	
g_wMotionProfileNumber	WORD	Preselection of motion profile • Switch-over to this profile will not take place until profile zero crossing. • Immediate switch-over can be forced via the g_bProfileForcedShift input. • 0 ... 47 Number of the motion profile (The selection range is limited to the number of motion profiles defined in the *.lc9 file.)	
g_bProfileForcedShift	BOOL	Forced switch-over of motion profile FALSE => TRUE Forced switch-over	
g_nYAxisVelocityMin_v	INT	Lower window limit to limit the speed of the cross cutter in [incr./ms]	
g_nYAxisVelocityMax_v	INT	Upper window limit to limit the speed of the cross cutter in [incr./ms]	
g_bReloadData	BOOL	Reset X axis to starting value. TRUE Resets the value for the master value axis in FB L_CamData to the starting value.	
g_bCamDataXZeroPosTrigger	BOOL	Initialisation of profile data via upstream FB L_CamSectionShift • If you use FB L_CamSectionShift, link this input to output bXZeroPosTrigger in FB L_CamSectionShift. TRUE Initialise profile data.	
g_dnVertShaftCamDataIn_p	DINT	Set phase of master value axis in [incr.] • Internal limitation to 0 ... Xmax of profile (Xmax_ + 231 - 1)	
g_bySelectMem	BYTE	Selection of data structure (*.lc9 file) in memory. 0 RAM memory block 1 (foreground data field, standard) 1 RAM memory block 2 (background data field)	

Cross Cutter Appendix

Outputs		
Global variables (Variable type: VAR_GLOBAL)		
Name	Data type	Value/Meaning
g_wMotionProfileAct	WORD	Number of current motion profile 0 ... 47 Number of motion profile
g_bVertShaftPosZero	BOOL	"Profile zero crossing" status signal TRUE Electrical shaft at profile zero crossing
g_dnVertShaftPosAct_p	DINT	Actual angle of electrical shaft in [incr.]
g_nVertShaftVelocityAct_v	INT	Current speed of electrical shaft
g_bSignGradientProfileAct	BOOL	Sign of gradient of current profile TRUE Negative sign FALSE Positive sign
g_dnVertShaftLengthAct_p	DINT	Length of X axis of current profile in [incr.]
g_bYAxisLimitBusy	BOOL	"Y axis speed limitation active" status signal Please note that as long as limitation is active, the motor will not follow the motion profile precisely! TRUE The speed of the Y axis is outside the window defined using g_nYAxisVelocityMin_v and g_nYAxisVelocityMin_v.
g_dnYAxisPosAct_p	DINT	Position of the Y axis - This variable maps the Y integrator of FB L_CamData. - The variable is used when connecting a cam group.
g_nCamDataState	INT	Status of FB L_CamData 0 No errors 10 Warning: Profile not found. 20 Warning: Excessive master value step change
g_nCamVelocitySet_v	INT	Setpoint speed of profile to be processed
g_dnYEndPos_p	DINT	Current stroke at the end of a feed profile
g_CamAxis	CAM_AXIS_REF	Profile parameter Data structure, the elements of which are written to the machine parameters in the internal measuring system.
g_ProfileValues R/W	CAM_PROFILE_REF	Data structure with the major points of the selected motion profile Positions in [incr.]

6.1.11 VarCamControl

Inputs		
Global variables (Variable type: VAR_GLOBAL)		
Name	Data type	Value/Meaning
g_nCamControlFactNSet	INT	Factor for scaling the speed precontrol value This variable is linked to g_MCTRL_nNmaxC11 (UserProcessInput POU, net 2).
g_bResetMainIntegrator	BOOL	Set cam position to actual value or zero position.
		TRUE Set cam position to actual position or zero position. Note: Please note the dependency on code C3639.
g_bClearAuxIntegrator	BOOL	Set tool position to "0"/clear. Only possible if the template is not in the CamOperation state.
		TRUE Tool position and actual position set to "0".

Outputs		
Global variables (Variable type: VAR_GLOBAL)		
Name	Data type	Value/Meaning
g_dnFollowError_p	DINT	Following error/positional deviation in [incr.] Sent to the position controller as a setpoint.
g_dnCamPositionSet_p	DINT	Position setpoint of motion profile in [incr.]
g_dnToolPositionAct_p	DINT	Calculated tool position in [incr.]
g_dnActPos_p	DINT	Relative actual position in [incr.], i.e. within the Y motion profile Position value is saved to non-volatile memory and will remain available even if a new download is performed.
g_dnActPosInSave_p	DINT	Absolute actual position in [incr.] Position value is saved to non-volatile memory and will remain available even if a new download is performed.

6.1.12 VarHoming

Inputs		
Global variables (Variable type: VAR_GLOBAL)		
Name	Data type	Value/Meaning
g_byHomingMode	BYTE	Homing mode
		0 > _Rn_MP
		1 < _Rn_MP
		4 > _Rp < _Rn_MP
		5 < _Rp > _Rn_MP
		6 > _Rn > _TP
		7 < _Rn < _TP
		8 > _TP
		9 < _TP
		Symbols: > Movement in positive direction < Movement in negative direction Rp Positive edge of homing switch Rn Negative edge of homing switch TP Touch-probe edge of input I4 for Servo PLC and I2 for ECS MP Zero pulse edge of motor The execution of the last action in each case will set the home position (e.g. the zero pulse in the case of "MP") even if the drive continues to move subsequently.
g_bHomingStart	BOOL	Start/stop homing. • Only in conjunction with controller enable.
		FALSE => TRUE Start homing.
		FALSE Stop/abort homing.
g_bHomingMark	BOOL	Input for homing switch preliminary stop position
		TRUE Depends on the selected mode.
		FALSE Depends on the selected mode.
g_dnMachineZeroDistance	DINT	Position of homing switch in [unit] • The value is entered in fixed point format with 4 decimal positions (1 [unit] = 10000). The value is converted into increments internally. If you use this variable, to avoid inconsistencies, do not write variable g_dnMachineZeroDistance_p!
g_dnMachineZeroDistance_p	DINT	Position of homing switch in [incr.] • If you use this variable, to avoid inconsistencies, do not write variable g_dnMachineZeroDistance!
g_dnHomingTargetPos	DINT	Target position of the tool once homing is complete in [s_unit] • With reference to the machine zero point. • The value is entered in fixed point format with 4 decimal positions (1 [unit] = 10000). The value is converted into increments internally. • If you use this variable, to avoid inconsistencies, do not write variable g_dnHomingTargetPos_p!
g_dnHomingTargetPos_p	DINT	Target position of the tool once homing is complete in [incr.] • With reference to the machine zero point. • If you use this variable, to avoid inconsistencies, do not write variable g_dnHomingTargetPos!
g_bHomePositionSet	BOOL	Setting of home position • Only possible if controller inhibit (CINH) set!
		FALSE => TRUE The setpoint defined in g_dnMachineZeroDistance(_p) is applied and g_bHomingDone is set to TRUE.

Cross Cutter Appendix

Inputs			
Global variables (Variable type: VAR_GLOBAL)			
Name	Data type	Value/Meaning	
g_bHomePositionReset		Reset home.	
		TRUE	The "Home position detected" status is reset (g_bHomingDone = FALSE). The g_bHomingDone output is reset to FALSE.

Outputs			
Global variables (Variable type: VAR_GLOBAL)			
Name	Data type	Value/Meaning	
g_bHomingBusy	BOOL	"Homing activated" status signal	
		TRUE	Homing active.
g_bHomingDone	BOOL	Home position status	
		FALSE	Home position not detected.
		FALSE => TRUE	g_dnHomingTargetPos(_p) reached.
		TRUE	Homing complete, home position detected.
g_bHomingError	BOOL	"Error" status signal	
		TRUE	An error occurred when setting the home position. See g_nHomingErrorNo for more detailed information about the error.
g_nHomingErrorNo	INT	Error number	
		0	OK
		1	An error has occurred during bus transport.
		2	Time-out
		4	Subindex does not exist.
		5	Index does not exist.
		13	Invalid value range
		111	More than 4 decimal positions entered or no index with fixed point format.
		117	Communication channel does not exist.
		-10	Controller inhibit (CINH) not set.

6.1.13 VarManualJog

Inputs		
Global variables (Variable type: VAR_GLOBAL)		
Name	Data type	Value/Meaning
g_bManualJogNeg	BOOL	Manual jog in negative direction (CCW)
		TRUE Manual jog CCW
g_bManualJogPos	BOOL	Manual jog in positive direction (CW)
		TRUE Manual jog CW
g_bReleaseLimitSwitch	BOOL	"Retracting" subfunction During retraction, the entries in g_bExternalJogEnable and g_nExternalJog_v are rendered ineffective.
		TRUE Retract from the limit switch activated in the corresponding retract direction, until the limit switch is re-enabled (no longer activated): - g_bLimitSwitchPos = TRUE: - Retract in g_bLimitSwitchNeg direction until g_bLimitSwitchPos = FALSE. - g_bLimitSwitchNeg = TRUE: - Retract in g_bLimitSwitchPos direction until g_bLimitSwitchNeg = FALSE.
g_bExternalJogEnable	BOOL	Enable external speed. Activation will only be possible if g_bManualJogNeg and g_bManualJogPos are both set to FALSE.
		TRUE External speed entered in g_nExternalJog_v enabled.
g_nExternalJog_v	INT	External speed entered in [incr./ms]

6.1.14 VarPositioning

Inputs			
Global variables (Variable type: VAR_GLOBAL)			
Name	Data type	Value/Meaning	
g_bMotionStart	BOOL	Execute positioning.	
		TRUE	The drive is moved to the target position g_dnToolPositionSet(_p) along the set ramps and at the set speed.
		FALSE	Positioning interrupted.
g_byPositioningMode	ENUM "PositionMode"	Positioning mode	
		0	Move to target position via shortest route.
		1	Move to target position only in positive direction (CW).
		2	Move to target position only in negative direction (CCW).
		3	Reserved for template function "PosToCam".
		4	Positioning relative to entry in g_dnToolPositionSet(_p).
g_byPositioningModeCode	BYTE	Positioning mode	
		0	Move to target position via shortest route.
		1	Move to target position only in positive direction (CW).
		2	Move to target position only in negative direction (CCW).
		3	Reserved for template function "PosToCam".
		4	Positioning relative to entry in g_dnToolPositionSet(_p).
g_dnToolPositionSet	DINT	Target position in [s_unit]	
		- With reference to the machine zero point. - The value is entered in fixed point format with 4 decimal positions (1 [unit]/10000). The value is converted into increments internally. (See the information at the bottom of the table.) - If you use this variable, to avoid inconsistencies, do not write variable g_dnToolPositionSet_p!	
g_dnToolPositionSet_p	DINT	Target position in [incr.]	
		- With reference to the machine zero point. - If you use this variable, to avoid inconsistencies, do not write variable g_dnToolPositionSet!	

Cross Cutter Appendix

Outputs			
Global variables		(Variable type: VAR_GLOBAL)	
Name	Data type	Value/Meaning	
g_bToolPositioningBusy	BOOL	"Positioning in progress" status signal	
		TRUE	Positioning in progress.
g_bToolPositioningDone	BOOL	"Target position reached" status signal	
		TRUE	Target position g_dnToolPositionSet(_p) reached.
g_wPosProfilerState	ENUM "StatePositioning"	Status of positioning	
		0 NoAction	No action
		1 PositioningDone	Target position g_dnToolPositionSet(_p) reached.
		13 PositioningBusy	Positioning in progress.
		14 PositioningToCamBusy	Positioning to cam position in progress.
		102 AbortIsActive	Aborting positioning.

6.1.15 VarStateMachine

Inputs			
Global variables (Variable type: VAR_GLOBAL)			
Name	Data type	Value/Meaning	
g_bLockToCam	BOOL	Activate locking of drive to motion profile.	
		TRUE	Locks the drive to the corresponding position on the motion profile.
		TRUE=>False	Unlocks the drive from the motion profile and sets it to standstill.
g_nLockToCamFact	INT	Factor for the ratio Y/X (maximum gradient) for the relevant profile. 16384 = 100% - Default setting: 250 - Only relevant for mode 3	
g_dnLockToCamLimitX_p	DINT	Maximum number of increments used per cycle to lock the drive. - Default setting: 300 [incr.] - Example: g_dnLockToCamLimitX = 1500 Up to 1500 increments are used per cycle to move the drive to the stationary tool position.	
g_dnLockToCamWindow_p	DINT	Following the correction operation performed in mode 3, the tool and profile position are located in this window. Default setting: 30 [incr.]	
g_byLockToCamMode	ENUM "PositionMode"	Selection of the mode in which the drive will be locked to the feed profile.	
		0	The drive is locked to the motion profile along the shortest path. Only this mode is permissible in the case of a forward/reverse profile.
		1	Lock the drive to the motion profile using forward motion only. If an attempt is made to use a forward/reverse profile, an error message will be output and the system will switch to the "Trouble" state via a quick stop (QSP).
		2	Lock the drive to the motion profile using reverse motion only. If an attempt is made to use a forward/reverse profile, an error message will be output and the system will switch to the "Trouble" state via a quick stop (QSP).
g_byLockToCamModeCode	BYTE	Helpvariable for selection of the mode in which the drive will be locked to the feed profile via Codenumber C3630.	
		0	The drive is locked to the motion profile along the shortest path. Only this mode is permissible in the case of a forward/reverse profile.
		1	Lock the drive to the motion profile using forward motion only. If an attempt is made to use a forward/reverse profile, an error message will be output and the system will switch to the "Trouble" state via a quick stop (QSP).
		2	Lock the drive to the motion profile using reverse motion only. If an attempt is made to use a forward/reverse profile, an error message will be output and the system will switch to the "Trouble" state via a

Cross Cutter Appendix

Inputs			
Global variables (Variable type: VAR_GLOBAL)			
			quick stop (QSP).

Outputs			
Global variables (Variable type: VAR_GLOBAL)			
Name	Data type	Value/Meaning	
g_bLockToCamBusy	BOOL	"Traversing to X position on motion profile active" status signal	
		TRUE	Moves the drive to the corresponding position on the motion profile.
g_bLockToCamDone	BOOL	"X position on motion profile reached" status signal	
		TRUE	The drive has arrived at the corresponding position on the motion profile.
g_byOperationState	BYTE	Display indicating the main states of the template	
		100	"Init" state
		51	"LockToCamBusy" state
		50	"CamOperation" state
		40	"PositioningOperation" state
		30	"HomingOperation" state
		20	"ManualOperation" state
		10	"StandBy" state
			Switching over to other states is performed from this initial state.
		1	"Trouble" state

6.1.16 VarTorquePrecontrol

Inputs			
Global variables (Variable type: VAR_GLOBAL)			
Name	Data type	Value/Meaning	
g_bTorquePrecontrolEnable	BOOL	Enable acceleration precontrol.	
		External torque precontrol remains enabled even when acceleration precontrol is disabled.	
		TRUE	Acceleration precontrol enabled.
g_nFactorInertia_a	INT	Factor for adapting the precontrol value to the moment of inertia of the drive	
		- Value range: -32767 ... 32767 16384 = 100% - The variable is a retain variable.	
g_nTorquePreAdd_a	INT	External torque precontrol	
		Remains activated even when acceleration precontrol is deactivated.	

Outputs			
Global variables (Variable type: VAR_GLOBAL)			
Name	Data type	Value/Meaning	
g_nSetTorquePrecontrol_a	INT	Resulting precontrol for torque controller	
		16384 = 100%	

6.1.17 VarTrouble

Inputs		
Global variables (Variable type: VAR_GLOBAL)		
Name	Data type	Value/Meaning
g_bErrorSet	BOOL	Switch to "Error" state. - This variable can be set using a "Set" instruction. - It is reset to FALSE automatically via the template.
		TRUE Switch to "Trouble" state.
g_bErrorReset	BOOL	Reset internal and user-defined errors. Setting the g_bErrorResetEqTripReset variable to TRUE will trigger a TRIP reset simultaneously on the PLC.
		TRUE Reset error.
g_bUserQuickstop	BOOL	User quick stop (quick stop of the drive system)
		TRUE Initiate user quick stop.
g_bControllerInhibit	BOOL	Controller inhibit (CINH)
		TRUE Set controller inhibit on the PLC.
g_bUserStateBus	BOOL	Output control signal via state bus.
		TRUE Set state bus to LOW level to switch all nodes connected to the state bus to a preselected state (e.g. TRIP, QSP or controller inhibit).

Outputs		
Global variables (Variable type: VAR_GLOBAL)		
Name	Data type	Value/Meaning
g_bGlobalError	BOOL	"Global error" status signal - The template is in the "Trouble" state. - If this is a template or user error, g_bErrorSet will also be TRUE.
		TRUE A PLC, template or user error has occurred.
g_wActFault	WORD	Error channel for communication with the "FailManager". This variable can only be used for the FWM block.
g_wErrorNumber	ENUM "ErrorCode"	Display indicating the error designation See the Object Organizer: Data Types tab > UserDataTypes folder > ErrorCode (ENUM)
g_wErrorType	WORD	Output of error response
		0 TRIP
		1 Message
		2 Warning
		3 Fail QSP
		4 Ready
g_bFollorErrorWarn	BOOL	"Following error warning" status signal
		TRUE The following error warning limit (C3645) has been exceeded.
g_bFollorErrorQSP	BOOL	"Following error" status signal
		TRUE The following error limit (C3643) has been exceeded.
g_bErrorResetBusy	BOOL	"Error reset active" status signal
		TRUE A reset of template-internal and user-defined errors is currently in progress.

6.1.18 VarVertShaftExtPosition

Inputs			
Global variables		(Variable type: VAR_GLOBAL)	
Name	Data type	Value/Meaning	
g_bVertShaftExtEnable	BOOL	Enable external master signal.	
		TRUE	X axis defined externally.
		FALSE	X axis generated by virtual master.
g_bPosToVertShaftExt	BOOL	Positioning to external master angle	
		TRUE	Positioning to external master angle.
		FALSE	Direct switch to external master angle.

6.1.19 VarVertShaftExtVelocity

Inputs			
Global variables		(Variable type: VAR_GLOBAL)	
Name	Data type	Value/Meaning	
g_bTypeOfVertShaftExt	BOOL	Selection of external master signal	
		TRUE	Digital frequency via digital frequency input DFIN.
		FALSE	Master angle via system bus (CAN).
g_byVertShaftExtVelMode	BYTE	Selection of operating mode	
		0	Continuous operation The external digital frequency entry is integrated continuously.
		1	Start/stop operation The external digital frequency entry is only integrated when a corresponding start/stop signal is pending at g_bStartStopVertShaftExtVel.
g_bStartStopVertShaftExtVel	BOOL	Start/stop signal for "Start/stop operation"	
		TRUE	Start integration. Integration will continue at least until the next zero crossing.
		FALSE	No integration/abort previously activated integration on the next zero crossing.
g_nVertShaftExt_v	INT	Entry indicating the external digital frequency/speed in [incr./ms]	
g_bVertShaftExtVelCwCcw	BOOL	Invert external digital frequency entry.	
		TRUE	g_nVertShaftExt_v inverted internally.

Cross Cutter Appendix

Outputs			
Global variables		(Variable type: VAR_GLOBAL)	
Name	Data type	Value/Meaning	
g_bVertShaftExtVelXZero	BOOL	"Zero crossing of X axis" status signal	
		TRUE	Zero crossing/zero position at standstill of X axis
g_nVertShaftExtVelState	INT	Status	
		0	OK
		Warnings:	
		10	Starting position g_dnVertShaftStartPosition_p < 0
		11	Starting position g_dnVertShaftStartPosition_p > XMax
		12	Denominator stretch/compress factor = 0
		13	Resulting position change per cycle outside integer value range, limitation active
		Error: (acknowledgement required)	
		-11	Resulting position change per cycle too large for synchronised stretch/compress
		-20	Touch-probe-sensor position g_dnTpPos_p outside adjustable range

6.1.20 VarVirtualMaster

Inputs			
Global variables		(Variable type: VAR_GLOBAL)	
g_bLoadVertShaftStart	BOOL	Accept starting position.	
		FALSE => TRUE	Accept the starting position at the g_dnVertShaftStartPosition_p input for the g_dnVertShaftPosSet_p position output.
		TRUE	Write the starting position to the g_dnVertShaftPosSet_p position output.
g_nVertShaftVelocitySet	INT	Speed setpoint of the X axis in [rpm]	
g_nVertShaftVelocitySet_v	INT	Speed setpoint of the X axis in [incr./ms]	
g_dnVertShaftStartPosition	DINT	Starting position of the X axis in [m_unit] • Internal limitation to 0 ... g_dnVertShaftLengthAct_p • The value is entered in fixed point format with 4 decimal positions (1 [m_unit] = 10000). The value is converted into increments internally. • If you use this variable, to avoid inconsistencies, do not write variable g_dnVertShaftStartPosition_p!	
g_dnVertShaftStartPosition_p	DINT	Starting position of the X axis in [incr.] • Internal limitation to 0 ... g_dnVertShaftLengthAct_p • If you use this variable, to avoid inconsistencies, do not write variable g_dnVertShaftStartPosition!	

Outputs		
Global variables		(Variable type: VAR_GLOBAL)
Name	Data type	Value/Meaning
g_dnVertShaftPosSet_p	DINT	Position output of electrical shaft • Value is also transferred to VirtualClutchXAxis clutch function.
g_nVertShaftMasterOut_v	INT	Speed output of electrical shaft • Speed in [incr./ms] • Can be used for the cam group setpoint or assigned to a downstream network, for example.

6.1.21 VarXTouchProbeSynchronisation

Inputs			
Global variables		(Variable type: VAR_GLOBAL)	
g_bXTPEnable	BOOL	Enable print-mark synchronisation.	
		TRUE	Print-mark synchronisation enabled.
		FALSE	Print-mark synchronisation disabled.
g_bRfgXTPExecute	BOOL	Start ramp generator to correct position deviations.	
		FALSE => TRUE	The position deviation calculated (g_dnXDifference_p) is corrected via the internal ramp generator.
		FALSE	The ramp generator is not active and the position deviation calculated is not corrected. The starting position is retained.
g_bRfgXTPEnable	BOOL	Selection of correction mode for print-mark synchronisation.	
		TRUE	Correct measured distance via internal ramp generator.
		FALSE	Correct measured distance via constant speed value at g_nXTPMakeUpLeeway_v.
g_bXTPReceived	BOOL	Trigger input for calculating position deviation	
		FALSE => TRUE	Recalculate position deviation.
g_nXTPMakeUpLeeway_v	INT	Speed in [incr./ms] for correcting the calculated position deviation. Only relevant for correction via constant speed value (g_bRfgXTPEnable = FALSE). Internal limitation to 0 ... g_dnVertShaftLengthAct_p	
g_dnXTPPos	DINT	Distance from zero point to touch-probe sensor in [m_unit] - Internal limitation to 0 ... 2 * g_dnVertShaftLengthAct_p - The value is entered in fixed point format with 4 decimal positions (1 [unit]/10000). The value is converted into increments internally. - The variable is a retain variable. If you use this variable, to avoid inconsistencies, do not write variable g_dnXTPPos_p!	
g_dnXTPPos_p	DINT	Distance from zero point to TP sensor in [incr.] If you use this variable, to avoid inconsistencies, do not write variable g_dnXTPPos!	
g_dnXTPActLastScan_p	DINT	Δ incr between touch probe and start of task	

Outputs			
Global variables		(Variable type: VAR_GLOBAL)	
Name	Data type	Value/Meaning	
g_bXTPSyncBusy	BOOL	Status of print-mark synchronisation.	
		TRUE	Synchronisation active, correcting X difference.
		FALSE	No correction of setpoint currently active.
g_dnXDifference_p	DINT	Correction value for touch-probe signal Only if g_bRfgXTPEnable = TRUE.	
g_dnXTPDifference_p	DINT	Difference between two touch-probe signals Only if g_bRfgXTPEnable = TRUE.	

6.1.22 VarYStretchCompress

Inputs		
Global variables		(Variable type: VAR_GLOBAL)
g_nYStretchCompress_a	INT	Stretch/compress factor for motion profile. 16384 = 100%

6.1.23 VarYTouchProbeSynchronisation

Inputs		
Global variables		(Variable type: VAR_GLOBAL)
g_bYTpReceived	BOOL	Trigger input for calculating position deviation on tool axis
		FALSE => TRUE Recalculate position deviation.
g_dnYTpActLastScan_p	DINT	Δ incr between touch probe and start of task

6.2 Prepared solution codes

6.2.1 Table of application codes

In addition to the system codes (see 9300ET ServoPLC documentation) in the prepared solution, the settings that are important for the function of the machine can be made using application codes starting at C3000/000.

Code	Possible settings:		Comment
	Default	Selection	
C3010/ 000	0	0 {1}	9 Homing mode
			0 > _Rn_MP
			1 < _Rn_MP
			4 > _Rp < _Rn_MP
			5 < _Rp > _Rn_MP
			6 > _Rn > _TP
			7 < _Rn < _TP
			8 > _TP
			9 < _TP
			Symbols for code C3010: > Movement in positive direction < Movement in negative direction Rp Positive edge of homing switch Rn Negative edge of homing switch TP Touch-probe edge of input I4 for 9300 Servo PLC and I2 for ECS MP Zero pulse edge of motor (one per motor revolution) The execution of the last action in each case will set the home position (e.g. the zero pulse in the case of "MP") even if the drive continues to move subsequently.
C3303/ 000	-	-	- Display code: Master gearbox numerator
C3304/ 000	-	-	- Display code: Master gearbox denominator
C3305/ 000	-	-	- Display code: Master feed constant
C3306/ 000	-	-	- Display code: Slave gearbox numerator
C3307/ 000	-	-	- Display code: Slave gearbox denominator
C3308/ 000	-	-	- Display code: Slave feed constant
C3350/ 000	-	- {1[m_units]}	- Display code: Master angle setpoint
C3351/ 000	-	- {1[m_units]}	- Display code: Current master angle
C3355/ 000	-	- {1[s_units]}	- Display code: Current position
C3356/ 000	-	- {1[s_units]}	- Display code: Profile set position
C3357/ 000	-	- {1[s_units]}	- Display code: Auxiliary integrator
C3359/ 000	-	- {1[s_units]}	- Display code: Following error
C3360/ 000	-	- {1[incr.]}	- Display code: Master angle setpoint

Cross Cutter Appendix

Code	Possible settings:				Comment
	Default	Selection			
C3361/ 000	-	-	{1[incr.]}	-	Display code: Current master angle
C3365/ 000	-	-	{1[incr.]}	-	Display code: Main integrator, current cam position
C3366/ 000	-	-	{1[incr.]}	-	Display code: Auxiliary integrator, current tool position
C3367/ 000	-	-	{1[incr.]}	-	Display code: Profile set position
C3369/ 000	-	-	{1[incr.]}	-	Display code: Following error
C3501/ 000	0	0	{1[m_units]}	1	Selection of application units
				0	[mm]
				1	[inch]
C3505/ 000	-	-	-	-	Display code: Status of state machine
				100	"Init" state
				51	"LockToCamBusy" state
				50	"CamOperation" state
				40	"PositioningOperation" state
				30	"HomingOperation" state
				20	"ManualOperation" state
				10	"StandBy" state Switching over to other states is performed from this initial state.
				1	"Trouble" state
C3620/ 001	5.00	0.01...	{0.01[s]}	...129.99	Deceleration ramp for immediate stop Assuming 4000 rpm
C3620/ 002	5.00	0.01...	{0.01[s]}	...129.99	Acceleration ramp for positioning Assuming 4000 rpm
C3620/ 003	5.00	0.01...	{0.01[s]}	...129.99	Deceleration ramp for positioning Assuming 4000 rpm
C3621/ 001	200	0 ...	{1 rpm}	...15000	Speed for positioning in positive direction (CW)
C3621/ 002	200	0 ...	{1 rpm}	...15000	Speed for positioning in negative direction (CCW)
C3622/ 001	5.000	0.000...	{0.0001[s]}	...129.99	Deceleration ramp for immediate stop for manual jog Assuming 4000 rpm
C3622/ 002	5.000	0.000...	{0.0001[s]}	...129.99	Acceleration and deceleration ramp for normal operation; Assuming 4000 rpm
C3622/ 003	5.000	0.000...	{0.0001[s]}	...129.99	Acceleration and deceleration ramp for manual jog Assuming 4000 rpm
C3623/ 001	200	0 ...	{1 rpm}	...15000	Speed for manual jog in positive direction (CW)
C3623/ 002	200	0 ...	{1 rpm}	...15000	Speed for manual jog in negative direction (CCW)
C3624 /001	5.00	0.01...	{0.01[s]}	...129.99	Deceleration ramp for immediate stop (function abort) for homing Assuming 4000 rpm
C3624 /002	5.00	0.01...	{0.01[s]}	...129.99	Acceleration ramp for homing Assuming 4000 rpm
C3624 /003	5.00	0.01...	{0.01[s]}	...129.99	Deceleration ramp for homing Assuming 4000 rpm
C3625 /001	200	0 ...	{1 rpm}	...15000	Speed for homing in positive direction (CW)
C3625 /002	200	0 ...	{1 rpm}	...15000	Speed for homing in negative direction (CCW)
C3626 /000	0	0	{0.0001 [s_units]}	214000.0000	Entry indicating the position of the homing switch
C3627 /000	0	0	{0.0001 [s_units]}	214000.0000	Target position of tool following homing
C3628/ 000	0.0000	0	{0.0001 [s_units]}	360.0000	Target position of positioning

Cross Cutter Appendix

Code	Possible settings:		Comment
	Default	Selection	
C3629/ 000	0	0 {1}	4 Positioning mode
			0 Move to target position via shortest route.
			1 Move to target position only in positive direction (CW).
			2 Move to target position only in negative direction (CCW).
			4 Positioning relative to entry in C3628/000 (g_dnToolPositionSet).
C3630/ 000	0	0 {1}	2 Positioning mode Lock to cam
			0 Move to cam position via shortest route.
			1 Move to cam position only in positive direction (CW).
			2 Move to cam position only in negative direction (CCW).
C3635/ 000	0	0 {1}	1 Inversion of DFIN signal
			TRUE Invert DFIN signal.
			FALSE Do not invert DFIN signal.
C3636/ 000	0	0 {1}	1 Motor mounting direction
			TRUE Motor mounting direction inverted.
			FALSE Motor mounting direction not inverted.
C3637/ 000	1	-32767 {1} ... 32767	Gearbox (numerator) position feedback It is not allowed to set the value Zero!
C3638/ 000	1	1... {1} ... 32767	Gearbox (denominator) position feedback
C3639/ 000	1	-200... {1 %} ... 200	external torque precontrol
C3640/ 000	1	0 {1}	5 Function of external torque precontrol
			0 TorqueOut = TorqueIn1
			1 TorqueOut = TorqueIn1 + TorqueIn2
			2 TorqueOut = TorqueIn1 – TorqueIn2
			3 $\text{TorqueOut} = \frac{\text{TorqueIn1} \cdot \text{TorqueIn2}}{16384}$
			4 $\text{TorqueOut} = \frac{\text{TorqueIn1}}{[\text{TorqueIn2}]^{164}}$
			5 $\text{TorqueOut} = \frac{\text{TorqueIn1}}{16384 - \text{TorqueIn2}}^{16384}$
C3643	3.0000	0.0001... {0.0001 [s_units]} ... 214748.0000	Selection of following error switch-off threshold
C3644	0.5000	0.0001... {0.0001 [s_units]} ... 214748.0000	Selection of following error hysteresis
C3645	1.5000	0.0001... {0.0001 [s_units]} ... 214748.0000	Selection of following error warning limit
C3646	0.5000	0.0001... {0.0001 [s_units]} ... 214748.0000	Selection of following error warning limit hysteresis
C3649/ 001	0.100	0.000... {0.01[s]} ... 129.99	Abort ramp for ramp generator: Assumes a speed of 4000 rpm. Used if the knife cuts into the material whilst mark correction is in progress.
C3649/ 002	5.000	0.000... {0.01[s]} ... 129.99	Acceleration ramp for ramp generator: Assumes a speed of 4000 rpm.
C3649/ 003	5.000	0.000... {0.01[s]} ... 129.99	Deceleration ramp for ramp generator: Assumes a speed of 4000 rpm.
C3650/ 001	200	0 {1 rpm} 15000	Maximum positive correction speed for mark synchronisation with a ramp generator and correction speed for mark synchronisation without a ramp generator.
C3650/ 002	200	{1 rpm} 15000	Maximum negative correction speed for mark synchronisation with a ramp generator.
C3653/ 000	628.3185	0.0000... {0.0001 m_units} ... 214748.0000	Cutting circle circumference of knife drum

Cross Cutter Appendix

Code	Possible settings:		Comment	
	Default	Selection		
C3654/ 000	30.0000	0.0000... {0.0001 s_units} ... 360.0000	Cutting angle (synchronous angle)	
C3655/ 000	300.0000	0.0001... {0.0001 m_units} ... 214000.0000	Setpoint cutting length (valid for synchronous and asynchronous cross-cutter operation).	
C3656/ 000	150.0000	0.0001... {0.0001 m_units} ... 214000.0000	Setpoint for reject length	
C3657/ 000	2400.0000	0.0001... {0.0001 m_units} ... 214000.0000	Setpoint for test length	
C3658/ 000	-	- {1}	- Display code: Cutting counter for cutting operation The counter is limited to a maximum value of 65535; on reaching this value it will overflow to 0.	
C3659/ 000	-	- {1}	- Display code: Cutting counter for reject operation The counter is limited to a maximum value of 65535; on reaching this value it will overflow to 0.	
C3660/ 000	-	- {1}	- Display code: Cutting counter for test operation The counter is limited to a maximum value of 65535; on reaching this value it will overflow to 0.	
C3661/ 000	-	-	- {1}	- Display code: Status of calculation sequence for new cutting length
			0	Calculation ok.
			1	- No LC9 file found. - Profile data not found in absolute data model. - Read error when accessing profile data.
			2	Selected motion profile not found.
			4	The angle of motion of the knife drum for the asynchronous motion phase is less than or equal to zero (check y final value of basic profile and cutting angle!).
			8	A value of zero has been entered for the knife drum's cutting circle circumference.
			6	The preset cutting length is less than the material length corresponding to the cutting angle.
			32	The selected motion profile contains fewer than 20 interpolation points.
			64	The selected motion profile does not start at zero (0.0000/0.0000).
			128	Cutting angle too small.
			256	The y final value of the basic profile does not correspond to value $360.0000[^\circ]/n_{\text{cuts/rev}}$. ($n_{\text{cuts/rev}}$ = Number of knives on knife drum, values of 1, 2, 3, 4, 5 or 6 are possible)
			512	Error during internal read access to C0011/000
			If the status variable is exhibiting non-zero values, the calculated variables may exhibit invalid values, in which case it will not be possible to use them subsequently in the control program!	

Cross Cutter Appendix

Code	Possible settings:		Comment
	Default	Selection	
C3662/ 000	-	{1}	- Display code: Status of calculation sequence for new reject length
			0 Calculation ok.
			1 - No LC9 file found. - Profile data not found in absolute data model. - Read error when accessing profile data.
			2 Selected motion profile not found.
			4 The angle of motion of the knife drum for the asynchronous motion phase is less than or equal to zero (check y final value of basic profile and cutting angle!).
			8 A value of zero has been entered for the knife drum's cutting circle circumference.
			16 The preset cutting length is less than the material length corresponding to the cutting angle.
			32 The selected motion profile contains fewer than 20 interpolation points.
			64 The selected motion profile does not start at zero (0.0000/0.0000).
			128 Cutting angle too small.
			256 The y final value of the basic profile does not correspond to value $360.0000[^\circ]/n_{\text{cuts/rev}}$. ($n_{\text{cuts/rev}}$ = Number of knives on knife drum, values of 1, 2, 3, 4, 5 or 6 are possible)
			512 Error during internal read access to C0011/000
			If the status variable is exhibiting non-zero values, the calculated variables may exhibit invalid values, in which case it will not be possible to use them subsequently in the control program!
C3663/ 000	-	{1}	- Display code: Status of calculation sequence for new test length
			0 Calculation ok.
			1 - No LC9 file found. - Profile data not found in absolute data model. - Read error when accessing profile data.
			2 Selected motion profile not found.
			4 The angle of motion of the knife drum for the asynchronous motion phase is less than or equal to zero (check y final value of basic profile and cutting angle!).
			8 A value of zero has been entered for the knife drum's cutting circle circumference.
			16 The preset cutting length is less than the material length corresponding to the cutting angle.
			32 The selected motion profile contains fewer than 20 interpolation points.
			64 The selected motion profile does not start at zero (0.0000/0.0000).
			128 Cutting angle too small.
			256 The y final value of the basic profile does not correspond to value $360.0000[^\circ]/n_{\text{cuts/rev}}$. ($n_{\text{cuts/rev}}$ = Number of knives on knife drum, values of 1, 2, 3, 4, 5 or 6 are possible)
			512 Error during internal read access to C0011/000
			If the status variable is exhibiting non-zero values, the calculated variables may exhibit invalid values, in which case it will not be possible to use them subsequently in the control program!

Cross Cutter Appendix

Code	Possible settings:		Comment
	Default	Selection	
C3664/ 000	-	{1}	- Display code: Status of calculation sequence for new start/stop length
			0 Calculation ok.
			1 - No LC9 file found. - Profile data not found in absolute data model. - Read error when accessing profile data.
			2 Selected motion profile not found.
			4 The angle of motion of the knife drum for the asynchronous motion phase is less than or equal to zero (check y final value of basic profile and cutting angle!).
			8 A value of zero has been entered for the knife drum's cutting circle circumference.
			16 The preset cutting length is less than the material length corresponding to the cutting angle.
			32 The selected motion profile contains fewer than 20 interpolation points.
			64 The selected motion profile does not start at zero (0.0000/0.0000).
			128 Cutting angle too small.
			256 The y final value of the basic profile does not correspond to value $360.0000[^\circ]/n_{\text{cuts/rev}}$. ($n_{\text{cuts/rev}}$ = Number of knives on knife drum, values of 1, 2, 3, 4, 5 or 6 are possible)
			512 Error during internal read access to C0011/000
			If the status variable is exhibiting non-zero values, the calculated variables may exhibit invalid values, in which case it will not be possible to use them subsequently in the control program!
C3665/ 000	100	80... {1%} ... 120	Cutting speed adaptation factor
C3666/ 000	0	0 {1}	1 Enable torque precontrol.
			TRUE Torque precontrol is enabled.
			FALSE Torque precontrol is disabled.
C3667/ 000	10.0	-200.0... {1%} ... 200.0	Adaptation factor for torque precontrol
C3668/ 000	370.0000	0.0000 {1.0000 m_units} 214000.0000	Distance measured from mark sensor to 6 o'clock position of knife.
C3669/ 000	60.0000	0.0000... {0.0001 m_units} ... 214000.0000	Size of mark window
C3670/ 000	1	0 ... {1} ... 65535	Number of cycles permitted without a mark signal appearing in the mark window. A warning signal is generated if this number is exceeded.
C3671/ 000	-	- - -	Display code: X trimming outputvalue scaled in 1.0000 m_units
C3672/ 002	20.0000	0.0100... {0.01 s} ... 130.0000	Acceleration ramp time for position trimming
C3672/ 003	20.0000	0.0100... {0.01 s} ... 130.0000	Deceleration ramp time for position trimming
C3673/ 001	200.0000	0 ... {1 rpm} ... 15000	Speed for position trimming in a positive direction
C3673/ 002	200.0000	0 ... {1 rpm} ... 15000	Speed for position trimming in a negative direction
C3674/ 000	0.0000	-214748.0000... {0.0001 m_units} ... 214748.0000	Selection of the absolute trimming target position
C3676/ 000	10.000	0.000... {0.001 s} ... 999.999	Acceleration ramp time for inching trimming
C3677/ 000	10.000	0.000... {0.001 s} ... 999.999	Deceleration ramp time for inching trimming
C3678/ 000	6	0 ... {1 rpm} ... 15000	Selection of the inching speed
C3679/ 000	0.0000	0.0000... {0.0001 m_units} ... 214000.0000	Distance from reject gate to cutting position
C3680/ 000	0	0 ... {1 ms} ... 65535	Switching dwell time (switching delay time) of reject gate

Cross Cutter Appendix

Code	Possible settings:		Comment
	Default	Selection	
C3681/000	0	-15000... {1 rpm} ...15000	Selection of the test signal for the simulation of the DFIN input
C3683/000	15000	-15000... {1 rpm} ...15000	Selection of speed limitation, upper speed limit
C3684/000	-15000	0 ... {1 rpm} ...15000	Selection of speed limitation, lower speed limit
C3685/000	-	- {1 [m_units/min]}	- Display code: Line speedt
C3686/000	-	- {1 [m_units/min]}	- Display code: Maximum possible line speed for set synchronus cutting length.
C3687/000	-	- {1 [m_units/min]}	- Display code: Maximum possible line speed for set reject length.
C3688/000	-	- {1 [m_units/min]}	- Display code: Maximum possible line speed for set test length.
C3689/000	-	- {1 [m_units/min]}	- Display code: Maximum possible line speed for set start/stop length.
C3690/000	-	- {1}	- Display code: Number of knives mounted on the knife drum. Up to 6 knives can be mounted on the knife drum.
C3691/000	1	0 {1}	2 Selection of calculation method for asynchronous section of motion profile
			0 Second order polynomial (linear speed ramps)
			1 Fifth order polynomial
			2 Sloping sine line
C3692/000	-	- {1}	- Display code: Status of calculation sequence for new asyn. cutting length
			0 Calculation ok.
			1 - No LC9 file found. - Profile data not found in absolute data model. - Read error when accessing profile data.
			2 Selected motion profile not found.
			4 The angle of motion of the knife drum for the asynchronous motion phase is less than or equal to zero (check y final value of basic profile and cutting angle!).
			8 A value of zero has been entered for the knife drum's cutting circle circumference.
			16 The preset cutting length is less than the material length corresponding to the cutting angle.
			32 The selected motion profile contains fewer than 20 interpolation points.
			64 The selected motion profile does not start at zero (0.0000/0.0000).
			128 Cutting angle too small.
			256 The y final value of the basic profile does not correspond to value $360.0000[^\circ]/n_{\text{cuts/rev}}$. ($n_{\text{cuts/rev}}$ = Number of knives on knife drum, values of 1, 2, 3, 4, 5 or 6 are possible)
			512 Error during internal read access to C0011/000
If the status variable is exhibiting non-zero values, the calculated variables may exhibit invalid values, in which case it will not be possible to use them subsequently in the control program!			
C3693/000	-	- {1 [m_units/min]}	- Display code: Maximum possible line speed for set asynchronous cutting length.
C3694/000	-	- {1[s_units]}	- Display code: Starting position of the cross cutter.
C3700/000	0	0 {1}	1 Display code: Cutting length induces start/stop operation of cross cutter.
			TRUE The selected cutting length exceeds the limit cutting length for start/stop operation => The pause will be extended accordingly in the asynchronous motion phase.
			FALSE The selected cutting length undershoots the limit cutting length for start/stop operation.

Cross Cutter Appendix

Code	Possible settings:		Comment
	Default	Selection	
C3701/ 000	0	0 {1}	1 Display code: Reject length induces start/stop operation of cross cutter.
			TRUE The selected reject length exceeds the limit cutting length for start/stop operation => The pause will be extended accordingly in the asynchronous motion phase.
			FALSE The selected reject length undershoots the limit cutting length for start/stop operation.
C3702/ 000	0	0 {1}	1 Display code: Test length induces start/stop operation of cross cutter.
			TRUE The selected test length exceeds the limit cutting length for start/stop operation => The pause will be extended accordingly in the asynchronous motion phase.
			FALSE The selected test length undershoots the limit cutting length for start/stop operation.
C3703/ 000	0	0 {1}	1 Display code: Start/stop length induces start/stop operation of cross cutter.
			TRUE The selected start/stop length exceeds the limit cutting length for start/stop operation => The pause will be extended accordingly in the asynchronous motion phase.
			FALSE The selected start/stop length undershoots the limit cutting length for start/stop operation.
C3704/ 000	-	- {0.0001 [m_units]}	- Display code: Starting point of pause in motion profile for cutting length. Only if C3700/000 = TRUE.
C3705/ 000	-	- {0.0001 [m_units]}	- Display code: Starting point of pause in motion profile for reject length. Only if C3701/000 = TRUE.
C3706/ 000	-	- {0.0001 [m_units]}	- Display code: Starting point of pause in motion profile for test length. Only if C3702/000 = TRUE.
C3707/ 000	-	- {0.0001 [m_units]}	- Display code: Starting point of pause in motion profile for start/stop length. Only if C3703/000 = TRUE.
C3708/ 000	-	- {0.0001 [m_units]}	- Display code: Finishing point of pause in motion profile for cutting length. Only if C3700/000 = TRUE.
C3709/ 000	-	- {0.0001 [m_units]}	- Display code: Finishing point of pause in motion profile for reject length. Only if C3701/000 = TRUE.
C3710/ 000	-	- {0.0001 [m_units]}	- Display code: Finishing point of pause in motion profile for test length. Only if C3702/000 = TRUE.
C3711/ 000	-	- {0.0001 [m_units]}	- Display code: Finishing point of pause in motion profile for start/stop length. Only if C3703/000 = TRUE.
C3712/ 000	-	- {0.0001 [m_units]}	- Display code: Material length which the cross cutter works to as a cutting length in the motion profile in the asynchronous motion phase. The value is the difference between the cutting length and the synchronous X length. Note: The value of this code is only recalculated during a calculation sequence.
C3713/ 000	-	- {0.0001 [m_units]}	- Display code: Material length which the cross cutter works to as a reject length in the motion profile in the asynchronous motion phase. The value is the difference between the reject length and the synchronous X length. Note: The value of this code is only recalculated during a calculation sequence.
C3714/ 000	-	- {0.0001 [m_units]}	- Display code: Material length which the cross cutter works to as a test length in the motion profile in the asynchronous motion phase. The value is the difference between the test length and the synchronous X length. Note: The value of this code is only recalculated during a calculation sequence.

Cross Cutter Appendix

Code	Possible settings:		Comment
	Default	Selection	
C3715/ 000	-	- {0.0001 [m_units]}	- Display code: Material length which the cross cutter works to as a start/stop length in the motion profile in the asynchronous motion phase. The value is the difference between the start/stop length and the synchronous X length. Note: The value of this code is only recalculated during a calculation sequence.
C3716/ 000	-	- {0.0001 [s_units]}	- Display code: Y position in the motion profile for the cutting length at which the cross-cutter knife drum is located in the starting position.
C3717/ 000	-	- {0.0001 [s_units]}	- Display code: Y position in the motion profile for the reject length at which the cross-cutter knife drum is located in the starting position.
C3718/ 000	-	- {0.0001 [s_units]}	- Display code: Y position in the motion profile for the test length at which the cross-cutter knife drum is located in the starting position.
C3719/ 000	-	- {0.0001 [s_units]}	- Display code: Y position in the motion profile for the start/stop length at which the cross-cutter knife drum is located in the starting position.
C3997/ 000	-	- {0.01}	- Display code: Template version
C3998/ 000	-	- {1}	- Display code: Display of current error (see Chapter 6.3)
C3999 1 2 3	-	- {1}	- Display code: Version release of the prepared solution (subcode 1: project file, subcode 2: application library, subcode 3: basic library): ▪ The first and second numeral read from right-to-left define the service pack. ▪ The third and forth numeral read from right-to-left define the sub version. ▪ The fifth and sixth numeral read from right-to-left define the main version.
C4000/ 000	0	0 {1}	6 Selection of control interface
			0 Control via code
			1 Control via CAN 1
			2 Control via CAN 2
			3 Control via CAN 3
			4 Control via AIF 1
			5 Control via AIF 2
			6 Control via AIF 3

Cross Cutter Appendix

Code	Possible settings:		Comment
	Default	Selection	
C4010/		0 {1} 31	Configuration codes for the control interface: Each subcode defines, for a specific control functionality, the source of the control information (which bit is assigned):
001	0		The single subcodes are assigned with the following specific control functionality: 001: Homing start 002: Home position set 003: Home position reset. 004: Manual jog in positiv 005: Manual jog in negativ 006: Positioning start 007: User trip set 008: Trip reset 009: Controller inhibit. 010: Quickstop 011: Switch to internal line speed (C3681) 012: Reset cutting length counter 013: Reset reject length counter 014: Reset test length counter. 015: Automatic mode (Cross cutter operation) 016: Accept cutting length 017: Accept reject length 018: Accept test length 019: Activate x-trimming 020: Positiv x-trimming 021: Negativ x-trimming 022: Enable print-mark controller 023: Ramp based adjustment 024: Reset missingmark counter 025: Abort cross cutter operation 026: Activate x-trimming positioning 027: (not assigned) 028: (not assigned) 029: (not assigned) 030: (not assigned) 031: (not assigned) 032: (not assigned) The cross-cutter axis can be controlled via the control interface. The default values are set via code initialisation.
002	1		
003	2		
004	3		
005	4		
006	5		
007	6		
008	7		
009	8		
010	9		
011	10		
012	11		
013	12		
014	13		
015	14		
016	15		
017	16		
018	17		
019	18		
020	19		
021	20		
022	21		
023	22		
024	23		
025	24		
026	25		
027	26		
028	27		
029	28		
030	29		
031	30		
032	31		

Cross Cutter Appendix

Code	Possible settings:		Comment
	Default	Selection	
C4012/		0 {1} 31	Configuration codes for the status interface: Each subcode defines, for a specific status bit, the source of the status information: The single subcodes are assigned with the following Bits: 001: Bit 0 002: Bit 1 032: Bit 31 The following status information, can be assigned free of choice to the 31 Bits: 0: Trip state 1: Follow error warning 2: Follow error trip 3: Positioning active 4: Positioning done 5: Lock to cam busy 6: Cross Cutter Operation 7: Print mark correction active 8: Recalculation of cutting profile 9: Recalculation of reject profile 10: Recalculation of test profile 11: Recalculation of start/stop profile 12: Calculation of cutting profile OK 13: Calculation of reject profile OK 14: Calculation of test profile OK 15: Calculation of start/stop profile OK 16: Homing busy 17: Home position available 18: Homing error 19: NC 20: asynchronus cutting operation 21: synchronus cutting operation 22: Cutting mode 23: Reject mode 24: Test mode 25: Open reject gate 26: Maximum missed marks exceeded 27: Print mark controller enabled 28: Print mark register active 29: Knife in material (sync angle) 30: NC 31: Profile calculations done The cross-cutter axis can be monitored via the status interface. The default values are set via code initialisation.
001	0		
002	1		
003	2		
004	3		
005	4		
006	5		
007	6		
008	7		
009	8		
010	9		
011	10		
012	11		
013	12		
014	13		
015	14		
016	15		
017	16		
018	17		
019	18		
020	19		
021	20		
022	21		
023	22		
024	23		
025	24		
026	25		
027	26		
028	27		
029	28		
030	29		
031	30		
032	31		

Cross Cutter Appendix

Code	Possible settings:			Comment
	Default	Selection		
C4135/ 000 Bit 0	0	0 {1}	1	Application control word bit 0: Start/stop homing, only in conjunction with controller enable.
			FALSE => TRUE	Start homing.
			FALSE	Stop/abort homing.
C4135/ 000 Bit 1	0	0 {1}	1	Application control word bit 1: Set home position.
			FALSE => TRUE	The setpoint defined in C3626 (position of homing switch) is applied and g_bHomingDone is set to TRUE.
			FALSE	Home position not set.
C4135/ 000 Bit 2	0	0 {1}	1	Application control word bit 2: Reset home position.
			TRUE	The home position is reset. The g_bHomingDone output is reset to FALSE. The "Home position detected" status is reset.
			FALSE	Home position not reset.
C4135/ 000 Bit 3	0	0 {1}	1	Application control word bit 3: Manual jog in positive direction (CW)
			TRUE	Manual jog in positive direction
			FALSE	Stop.
C4135/ 000 Bit 4	0	0 {1}	1	Application control word bit 4: Manual jog in negative direction (CCW)
			TRUE	Manual jog in negative direction
			FALSE	Stop.
C4135/ 000 Bit 5	0	0 {1}	1	Application control word bit 5: Execute positioning.
			TRUE	The drive is moved to the target position along the set ramps and at the set speed.
			FALSE	Positioning interrupted.
C4135/ 000 Bit 6	0	0 {1}	1	Application control word bit 6: Activation of the external user error.
			TRUE	External user error pending
			FALSE	No external user error pending
C4135/ 000 Bit 7	0	0 {1}	1	Application control word bit 7: Reset error.
			TRUE	Resetting error as long as its cause has been removed.
			FALSE	Not resetting error.
C4135/ 000 Bit 8	0	0 {1}	1	Application control word bit 8: Inhibit controller.
			TRUE	Controller inhibited (Controller Inhibit).
			FALSE	Controller enabled.
C4135/ 000 Bit 9	0	0 {1}	1	Application control word bit 9: QSP
			TRUE	Quickstop activ
			FALSE	Quickstop not activ
C4135/ 000 Bit 10	0	0 {1}	1	Application control word bit 10: Activates the DFIN test signal to simulate the DFIN signal.
			TRUE	Using the DFIN test signal (<i>g_nTestsignalDFIN_v</i>).
			FALSE	Using the DFIN signal.
C4135/ 000 Bit 11	0	0 {1}	1	Application control word bit 11: Reset cutting counter for cutting operation.
			TRUE	Reset counter.
			FALSE	Counter will not be reset.
C4135/ 000 Bit 12	0	0 {1}	1	Application control word bit 12: Reset cutting counter for reject operation.
			TRUE	Reset counter.
			FALSE	Counter will not be reset.

Cross Cutter Appendix

Code	Possible settings:				Comment
	Default	Selection			
C4135/ 000 Bit 13	0	0 {1}	1		Application control word bit 13: Reset cutting counter for test operation.
					TRUE Reset counter.
					FALSE Counter will not be reset.
C4135/ 000 Bit 14	0	0 {1}	1		Application control word bit 14: Start and stop selected cross-cutter operating mode.
					TRUE Start operation.
					FALSE Stop operation.
C4135/ 000 Bit 15	0	0 {1}	1		Application control word bit 15: Activate the bit to immediately apply a new cutting length you have calculated and work with the new cutting length.
					TRUE Apply new cutting length calculated.
					FALSE Do not apply new cutting length calculated, continue to work with the last cutting length calculated.
C4135/ 000 Bit 16	0	0 {1}	1		Application control word bit 16: Activate the bit to immediately apply a new reject length you have calculated and work with the new cutting length.
					TRUE Apply new reject length calculated.
					FALSE Do not apply new reject length calculated, continue to work with the last cutting length calculated.
C4135/ 000 Bit 17	0	0 {1}	1		Application control word bit 17: Activate the bit to immediately apply a new test length you have calculated and work with the new cutting length.
					TRUE Apply new test length calculated.
					FALSE Do not apply new test length calculated, continue to work with the last cutting length calculated.
C4135/ 000 Bit 18	0	0 {1}	1		Application control word 18: Activates X trimming and switches from the calculated mark sensor distance to the mark sensor distance specified by X trimming.
					TRUE X trimming activated.
					FALSE X trimming deactivated.
C4135/ 000 Bit 19	0	0 {1}	1		Application control word bit 19: Key this input to perform X trimming in a positive direction at the set speed.
					TRUE Perform X trimming in a positive direction.
					FALSE Do not perform X trimming.
C4135/ 000 Bit 20	0	0 {1}	1		Application control word bit 20: Key this input to perform X trimming in a negative direction at the set speed.
					TRUE Perform X trimming in a negative direction.
					FALSE Do not perform X trimming.
C4135/ 000 Bit 21	0	0 {1}	1		Application control word bit 21: Activate print-mark synchronisation.
					TRUE Print-mark synchronisation enabled.
					FALSE Print-mark synchronisation disabled.
C4135/ 000 Bit 22	0	0 {1}	1		Application control word bit 22: Activate ramp generator for print-mark synchronisation.
					TRUE Ramp generator activated.
					FALSE Ramp generator deactivated.
C4135/ 000 Bit 23	0	0 {1}	1		Application control word bit 23: Reset of Counter for not detected printmarks
					TRUE Reset of counter for not detected printmarks
					FALSE No Reset of counter for not detected printmarks

Cross Cutter Appendix

Code	Possible settings:				Comment
	Default	Selection			
C4135/ 000 Bit 24	0	0 {1}	1		Application control word bit 24: Abort cross cutter operation
					TRUE Abort cross cutter operation
					FALSE No abort of cross cutter operation
C4135/ 000 Bit 25	0	0 {1}	1		Application control word bit 25: Enable X-trimming positioning
					TRUE Enable X-trimming positioning
					FALSE Disable X-trimming positioning
C4135/ 000 Bit 26	0	0 {1}	1		Application control word bit 26: Not assigned
					TRUE -
					FALSE -
C4135/ 000 Bit 27	0	0 {1}	1		Application control word bit 27: Not assigned
					TRUE -
					FALSE -
C4135/ 000 Bit 28	0	0 {1}	1		Application control word bit 28: Not assigned
					TRUE -
					FALSE -
C4135/ 000 Bit 29	0	0 {1}	1		Application control word bit 29: Not assigned
					TRUE -
					FALSE -
C4135/ 000 Bit 30	0	0 {1}	1		Application control word bit 30: Not assigned
					TRUE -
					FALSE -
C4135/ 000 Bit 31	0	0 {1}	1		Application control word bit 31: Not assigned
					TRUE -
					FALSE -
C4136/ 000	-	-	-	-	Display code: Current application control word
C4137/ 000	0	1 {1}	4		Control word: Selection of operating mode for the cross cutter Note: You can switch between modes 1 to 3 during operation. To switch to mode 4 from one of modes 1 to 3, you will first need to stop the cross cutter via C4135 bit 14 (<i>g_bStartStopOperation</i>). To switch from mode 4 to one of modes 1 to 3, you will first need to stop the cross cutter via C4135 bit 14 (<i>g_bStartStopOperation</i>).
					1 Synchronous cross cutter, continuous cutting
					2 Synchronous cross cutter, reject operation
					3 Synchronous cross cutter, test operation
					4 Asynchronous cross cutter, continuous cutting
C4138/ 000	0	- {1}	-		Display code: Current value of control word. (Selection of operating mode for cross cutter.)
					1 Synchronous cross cutter, continuous cutting
					2 Synchronous cross cutter, reject operation
					3 Synchronous cross cutter, test operation
					4 Asynchronous cross cutter, continuous cutting

Cross Cutter Appendix

Code	Possible settings:		Comment
	Default	Selection	
C4150/ 000 Bit 0	-	-	Application status word bit 0: System error
			TRUE An error is pending.
			FALSE No errors are pending.
C4150/ 000 Bit 1	-	-	Application status word bit 1: Following error warning
			TRUE A following error warning is pending.
			FALSE No following error warnings are pending.
C4150/ 000 Bit 2	-	-	Application status word bit 2: Following error
			TRUE A following error is pending.
			FALSE No following errors are pending.
C4150/ 000 Bit 3	-	-	Application status word bit 3: Positioning in progress.
			TRUE Positioning in progress.
			FALSE Positioning not in progress.
C4150/ 000 Bit 4	-	-	Application status word bit 4: Positioning target position reached.
			TRUE Target position reached.
			FALSE Target position not reached.
C4150/ 000 Bit 5	-	-	Application status word bit 5: System locking to motion profile.
			TRUE Locking to motion profile in progress.
			FALSE Locking to motion profile not in progress.
C4150/ 000 Bit 6	-	-	Application status word bit 6: System has locked to motion profile.
			TRUE Locking to motion profile completed successfully.
			FALSE System has unlocked from motion profile.
C4150/ 000 Bit 7	-	-	Application status word bit 7: Mark correction active.
			TRUE Mark correction active.
			FALSE Mark correction not active.
C4150/ 000 Bit 8	-	-	Application status word bit 8: Calculation sequence for new cross-cutter motion profile for a new cutting length in progress.
			TRUE Calculation sequence for cross-cutter motion profile for a new cutting length in progress.
			FALSE Calculation sequence for new cross-cutter motion profile for a new cutting length completed successfully or aborted.
C4150/ 000 Bit 9	-	-	Application status word bit 9: Calculation sequence for new cross-cutter motion profile for a new reject length in progress.
			TRUE Calculation sequence for cross-cutter motion profile for a new reject length in progress.
			FALSE Calculation sequence for new cross-cutter motion profile for a new reject length completed successfully or aborted.
C4150/ 000 Bit 10	-	-	Application status word bit 10: Calculation sequence for new cross-cutter motion profile for a new test length in progress.
			TRUE Calculation sequence for cross-cutter motion profile for a new test length in progress.
			FALSE Calculation sequence for new cross-cutter motion profile for a new test length completed successfully or aborted.

Cross Cutter Appendix

Code	Possible settings:		Comment
	Default	Selection	
C4150/ 000 Bit 11	-	-	Application status word bit 11: Calculation sequence for new cross-cutter motion profile for a new start/stop length in progress.
			TRUE Calculation sequence for cross-cutter motion profile for a new start/stop length in progress.
			FALSE Calculation sequence for new cross-cutter motion profile for a new start/stop length completed successfully or aborted.
C4150/ 000 Bit 12	-	-	Application status word bit 12: Calculation sequence for new cross-cutter motion profile for a new cutting length completed successfully. Starting a new calculation sequence will reset this status output to FALSE.
			TRUE Calculation sequence for new cross-cutter motion profile for a new cutting length completed successfully.
			FALSE Calculation sequence for new cross-cutter motion profile for a new cutting length not completed.
C4150/ 000 Bit 13	-	-	Application status word bit 13: Calculation sequence for new cross-cutter motion profile for a new reject length completed successfully. Starting a new calculation sequence will reset this status output to FALSE.
			TRUE Calculation sequence for new cross-cutter motion profile for a new reject length completed successfully.
			FALSE Calculation sequence for new cross-cutter motion profile for a new reject length not completed.
C4150/ 000 Bit 14	-	-	Application status word bit 14: Calculation sequence for new cross-cutter motion profile for a new test length completed successfully. Starting a new calculation sequence will reset this status output to FALSE.
			TRUE Calculation sequence for new cross-cutter motion profile for a new test length completed successfully.
			FALSE Calculation sequence for new cross-cutter motion profile for a new test length not completed.
C4150/ 000 Bit 15	-	-	Application status word bit 15: Calculation sequence for new cross-cutter motion profile for a new start/stop length completed successfully. Starting a new calculation sequence will reset this status output to FALSE.
			TRUE Calculation sequence for new cross-cutter motion profile for a new start/stop length completed successfully.
			FALSE Calculation sequence for new cross-cutter motion profile for a new start/stop length not completed.
C4150/ 000 Bit 16	-	-	Application status word bit 16: Homing activated.
			TRUE Homing active.
			FALSE Homing not active.
C4150/ 000 Bit 17	-	-	Application status word bit 17: Home position detected.
			TRUE Homing complete, home position detected.
			FALSE Home position not detected (an encoder error may have occurred).

Cross Cutter Appendix

Code	Possible settings:		Comment
	Default	Selection	
C4150/ 000 Bit 18	-	-	Application status word bit 18: Homing error
			TRUE An error occurred when setting the home position. See g_nHomingErrorNo for more detailed information about the error.
			FALSE No errors occurred when setting the home position.
C4150/ 000 Bit 19	-	-	Application status word bit 19: Not assigned
			TRUE -
			FALSE -
C4150/ 000 Bit 20	-	-	Application status word bit 20: The cross cutter is operating as an asynchronous cross cutter.
			TRUE The cross cutter is operating as an asynchronous cross cutter.
			FALSE The cross cutter is not operating as an asynchronous cross cutter.
C4150/ 000 Bit 21	-	-	Application status word bit 21: The cross cutter is operating as a synchronous cross cutter.
			TRUE The cross cutter is operating as a synchronous cross cutter.
			FALSE The cross cutter is not operating as a synchronous cross cutter.
C4150/ 000 Bit 22	-	-	Application status word bit 22: The cross cutter is operating in "Continuous cutting" mode.
			TRUE The cross cutter is operating in "Continuous cutting" mode.
			FALSE The cross cutter is not operating in "Continuous cutting" mode.
C4150/ 000 Bit 23	-	-	Application status word bit 23: The cross cutter is operating in "Reject operation" mode.
			TRUE The cross cutter is operating in "Reject operation" mode.
			FALSE The cross cutter is not operating in "Reject operation" mode.
C4150/ 000 Bit 24	-	-	Application status word bit 24: The cross cutter is operating in "Test operation" mode.
			TRUE The cross cutter is operating in "Test operation" mode.
			FALSE The cross cutter is not operating in "Test operation" mode.
C4150/ 000 Bit 25	-	-	Application status word bit 25: Opening reject gate.
			TRUE Open the reject gate.
			FALSE Do not open the reject gate.
C4150/ 000 Bit 26	-	-	Application status word bit 26: The maximum number of cycles without a mark signal appearing in the mark window has been exceeded.
			TRUE The maximum number of cycles without a mark signal appearing in the mark window has been reached.
			FALSE The maximum number of cycles without a mark signal appearing in the mark window has not been reached.
C4150/ 000 Bit 27	-	-	Application status word bit 27: System enabling print-mark control.
			TRUE Print-mark control enabled by the system.
			FALSE Print-mark control not enabled by the system.

Cross Cutter Appendix

Code	Possible settings:		Comment
	Default	Selection	
C4150/ 000 Bit 28	-	-	- Application status word bit 28: Using print-mark register.
			TRUE Use the print-mark register.
			FALSE Do not use the print-mark register.
C4150/ 000 Bit 29	-	-	- Application status word bit 29: Knife is in material.
			TRUE Knife in material.
			FALSE Knife not in material.
C4150/ 000 Bit 30	-	-	- Application status word bit 30: Not assigned
			TRUE
			FALSE
C4150/ 000 Bit 31	-	-	- Application status word 31: All calculation sequences for new cross-cutter motion profiles have been completed successfully. Starting a new calculation sequence will reset this status output to FALSE.
			TRUE All calculation sequences for new cross-cutter motion profiles have been completed successfully.
			FALSE Some calculation sequences for new cross-cutter motion profiles are still in progress.
C4152/ 000	-	-	- Display code: Current application status word

6.2.2 Code initialisation values

The following codes are initialised on a project-specific basis and differ from the default setting initialisation values.

Code	Initialisation	Comment
C0114/001	1,0000	Inversion of digital input I1 (0V = QSP) using 9300 Servo PLC
C0114/004	1,0000	Inversion of digital input I4 (0V = QSP) using ECS
C0352/000	1,0000	Device declared as CAN boot-up master.
C0427/000	0,0000	DFIN function of master signal is "2 phases".
C0428/000	1,0000	DFIN touch-probe source is digital input I5 (terminal X5/I5).
C0469/000	0,0000	Deactivation of stop pushbutton (keypad)
C0540/000	4,0000	DFOUT X9=X10
C2104/000	1,0000	PLC auto-start
C3649/001	0.1000	Time for abort ramp during print-mark synchronisation
C4010/002	1,0000	Control bit 1 Home position set read from input bit 1
C4010/003	2,0000	Control bit 2 Home position reset read from input bit 2
C4010/004	3,0000	Control bit 3 Manual jog in positiv read from input bit 3
C4010/005	4,0000	Control bit 4 Manual jog in negativ read from input bit 4
C4010/006	5,0000	Control bit 5 Positioning start read from input bit 5
C4010/007	6,0000	Control bit 6 User trip set read from input bit 6
C4010/008	7,0000	Control bit 7 Trip reset read from input bit 7
C4010/009	8,0000	Control bit 8 Controller inhibit read from input bit 8
C4010/010	9,0000	Control bit 9 Quickstop read from input bit 9
C4010/011	10,0000	Control bit 10 Switch to internal line speed (C3681) read from input bit 10
C4010/012	11,0000	Control bit 11 Reset cutting length counter read from input bit 11
C4010/013	12,0000	Control bit 12 Reset reject length counter read from input bit 12
C4010/014	13,0000	Control bit 13 Reset test length counter read from input bit 13
C4010/015	14,0000	Control bit 14 Automatic mode (Cross cutter operation) read from input bit 14
C4010/016	15,0000	Control bit 15 Accept cutting length read from input bit 15
C4010/017	16,0000	Control bit 16 Accept reject length read from input bit 16
C4010/018	17,0000	Control bit 17 Accept test length read from input bit 17
C4010/019	18,0000	Control bit 18 Activate x-trimming read from input bit 18
C4010/020	19,0000	Control bit 19 Positiv x-trimming read from input bit 19
C4010/021	20,0000	Control bit 20 Negativ x-trimming read from input bit 20
C4010/022	21,0000	Control bit 21 Enable print-mark controller read from input bit 21
C4010/023	22,0000	Control bit 22 Ramp based adjustment read from input bit 22
C4010/024	23,0000	Control bit 23 Reset missingmark counter read from input bit 23
C4010/025	24,0000	Control bit 24 Abort cross cutter operation read from input bit 24
C4010/026	25,0000	Control bit 25 Activate x-trimming positioning read from input bit 25
C4010/027	26,0000	Control bit 26 not assigned read from input bit 26
C4010/028	27,0000	Control bit 27 not assigned read from input bit 27
C4010/029	28,0000	Control bit 28 not assigned read from input bit 28
C4010/030	29,0000	Control bit 29 not assigned read from input bit 29
C4010/031	30,0000	Control bit 30 not assigned read from input bit 30
C4010/032	31,0000	Control bit 31 not assigned read from input bit 31
C4012/002	1,0000	Status bit 1 output to output bit 1, Follow error warning
C4012/003	2,0000	Status bit 2 output to output bit 2, Follow error trip
C4012/004	3,0000	Status bit 3 output to output bit 3, Positioning active
C4012/005	4,0000	Status bit 4 output to output bit 4, Positioning done
C4012/006	5,0000	Status bit 5 output to output bit 5, Lock to cam busy
C4012/007	6,0000	Status bit 6 output to output bit 6, Cross Cutter Operation
C4012/008	7,0000	Status bit 7 output to output bit 7, Print mark correction active
C4012/009	8,0000	Status bit 8 output to output bit 8, Recalculation of cutting profile
C4012/010	9,0000	Status bit 9 output to output bit 9, Recalculation of reject profile
C4012/011	10,0000	Status bit 10 output to output bit 10, Recalculation of test profile
C4012/012	11,0000	Status bit 11 output to output bit 11, Recalculation of start/stop profile
C4012/013	12,0000	Status bit 12 output to output bit 12, Calculation of cutting profile OK
C4012/014	13,0000	Status bit 13 output to output bit 13, Calculation of reject profile OK
C4012/015	14,0000	Status bit 14 output to output bit 14, Calculation of test profile OK
C4012/016	15,0000	Status bit 15 output to output bit 15, Calculation of start/stop profile OK
C4012/017	16,0000	Status bit 16 output to output bit 16, Homing busy
C4012/018	17,0000	Status bit 17 output to output bit 17, Home position available
C4012/019	18,0000	Status bit 18 output to output bit 18, Homing error
C4012/020	19,0000	Status bit 19 output to output bit 19, NC
C4012/021	20,0000	Status bit 20 output to output bit 20, Asynchronus cutting operation

Cross Cutter Appendix

Code	Initialisation	Comment
C4012/022	21,0000	Status bit 21 output to output bit 21, synchronus cutting operation
C4012/023	22,0000	Status bit 22 output to output bit 22, Cutting mode
C4012/024	23,0000	Status bit 23 output to output bit 23, Reject mode
C4012/025	24,0000	Status bit 24 output to output bit 24, Test mode
C4012/026	25,0000	Status bit 25 output to output bit 25, Open reject gate
C4012/027	26,0000	Status bit 26 output to output bit 26, Maximum missed marks exceeded
C4012/028	27,0000	Status bit 27 output to output bit 27, Print mark controller enabled
C4012/029	28,0000	Status bit 28 output to output bit 28, Print mark register active
C4012/030	29,0000	Status bit 29 output to output bit 29, Knife in material (sync angle)
C4012/031	30,0000	Status bit 30 output to output bit 30, NC
C4012/032	31,0000	Status bit 31 output to output bit 31, Profile calculations done

6.3 Error messages

Where error messages are concerned, the prepared solutions draw a distinction between what are known as system error messages (see Chapter 6.3.1) and process-based error messages (advanced error messages, see Chapter 6.3.2 and Chapter 6.3.3).

6.3.1 System error messages

System error messages are generated in the target system's operating system in response to all impermissible drive-based states (e.g. overcurrent, undervoltage, errors in the feedback system, etc.).

The following system error messages can occur on the ServoPLC:

Error number	Error code	Cause	Remedy
11	OC1	Actual short circuit between motor phases	Disconnect the motor and check for short circuits between all winding connections. Disconnect the supply cable and check for short circuits between phases. Run the inverter off-load: It ought to be possible to reset the error at this point. If it is not, an internal short circuit has occurred => Replace the controller or submit it for repair.
		High capacitive charging current on the motor cable	Use a shorter, low-capacitance motor cable. Install an output choke at the controller's output phases.
12	OC2	Actual earth fault on one of the motor phases	Disconnect the motor and check the winding connections for earth faults. Disconnect the supply cable and check for short circuits between phases and earth. Run the inverter off-load: It ought to be possible to reset the error at this point. If it is not, an internal earth fault has occurred => Replace the controller or submit it for repair.
		High capacitive charging current on the motor cable	Use a shorter, low-capacitance motor cable. Install an output choke at the controller's output phases.
15	OC5	Frequent/excessively long acceleration processes with overcurrent ($I_{\text{motor}} > I_{\text{rated, inverter}}$)	Check the drive dimensioning; you may need to increase the drive's power rating.
		Continuous overload with $I_{\text{motor}} > 1.05 \times I_{\text{rated, inverter}}$	
16	OC6	I2 x t overload (C0120)	Check the settings in C0120/000. Check the drive dimensioning; you may need to increase the drive's power rating.
17	OC7	I x t warning (C0123)	Check the settings in C0123/000. Check the drive dimensioning; you may need to increase the drive's power rating.
18	OC8	I2 x t warning (C0127)	Check the settings in C0127/000. Check the drive dimensioning; you may need to increase the drive's power rating.

Cross Cutter Appendix

Error number	Error code	Cause	Remedy
20	OU	Mains voltage too high	Check the mains voltage. Check that the mains voltage setting on the controller is correct (code C0173/000). Check the drive dimensioning.
		Too much energy regenerated during braking (regenerative operation)	In single mode with drive connected to mains: - Extend deceleration ramps (codes C0013/000, C0105/000). - If possible: Establish a DC-bus connection to other controllers. - Use a braking unit or power supply unit and brake module/regenerative module. - Check that the mains voltage setting on the controller is correct (code C0173/000). - Check the drive dimensioning. Drives on a DC-bus connection: - Check the fuses on all controllers on the DC-bus connection as well as the fuses on the brake modules/power supply modules/regenerative modules. - Check that the mains voltage setting on the controller is correct (code C0173/000). - Check the drive dimensioning.
		Too much energy regenerated during continuous operation (regenerative operation)	In single mode with drive connected to mains: - If possible: Establish a DC-bus connection to other controllers. - Use a regenerative module. - Check the drive dimensioning. Drives on a DC-bus connection: - Check the fuses on all controllers on the DC-bus connection as well as the fuses on the brake modules/power supply modules/regenerative modules. - Check the drive dimensioning.
30	LU	DC-bus voltage has dropped below the value set in code C0173/000.	Check mains voltage. Check mains contactors/mains fuses. If the incoming power supply is a DC supply: Check the power supply module.
32	LP1	A current-carrying motor phase has failed.	Disconnect the motor and check the windings. Disconnect the supply cable and carry out a loop check.
		The current limit value has been set too low.	Increase the current threshold in code C0599/000.
		This monitoring is not suitable for - Synchronous servo motors - Field frequencies above 480 Hz	Deactivate monitoring by setting code C0597/000 = 3.
50	OH	Ambient temperature too high ($T_{amb} > 40^{\circ}\text{C}$ or 50°C , depending on type/power reduction)	Allow the controller to cool and take action to improve ventilation (control cabinet fans?). Check the ambient temperature in the control cabinet.
		Heatsink badly soiled	Clean the heatsink and protect it against renewed soiling.
		Mounting position of controller	Check the mounting position of the controller and the prescribed minimum clearances below/above it.
51	OH1	Temperature inside the controller $> 90^{\circ}\text{C}$	Check the temperature/ventilation of the control cabinet in which the controller is located.
53	OH3	Overttemperature detection by PTC/KTY: Motor too hot due to continuous overload/overcurrent	Check the drive dimensioning.
		No PTC/KTY connected	Connect PTC/KTY. Deactivate the OH3 error by setting code C0583/000 = 3).
54	OH4	Ambient temperature too high ($T_{amb} > 40^{\circ}\text{C}$ or 50°C , depending on type/power reduction)	Allow the controller to cool and take action to improve ventilation (control cabinet fans?). Check the ambient temperature in the control cabinet.
		Heatsink badly soiled	Clean the heatsink and protect it against renewed soiling.
		Mounting position of controller	Check the mounting position of the controller and the prescribed minimum clearances below/above it.

Cross Cutter Appendix

Error number	Error code	Cause	Remedy
		The value set in code C0122/000 is too low.	Increase the value in code C0122/000.
57	OH7	Overtemperature detection by PTC/KTY: Motor too hot due to continuous overload/overcurrent	Check the drive dimensioning.
		No PTC/KTY connected	Connect PTC/KTY. Deactivate the OH7 error by setting code C0584/000 = 3).
		The value set in code C0121/000 is too low.	Increase the value in code C0121/000.
58	OH8	Overtemperature detection by temperature contact: Motor too hot due to continuous overload/overcurrent	Check the drive dimensioning.
		No temperature contact connected	Connect a temperature contact to terminals T1/T2. Deactivate the OH8 error by setting code C0585/000 = 3).
61	CE0	Fault during transmission of control commands via AIF interface (e.g. PROFIBUS DP)	Plug the automation module in securely. Screw the automation module tight.
62	CE1	CAN-IN1 channel receiving no data or faulty data	Check the CAN bus line (connector X4). Check the CAN terminating resistors (on the first and last CAN nodes). Check the configuration of the CAN node sending to CAN-IN1. Check the voltage supply of the CAN node sending to CAN-IN1. Check and if necessary increase the monitoring time in code C0357/001.
63	CE2	CAN-IN2 channel receiving no data or faulty data	Check the CAN bus line (connector X4). Check the CAN terminating resistors (on the first and last CAN nodes). Check the configuration of the CAN node sending to CAN-IN2. Check the voltage supply of the CAN node sending to CAN-IN2. Check and if necessary increase the monitoring time in code C0357/002.
64	CE3	CAN-IN3 channel receiving no data or faulty data	Check the CAN bus line (connector X4). Check the CAN terminating resistors (on the first and last CAN nodes). Check the configuration of the CAN node sending to CAN-IN3. Check the voltage supply of the CAN node sending to CAN-IN3. Check and if necessary increase the monitoring time in code C0357/003.
65	CE4	Controller has received too many faulty telegrams via the CAN bus and has disconnected itself from the bus.	Check the baud rates of all nodes. Check the wiring: - Check the bus terminating resistors. - Check the shield connection on the cables. - Check the PE connection. - Check the bus load. - Reduce the baud rate (CAUTION: Take the CAN bus cable lengths into account).
66	CE5	CAN time-out (gateway function C0370)	Check the settings in code C0370/000.
70	U15	Undervoltage internal 15 V supply voltage	Check the control voltage for voltage fluctuations or voltage dips.
71	CCr	Processor is overloaded or a program flow error has occurred.	Reduce the processor load: Remove function blocks which are not required from the processing table (C0465/0xx).
		Significant interference injections on the control cables	Install shielded control cables.
		Earth loops in the wiring	Make sure that the wiring is EMC-compliant (see "9300" System Manual, p. 4-34).
72	Pr1	Error transmitting the parameter set CAUTION: The default setting is loaded automatically!	Set the required parameters and save them to non-volatile EEPROM via code C0003/000.
73	Pr2	Error transmitting the parameter set CAUTION: The default setting is loaded automatically!	Set the required parameters and save them to non-volatile EEPROM via code C0003/000.

Cross Cutter Appendix

Error number	Error code	Cause	Remedy
74	<i>PEr</i>	Error in internal program flow	Read out the parameter set and send it to Lenze for further analysis along with the source project (contact Lenze first by calling the Service Hotline: (+49) 5154 82-1111).
75	<i>Pr0</i>	Invalid data in EEPROM when loading parameters from EEPROM to main memory CAUTION: The default setting is loaded automatically!	Set the required parameters and save them to non-volatile EEPROM via code C0003/000, then perform mains switching/disconnection of the control voltage.
79	<i>PI</i>	Error during internal transmission of parameters from EEPROM to main memory	Read out the parameter set and send it to Lenze for further analysis along with the source project (contact Lenze first by calling the Service Hotline: (+49) 5154 82-1111).
		Use of a parameter set which does not match the device (9300EI/ET) in the GDC software/in GDLoader	Use the parameter set matching the controller in the GDC software/in GDLoader, send it to the controller again and save it to non-volatile memory in the controller via C0003/000. Finally, perform mains switching/disconnection of the control voltage.
80	<i>PR6</i>	Too many user codes created	Check and reduce the number of user codes you have created.
82	<i>Sd2</i>	Resolver cable partially or entirely interrupted/disconnected	Check the resolver cable for open circuits. Check the resolver.
		Resolver not in use/not connected	Deactivate monitoring by setting code C0586/000 = 3 if you are not using a resolver.
83	<i>Sd3</i>	Encoder cable break	Disconnect the cable from X9 and check for open circuits.
		Pin 8 at input X9 not assigned	Assign +5 V DC to pin 8 at input X9. OR: Deactivate monitoring by setting code C0587/000 = 3.
85	<i>Sd5</i>	Current master value at X6 (terminals 1 and 2) less than 2 mA	Check the cable for open circuits. Check the master current value encoder. Check the setpoint configuration in C0034/000. OR: Deactivate monitoring by setting code C0598/000 = 3.
86	<i>Sd6</i>	Motor temperature detection encoder at X8 is indicating undefined values.	Check cable at X8 for snug fit and open circuits. Disconnect temperature element (PTC/KTY) and check resistance (must not be ∞). OR: Deactivate monitoring by setting code C0594/000 = 3.
87	<i>Sd7</i>	Absolute value encoder with RS485 interface is not sending any data.	Check cable at X8 for snug fit and open circuits. Check that the absolute value encoder is functioning correctly. Set voltage supply to 8.0 V via C0421/000. No encoder with HIPERFACE interface connected CAUTION: Once the cause of the error has been removed, disconnect the servo controller completely from the power supply.
88	<i>Sd8</i>	The connector on the sin/cos encoder or controller (terminal X8) has detached itself completely (open connection) or is loose.	Check the plug connection on the controller (terminal X8) and on the encoder system
		Error in the connecting cable to the sin/cos encoder	Check the cable connector and conductor pin assignment.
		Encoder electronics defect	Replace a defective encoder.
			Note: In particular for monitoring the encoder on a synchronous machine, set the error response "Error" (TRIP) by setting code C0580 = 0.
89	<i>PL</i>	Error during rotor position adjustment	Repeat rotor position adjustment.
91	<i>EEr</i>	One of the digital signals linked to the <i>DCTRL_bTripSet_b</i> system variable is indicating the status TRUE.	Check the signal source at the <i>DCTRL_bTripSet_b</i> system variable. The "Eer" error can be reset if this signal source has switched to FALSE.
105	<i>H05</i>		You need to contact Lenze (Service Hotline: (+49) 5154 82-1111). Reset only possible by means of mains switching if error cannot be reset: Replace or submit the controller for repair.
107	<i>H07</i>	An incorrect power section was detected during initialisation of the controller.	You need to contact Lenze (Service Hotline: (+49) 5154 82-1111). Reset only possible by means of mains switching if error cannot be reset: Replace or submit the controller for repair.

Cross Cutter Appendix

Error number	Error code	Cause	Remedy
110	<i>H10</i>	Heatsink temperature detection sensor is indicating undefined values, an internal temperature element might be faulty.	You need to contact Lenze (Service Hotline: (+49) 5154 82-1111). Reset only possible by means of mains switching if error cannot be reset: Replace or submit the controller for repair.
111	<i>H11</i>	In-drive temperature detection sensor is indicating undefined values, an internal temperature element might be faulty.	You need to contact Lenze (Service Hotline: (+49) 5154 82-1111). Reset only possible by means of mains switching if error cannot be reset: Replace or submit the controller for repair.
190	<i>NErr</i>	Drive mechanical overload (e.g. active load on hoists too great)	Check drive dimensioning. Check mechanics for jamming or blockages. Increase ramp times to reduce dynamics.
		Torque limit reached	Increase torque limit: - Directly via <i>MCTRL_nHiMLim_a</i> , <i>MCTRL_nLoMLim_a</i> system variables - Indirectly via current limit in code C0022/000 - Check the motor data (codes C0081/000 to C0092/000).
		Error in speed feedback system	Check speed feedback: - Check the parameter setting of the speed feedback system (C0025/000, C0495/000). - Check the rotor position and readjust if necessary (C0058/000, C0095/000). - Check the wiring/polarity of the speed feedback system.
		Tolerance window C0576/000 set too low	Increase the size of the tolerance window by adjusting the value in code C0576/000.
200	<i>NMAX</i>	Active load (e.g. on hoists) too great	Check the current limit in code C0022/000 and increase it if possible (CAUTION: Take the motor current limit into account!). Check drive dimensioning.
		Torque limit reached	Increase torque limit: - Directly via <i>MCTRL_nHiMLim_a</i> , <i>MCTRL_nLoMLim_a</i> system variables - Indirectly via current limit in code C0022/000
		Speed feedback: Actual speed/rotor position calculated incorrectly	Check feedback system selection (code C0025/000). Check the motor data (codes C0081/000 to C0092/000). Check the rotor position and readjust if necessary (C0058/000, C0095/000).
201	<i>Overrun Task1 Id2</i>	The runtime of the corresponding task is longer than defined in the watchdog time (as programmed in the task configuration in DDS).	Reduce the task load by moving non-time-critical program sections to lower-priority tasks, tasks with longer interval times or the cyclic task. Increase the task's interval time. Check for a division by 0 in arithmetic operations. Check your program code for infinite loops (REPEAT/UNTIL, WHILE, FOR loops) and recursive calls and remove them. If you are using free CAN objects: Reduce the number of telegrams received per time with identifiers which do not correspond to system blocks CAN1, CAN2 or CAN3. If data consistency has been activated: Deactivate the default setting for data consistency in DDS (target system menu item) and use the <i>Lenze32BitTransferDrv.lib</i> library.
202	<i>Overrun Task2 Id3</i>		
203	<i>Overrun Task3 Id4</i>		
204	<i>Overrun Task4 Id5</i>		
205	<i>Overrun Task5 Id6</i>		
206	<i>Overrun Task6 Id7</i>		
207	<i>Overrun Task7 Id8</i>		
208	<i>Overrun Task8 Id9</i>		
209	<i>Float error Sys-Task Id0</i>	An error has occurred during a floating point operation in the system task.	You need to contact Lenze (Service Hotline: (+49) 5154 82-1111).
210	<i>Float error PLC_PRG Id1</i>	An error has occurred during a floating point operation in the cyclic task (PLC_PRG or a called POU).	Check all floating point operations in the cyclic task and subroutines called by it. Remove/prevent divisions by 0.
211	<i>Float error Task1 Id2</i>	An error has occurred during a floating point operation in the cyclic task (PLC_PRG or a called POU).	Check all floating point operations in the cyclic task and subroutines called by it. Remove/prevent divisions by 0.
212	<i>Float error Task2 Id3</i>		
213	<i>Float error Task3 Id4</i>		
214	<i>Float error Task4 Id5</i>		

Cross Cutter Appendix

Error number	Error code	Cause	Remedy
215	<i>Float error Task5 Id6</i>		
216	<i>Float error Task6 Id7</i>		
217	<i>Float error Task7 Id8</i>		
218	<i>Float error Task8 Id9</i>		
219	<i>Overrun PLC_PRG Id0</i>	The cyclic task is taking too long or cannot be completed.	Check your program code for infinite loops (REPEAT/UNTIL, WHILE, FOR loops) and recursive calls and remove them.
220	<i>NoCredits</i>	The target system does not support the technology libraries in the project (e.g. positioning technology, electronic cam).	Check code C2115/000: If this code is not displaying any technology units, contact your Lenze sales company or the Lenze Service Hotline (+49) 5154 82-1111.
230	<i>NoPrg</i>	No executable PLC program stored on the target system	Transfer a compatible PLC program to the target system via DDS (PRO file) or GDLoader (BIN file).
231	<i>Unallowed Lib</i>	The target system does not support one of the libraries used in the source project.	Use a library which matches the target system. Make sure that the target system version release supports the version release of the library being used.
232	<i>NoAddData</i>	No additional data (e.g. LC9 file with tables of values for the electronic cam) found.	Transfer the project again together with the matching file for the additional data.
240	<i>TxBuffer Overflow</i>	Too many CAN telegrams sent using the <i>LenzeCanDrv.lib</i> library. The send data memory is full.	Reduce the number of CAN telegrams sent via the <i>L_CanPdoTransmit</i> function blocks by - Increasing the task's interval time - Selecting time-controlled instead of event-controlled transmission via <i>byTransmitMode</i> - Increasing the transmission cycle at the <i>tRepeatTime</i> input.
			Increase the baud rate of the system bus in code C0351/000.
241	<i>RxBuffer Overflow</i>	Too many CAN telegrams received using the <i>LenzeCanDrv.lib</i> library; further processing in the PLC program is taking longer than it ought to. The receive data memory is full.	Reduce the number of received telegrams to the free CAN objects processed following receipt by <i>L_CanPdoReceive</i> type function blocks. Call the program sections in which received free CAN telegrams are processed following receipt at shorter interval times.
250	<i>NoApplFlash</i>	The data memory for additional data/application data is missing.	Replace the current target system with a suitable target system featuring the memory module for additional data/application data. Contact your Lenze sales company or the Lenze Service Hotline (+49) 5154 82-1111.
251	<i>ChckErr (ApplRAM)</i>	The application data checksum in the main memory does not match the reference checksum.	Contact your Lenze sales company or the Lenze Service Hotline (+49) 5154 82-1111.
252	<i>DwnldErr (ApplFlash)</i>	An error occurred during the transmission of application data to the non-volatile application data memory.	Check the data circuit: - Check that the bus connectors and terminating resistors are connected correctly and have the correct polarity. - Check that the parameterisation of the transmission path is unique (e.g. more than one CAN node with the same bus address, baud rate, excessive bus load). - You may need to remove nodes from the system bus to reduce the bus load.
			The source file associated with the additional data is corrupt (impermissible changes): Regenerate the source data (e.g. LC9 file with profile data) and repeat the transmission.
260	<i>Err NodeGuard</i>	"Life Guarding Event": The PLC (CAN slave) has not received a "Node Guarding" telegram within the "Node Life Time" of the CAN master.	Check the CAN master to ascertain why no "Node Guarding" telegrams are being sent.

6.3.2 Application error messages

Error number	Error code	Cause	Remedy
400	<i>PosLimit Switch</i>	Positive limit switch reached	Retract the positive limit switch: Acknowledge the error message. Run the drive in a negative direction, e.g. via manual jog, until the positive limit switch retracts.
		Limit switch signal not connected to terminal E2	Set the level inversion for digital input I2 to inverse logic (C0114/002 = 1) or Establish a permanent wire-bound connection between terminal I2 and 24 V DC.
401	<i>NegLimit Switch</i>	Negative limit switch reached	Retract the negative limit switch: Acknowledge the error message. Run the drive in a positive direction, e.g. via manual jog, until the negative limit switch retracts.
		Limit switch signal not connected to terminal E1	Set the level inversion for digital input I1 to inverse logic (C0114/001 = 1) or Establish a permanent wire-bound connection between terminal I1 and 24 V DC.
402	<i>PosSWLimit</i>	Positive software limit position reached/overtravelled	Run the drive in a negative direction in the permissible traversing range. Set the limit value for the positive software limit position (C3223/000) to a higher value.
403	<i>NegSWLimit</i>	Negative software limit position reached/overtravelled	Run the drive in a positive direction in the permissible traversing range. Set the limit value for the negative software limit position (C3224/000) to a lower value.
405	<i>FollError</i>	The drive cannot follow the setpoint, the current following error is greater than the value set in C3218/001.	- Increase the current limit in C0022/000, thereby increasing the torque (CAUTION: Be aware of the increase in motor temperature!). - Reduce the dynamic requirements (accelerations/decelerations). - Increase the following error limit in C3218/001. - You may need to check the drive dimensioning.
406	<i>HomError</i>	An attempt has been made to approach an absolute position without the measuring system zero point being known to the drive.	- Prior to carrying out the first positioning operation, perform homing (manual referencing, homing via positioning program) and/or set the home position in the positioning program. - Define the measuring system zero point for operation with absolute value encoders. OR: - Select relative positioning if relative positioning (no absolute measuring system!) is possible.
408	<i>UserQSP</i>	Automatic operation has been aborted due to a quick stop command being sent by the user.	- A quick stop can be triggered via the terminals, a bus system (AIF or CAN bus) or the control code C0135/000 (system variable <i>MCTRL_bQspOut_b</i>). - Reset the quick stop command, acknowledge the error and perform a controlled reset to return the drive to the required operating mode.
409	<i>MaxVelConflit</i>	The traversing speed defined in the positioning profile is generating a motor speed in excess of the maximum speed C0011/000.	- Reduce the speed value in the traversing profile. - Alternatively, increase the maximum speed in code C0011/000. Take the motor limit speed, maximum system speeds, etc. into account when increasing the maximum speed.
410	<i>DriveNotRdy</i>	Automatic operation has been aborted due to the user setting a controller inhibit.	- A controller inhibit can be triggered via terminal 28, a bus system (AIF or CAN bus), control codes C0040/000 and C0135/000 or system variables <i>DCTRL_bCinh1_b</i> and <i>DCTRL_bCinh2_b</i>. - Re-enable the target system, acknowledge the error and perform a controlled reset to return the drive to the required operating mode.

Cross Cutter Appendix

Error number	Error code	Cause	Remedy
412	<i>InitError</i>	An error has occurred during initialisation following connection of the mains power supply or after a restart of the PLC program.	- Check the basic settings, e.g. axis data, operating modes, position values for software limit position, offsets, etc. ... if any values have changed, save them to non-volatile memory by setting C0003/000 = 1. - Once you have rectified the cause, disconnect the supply voltage briefly so that the target system can reinitialise.
414	<i>SwapPosMode</i>	In some applications, the target position cannot be reached via the preselected positioning direction.	Change the positioning mode, acknowledge the error message and restart positioning.
415	<i>XYPosChange Limit</i>	In applications in which tables of values are used (electronic cam) a significant change in position on the input axis (x value) will generate a skip of more than 20 points in the table of values for the cam function.	Do not adjust the input position (x value of cam function) point by point. Use the profile generators (e.g. function block <i>L_CamPosCtrlLin</i> from library <i>LenzeCamControl1Vxxyy.lib</i>).
416	<i>PosToExtNo Speed</i>	The speed setpoint for positioning in relation to the external master angle (electronic cam) is zero.	Enter a speed setpoint value > 0 for positioning in relation to the external master angle.
417	<i>Invalid SoftwareLimis</i>	The values for the software limit positions are invalid.	Enter a higher value for the positive software limit position (C3223/000) than for the negative software limit position (C32234/000).
418	<i>XTpPosition OutOfLimit</i>	The default value for the sensor position (master angle correction on the electronic cam) is less than zero or greater than the master cycle length.	Enter values between 0 and the maximum master cycle length for the touch-probe position of the master value.
419	<i>StateBusError</i>	There are errors on one of the other drive axes connected to the state bus. All connected controllers are set to the <i>StateBusError</i> error status via the ST terminals.	- Identify the drive on the state bus connection which caused the error, rectify the cause of the error and reset the error on this drive. - Then reset the state bus error on all other drives connected on the state bus.
420	<i>OU_MCTRL Error</i>	Operation has been aborted due to a short-time overvoltage on the DC bus.	An overvoltage on the DC bus is detected via system variable <i>MCTRL_bOverVoltage_b</i>: If this error occurs, proceed as follows: - Acknowledge the error. - Perform a controlled reset of the drive to return it to the required operating mode. To identify the cause of the error, please refer to error message no. 20 (OU) in the system error messages.
421	<i>LU_MCTRL Error</i>	Operation has been aborted due to an undervoltage on the DC bus.	An undervoltage on the DC bus is detected via system variable <i>MCTRL_bUnderVoltage_b</i>: If this error occurs, proceed as follows: - Acknowledge the error. - Perform a controlled reset of the drive to return it to the required operating mode. To identify the cause of the error, please refer to error message no. 30 (LU) in the system error messages.
422	<i>InvalidProfile Data</i>	The positioning profile defined contains data whose values are implausible in the traversing profile or in relation to the previous traversing profile.	Check the default values for the current traversing profile and set these to plausible values.
423	<i>PosIntegrator Overflow</i>	The phase integrator in the profile generator has exceeded the positive limit value ($2^{31}-1$ increments) or undershot the negative limit value (2^{31} increments).	Select the traversing targets for absolute or relative positioning without resetting the path integrators in order to ensure that the values remain within the internal value range of $2^{31} \dots 2^{31}-1$ increments.
501	<i>WrongProfile ForCalc</i>	A motion profile currently being used by the application has been calculated.	Use a motion profile not currently in use by the application for your calculation.

Cross Cutter Appendix

Error number	Error code	Cause	Remedy
502	<i>ExtUserError</i>	An external user error has been triggered.	Rectify the cause of the external user error.
503	<i>ErrorCalc CuttingProfile</i>	An error occurred whilst calculating the motion profile for the new cutting length.	Check the status message for the corresponding calculation block in code C3661 (<i>g_nCrossCutterCalcStateCuttingProfile</i>) and rectify the error in accordance with the information in the status message.
504	<i>ErrorCalc RejectProfile</i>	An error occurred whilst calculating the motion profile for the new reject length.	Check the status message for the corresponding calculation block in code C3662 (<i>g_nCrossCutterCalcStateRejectProfile</i>) and rectify the error in accordance with the information in the status message.
505	<i>ErrorCalc TestProfile</i>	An error occurred whilst calculating the motion profile for the new test length.	Check the status message for the corresponding calculation block in code C3663 (<i>g_nCrossCutterCalcStateTestProfile</i>) and rectify the error in accordance with the information in the status message.
506	<i>ErrorCalc StartStop Profile</i>	An error occurred whilst calculating the motion profile for the new start/stop length.	Check the status message for the corresponding calculation block in code C3664 (<i>g_nCrossCutterCalcStateStartStopProfile</i>) and rectify the error in accordance with the information in the status message.
507	<i>MaxNumberOf TPOutOfWin</i>	The number of cycles permitted without a mark signal appearing in the mark window has been exceeded. A warning signal is being generated.	Check the reason why no print marks have appeared in the window. Possible causes: • Print-mark sensor faulty • The mark window is too small. • The print-mark-sensor distance has been calculated incorrectly. • Slip in the system means that the print marks are not appearing in the mark window.

6.3.3 User-defined error messages

In the programmable variant of the prepared solution, users can program their own error messages in addition to the standard errors. To do this, proceed as follows:

1. Create a new network by copying and pasting one of the networks from the UserErrors(PRG) POU.
2. Rename the copied FB FWM.
3. Link the bExecute input from the instanced FB FWM to the boolean variable to be monitored. In order for errors to be resolved correctly, the boolean variable to be monitored may only be TRUE for the specific machine state. The trigger signal must be pending for at least 5 ms. You can safeguard this by using a pulse timer, for example (FB TP).
4. Set the required error number at the wFaultNumber input of the instanced FB FWM. Only error numbers from 508...999 may be used. Error numbers below 508 are already in use.

Input FB FWM		
Name	Data type	Value/Meaning
wFaultNumber	WORD	Error number for the user error message
		508 ... 999 Error number

5. You can assign an error message (text message) to the set error number in the ErrorCode (ENUM) data type (see Chapter 6.3.4).
6. Set the required error response at the byTypeOfFault input. You can select from the following options:

Input FB FWM		
Name	Data type	Value/Meaning
byTypeOfFault	BYTE	Selection of response to user error message
		0 TRIP
		1 Message
		2 Warning
		3 Off (no response)
		4 Fail QSP

7. Link the wActFaultNumber input to the *g_wActFault* global variable.

Input FB FWM		
Name	Data type	Value/Meaning
wActFaultNumber	WORD	Error channel for the system; link this variable to the <i>g_wActFault</i> global variable.



Note!

The wActFaultNumber input of the FB FWM acts as an error channel for communication with the internal "FailManager" and must be linked to the *g_wActFault* global error channel variable.

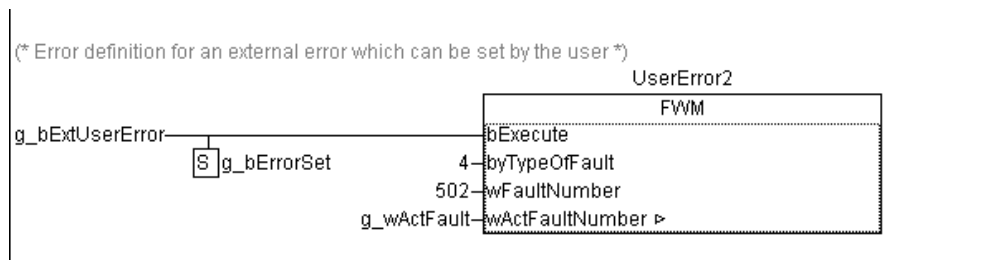


Figure 43: Example for programming a user-defined error

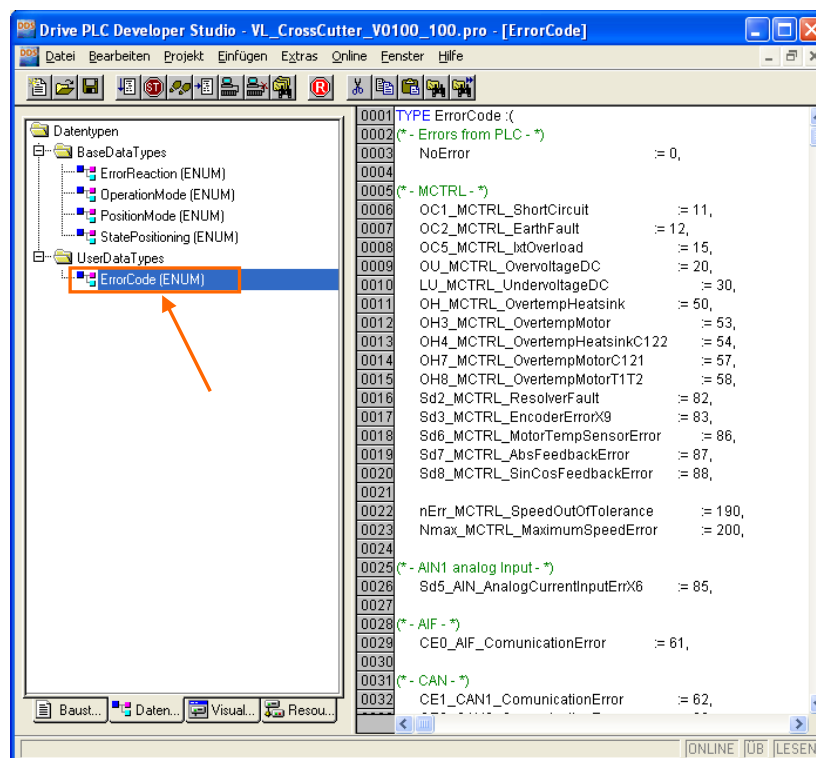
6.3.4 ENUM list for error numbers

An error number alone provides little information. The use of what is known as an ENUM list, which assigns a unique designation to each error number, enables more information to be provided. In online operation, the assigned text, rather than the error number, will be displayed, facilitating monitoring and diagnostics (e.g. via the visualisation screens in the DDS programming tool).

In the context of the prepared solution, the ENUM list is supplied with the error designations users will already be familiar with from the cam software packages and can be expanded accordingly by the addition of user-defined errors.

Proceed as follows to add user-defined errors:

1. Call up the *ErrorCode* (ENUM list) variable structure via **Data Types** → **ErrorCode**:



2. Scroll to the end of the ENUM list:

```
0120 (*User Errors *****)
0121 WrongProfileForCalc :=501,
0122 ExtUserError:=502,
0123 ErrorCalcCuttingProfile:=503,
0124 ErrorCalcRejectProfile:=504,
0125 ErrorCalcTestProfile:=505,
0126 ErrorCalcStartStopProfile:=506,
0127 MaxNumberOfTPOutOfWin:=507
0128);
0129 END_TYPE
0130
```

3. Add a brief designation appropriate for the new error message to the end of the ENUM list:

```
0120 (*User Errors *****)
0121 WrongProfileForCalc :=501,
0122 ExtUserError:=502,
0123 ErrorCalcCuttingProfile:=503,
0124 ErrorCalcRejectProfile:=504,
0125 ErrorCalcTestProfile:=505,
0126 ErrorCalcStartStopProfile:=506,
0127 MaxNumberOfTPOutOfWin:=507
0128);
0129 END_TYPE
0130
```

Do not forget to add a comma to the previous line!

The last line in the ENUM list does not finish with a comma!



6.4 Signal flow diagram for core function

The following diagram illustrates the signal flow within the core function of the Cross Cutter prepared solution.

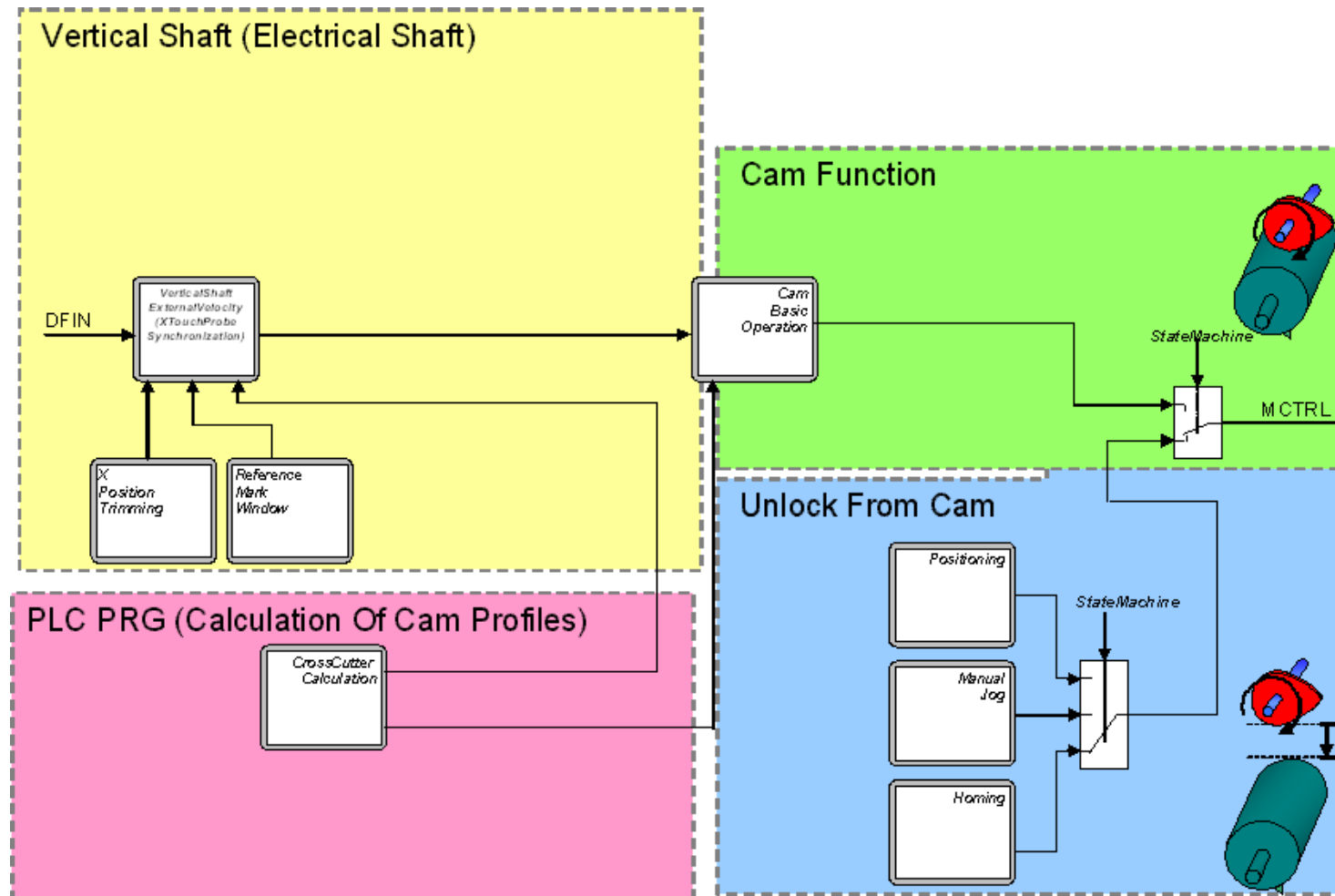


Figure 44: Signal flow diagram for the prepared solution

6.5 Interface signals

The Cross Cutter prepared solution has been designed to enable users to control their applications via various communication channels. Provision has been made for applications to be controlled via digital inputs, the AIF interface, CAN interface or codes. A multiplexer has been developed to make parameter setting as user-friendly as possible.

Both in the parameterisable and in the programmable variant, the prepared solution's interfaces have been designed to offer maximum flexibility. As the programmable variant offers users complete freedom, this section will concentrate solely on the parameterisable variant.

a) Digital inputs/outputs: for 9300 Servo PLC

The assignment of digital inputs/outputs is defined as follows:

Signal	Cross cutter
<i>DIGIN_bCinh_b</i>	Controller inhibit
<i>DIGIN_bIn1_b</i>	QSP
<i>DIGIN_bIn2_b</i>	Not assigned
<i>DIGIN_bIn3_b</i>	Homing switch
<i>DIGIN_bIn4_b</i>	TP sensor homing run
<i>DIGIN_bIn5_b</i>	Print-mark sensor
<i>DIGOUT_bOut1_b</i>	Locked to the motion profile. Cross cutter ready for operation. (<i>g_bLockToCamDone</i>)
<i>DIGOUT_bOut2_b</i>	Calculation of all motion profiles complete. (<i>g_bAllCalcDone</i>)
<i>DIGOUT_bOut3_b</i>	Open reject gate. (<i>g_bOpenRejectGate</i>)
<i>DIGOUT_bOut4_b</i>	System error. (<i>g_bGlobalError</i>)

b) Digital inputs/outputs: for ECS

The assignment of digital inputs/outputs is defined as follows:

Signal	Cross cutter
<i>SI1</i>	Controller inhibit
<i>DIGIN_bIn1_b</i>	Print-mark sensor
<i>DIGIN_bIn2_b</i>	TP sensor homing run
<i>DIGIN_bIn3_b</i>	Homing switch
<i>DIGIN_bIn4_b</i>	QSP
<i>DIGOUT_bOut1_b</i>	Open reject gate. (<i>g_bOpenRejectGate</i>)



Caution!

Digital input *DIGIN_bIn1_b* using 9300 Servo PLC or *DIGIN_bIn4_b* using ECS is inverted via the code initialisation values (see Chapter 6.2.2) in order to achieve wire-break protection (0 V = QSP).

Because there are so many signals, a single interface via hardware digital signals is not sufficient to control minimum functionality. For this reason, the vast majority of control and status signals can be preselected via other channels (data bus interfaces, control/status codes).

c) Control/Status interface with master control

Codes known as control and status codes can be used on systems which are to be controlled and monitored solely on the basis of parameter information. The parameterisable variant is supplied with the control and status interface preconfigured. An overview of signal assignment and reconfiguration options (via the multiplexer) for control and status signals appears below.

Control interface

		Meaning
Word 0	System control word	Bit 00: Dependent on the setting in code C4000/000 Bit 01: Dependent on the setting in code C4000/000 Bit 02: Dependent on the setting in code C4000/000 Bit 03: Dependent on the setting in code C4000/000 Bit 04: Dependent on the setting in code C4000/000 Bit 05: Dependent on the setting in code C4000/000 Bit 06: Dependent on the setting in code C4000/000 Bit 07: Dependent on the setting in code C4000/000 Bit 08: Dependent on the setting in code C4000/000 Bit 09: Dependent on the setting in code C4000/000 Bit 10: Dependent on the setting in code C4000/000 Bit 11: Dependent on the setting in code C4000/000 Bit 12: Dependent on the setting in code C4000/000 Bit 13: Dependent on the setting in code C4000/000 Bit 14: Dependent on the setting in code C4000/000 Bit 15: Dependent on the setting in code C4000/000 C4000/000 = 0: The DCTRL_wAIF1Ctrl and DCTRL_wCAN1Ctrl system variables are always set to zero. C4000/000 = 1, 2, 3: The drive control word is copied to DCTRL_wCAN1Ctrl; DCTRL_wAIF1Ctrl is set to zero. C4000/000 = 4, 5, 6: The drive control word is copied to DCTRL_wAIF1Ctrl; DCTRL_wCAN1Ctrl is set to zero.
Word 1	Data word 1	Control word: Selection of operating mode
Word 2	Application control word	Bit 00: Start/stop homing, only in conjunction with controller enable (g_bHomingStart) Bit 01: Set home position (g_bHomePositionSet) Bit 02: Reset home position (g_bHomePositionReset) Bit 03: Manual jog in positive direction (CW) (g_bManualJogPos) Bit 04: Manual jog in negative direction (CCW) (g_bManualJogNeg) Bit 05: Perform positioning (g_bMotionStart) Bit 06: Activate external user error (g_bExtUserError) Bit 07: Reset error (g_bErrorReset) Bit 08: Inhibit controller (g_bControllerInhibit) Bit 09: QSP (g_bUserQuickstop) Bit 10: Activate the DFIN test signal to simulate the DFIN signal (g_bUseTestsignal) Bit 11: Reset the cutting counter for cutting operation (g_bResetCuttingProfileCounter) Bit 12: Reset the cutting counter for reject operation (g_bResetRejectProfileCounter) Bit 13: Reset the cutting counter for test operation (g_bResetTestProfileCounter) Bit 14: Start and stop selected cross-cutter operating mode (g_bStartStopOperation) Bit 15: Activate the variable to accept a newly calculated cutting length immediately and work with the new cutting length (g_bWorkWithNewCuttingLength) Bit 16: Activate the variable to accept a newly calculated reject length immediately and work with the new cutting length (g_bWorkWithNewRejectLength) Bit 17: Activate the variable to accept a newly calculated test length immediately and work with the new cutting length (g_bWorkWithNewTestLength) Bit 18: Activate X trimming and switch from the calculated mark sensor distance to the mark sensor distance specified by X trimming (g_bUseXPositionTrimming) Bit 19: Key this input to perform X trimming in a positive direction at the set speed (g_bXAxisTrimmingPosDirection) Bit 20: Key this input to perform X trimming in a negative direction at the set speed (g_bXAxisTrimmingNegDirection) Bit 21: Enable print-mark synchronisation (g_bXTPEnable) Bit 22: Enable ramp generator for print-mark synchronisation (g_bRfgXTPEnable) Bit 23: Reset counter for number of print marks located outside the window (g_bResetCounterTPOutOfWindow) Bit 24: Abort cross-cutter operation (g_bAbordCrossCutter) Bit 25: Execute X trimming absolute positioning (g_bLoadSetXAxisTrimmingTPPos) Bit 26: (not assigned) Bit 27: (not assigned) Bit 28: (not assigned) Bit 29: (not assigned) Bit 30: (not assigned) Bit 31: (not assigned)
Word 3		

Cross Cutter Appendix

The signal sources for the data words (designated *Word 0*, *Word 1*, *Word 2* and *Word 3*) are dependent on the operating mode in C4000/000:

Signal source	Word 0	Word 1	Word 2	Word 3
C4000/000 = 0	C0135/000	C4137/000	C4135/000	
C4000/000 = 1	CAN1_wDctrlCtrl	CAN1_nlnW1_a	CAN1_nlnW2_a	CAN1_nlnW3_a
C4000/000 = 2	CAN2_nlnW1_a	CAN2_nlnW2_a	CAN2_nlnW3_a	CAN2_nlnW4_a
C4000/000 = 3	CAN3_nlnW1_a	CAN3_nlnW2_a	CAN3_nlnW3_a	CAN3_nlnW4_a
C4000/000 = 4	AIF1_wDctrlCtrl	AIF1_nlnW1_a	AIF1_nlnW2_a	AIF1_nlnW3_a
C4000/000 = 5	AIF2_nlnW1_a	AIF2_nlnW2_a	AIF2_nlnW3_a	AIF2_nlnW4_a
C4000/000 = 6	AIF3_nlnW1_a	AIF3_nlnW2_a	AIF3_nlnW3_a	AIF3_nlnW4_a

The illustration of the 32-bit application control word (*Word 3* and *Word 4*) on the previous page shows the factory-set default configuration. Users can also adapt the configuration of the application control word in accordance with their individual needs by setting the values of codes C4010/001 to C4010/032 accordingly.

An example of how the **GDC** parameterisation software can be used to reconfigure the application control word bit by bit appears below:

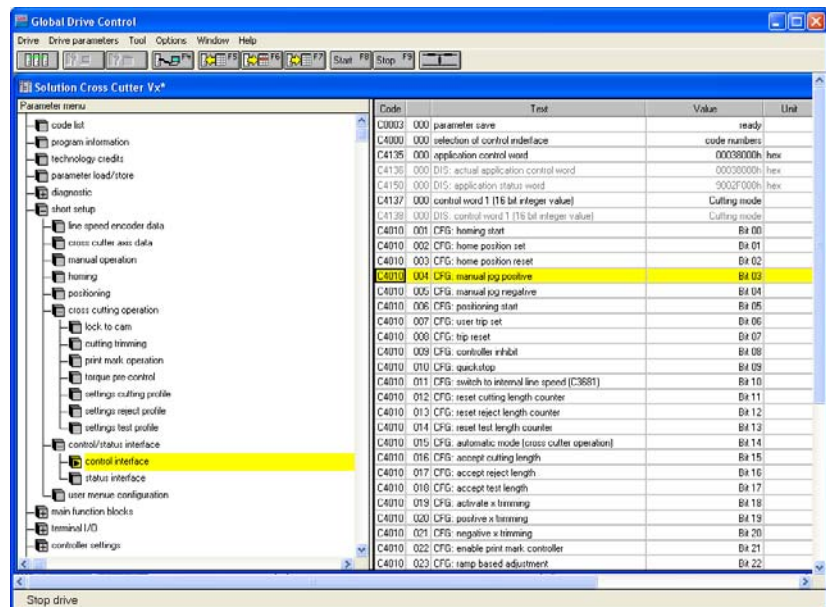
You will find the code in the complete code list or in the menu (Short Setup → Control/Status Interface → Control Interface):

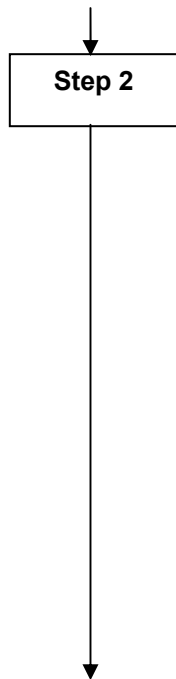
Start

Step 1

Select code C4010/00x which is assigned to the function to be configured.

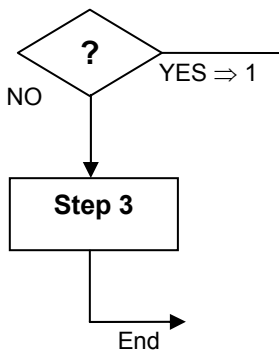
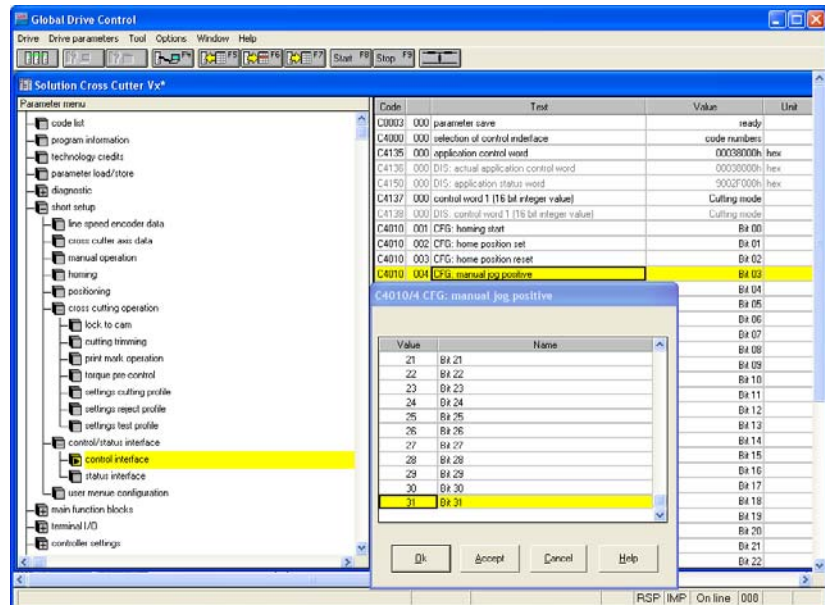
Example: The control signal for manual jog in a positive direction is defined in code C4010/004:





Click on the code and select the required control bit.

Example: Bit 31 is to be used to control manual jog in a positive direction on the cross-cutter axis. Set C4010/004 = 31 accordingly and confirm your entry with Ok.



To reconfigure other control bits, proceed as described in Step 1.

Save the modified settings to non-volatile memory with C0003/000 = 1.

Status interface

Meaning		
Bit 00: Dependent on the setting in code C4000/000	System status word	Word 0
Bit 01: Dependent on the setting in code C4000/000		
Bit 02: Dependent on the setting in code C4000/000		
Bit 03: Dependent on the setting in code C4000/000		
Bit 04: Dependent on the setting in code C4000/000		
Bit 05: Dependent on the setting in code C4000/000		
Bit 06: Dependent on the setting in code C4000/000		
Bit 07: Dependent on the setting in code C4000/000		
Bit 08: Dependent on the setting in code C4000/000		
Bit 09: Dependent on the setting in code C4000/000		
Bit 10: Dependent on the setting in code C4000/000		
Bit 11: Dependent on the setting in code C4000/000		
Bit 12: Dependent on the setting in code C4000/000		
Bit 13: Dependent on the setting in code C4000/000		
Bit 14: Dependent on the setting in code C4000/000		
Bit 15: Dependent on the setting in code C4000/000		
Error number (<i>g_wErrorNumber</i>)	Data word 1	Word 1
Bit 00: System error (<i>g_bGlobalError</i>)	Application status word	Word 2
Bit 01: Following error warning (<i>g_bFollowErrorWarn</i>)		
Bit 02: Following error (<i>g_bFollowErrorQSP</i>)		
Bit 03: Positioning in progress (<i>g_bToolPositioningBusy</i>)		
Bit 04: Target position reached (<i>g_bToolPositioningDone</i>)		
Bit 05: System locking to motion profile (<i>g_bLockToCamBusy</i>)		
Bit 06: System locked to motion profile (<i>g_bLockToCamDone</i>)		
Bit 07: Mark correction active (<i>g_bXTpSyncBusy</i>)		
Bit 08: Calculation sequence for new cross-cutter motion profile for a new cutting length in progress (<i>g_bCrossCutterCalcBusyCuttingProfile</i>)		
Bit 09: Calculation sequence for new cross-cutter motion profile for a new reject length in progress (<i>g_bCrossCutterCalcBusyRejectProfile</i>)		
Bit 10: Calculation sequence for new cross-cutter motion profile for a new test length in progress (<i>g_bCrossCutterCalcBusyTestProfile</i>)		
Bit 11: Calculation sequence for new cross-cutter motion profile for a new start/stop length in progress (<i>g_bCrossCutterCalcBusyStartStopProfile</i>)		
Bit 12: Calculation sequence for new cross-cutter motion profile for a new cutting length completed successfully. Starting a new calculation sequence will reset this status output to FALSE (<i>g_bCrossCutterCalcDoneCuttingProfile</i>).		
Bit 13: Calculation sequence for new cross-cutter motion profile for a new reject length completed successfully. Starting a new calculation sequence will reset this status output to FALSE (<i>g_bCrossCutterCalcDoneRejectProfile</i>).		
Bit 14: Calculation sequence for new cross-cutter motion profile for a new test length completed successfully. Starting a new calculation sequence will reset this status output to FALSE (<i>g_bCrossCutterCalcDoneTestProfile</i>).		
Bit 15: Calculation sequence for new cross-cutter motion profile for a new start/stop length completed successfully. Starting a new calculation sequence will reset this status output to FALSE (<i>g_bCrossCutterCalcDoneStartStopProfile</i>).		
Bit 16: Homing activated (<i>g_bHomingBusy</i>)		Word 3
Bit 17: Home position detected (<i>g_bHomingDone</i>)		
Bit 18: "Homing error" status signal (<i>g_bHomingError</i>)		
Bit 19: (not assigned)		
Bit 20: The cross cutter is operating as an asynchronous cross cutter (<i>g_bUseAsynchronousCrossCutter</i>).		
Bit 21: The cross cutter is operating as a synchronous cross cutter (<i>g_bUseSynchronousCrossCutter</i>).		
Bit 22: The cross cutter is operating in "Continuous cutting" mode (<i>g_bUseCuttingProfile</i>).		
Bit 23: The cross cutter is operating in "Reject operation" mode (<i>g_bUseRejectProfile</i>).		
Bit 24: The cross cutter is operating in "Test operation" mode (<i>g_bUseTestProfile</i>).		
Bit 25: Opening reject gate (<i>g_bOpenRejectGate</i>)		
Bit 26: The maximum number of cycles without a mark signal appearing in the mark window has been exceeded (<i>g_bMaxNumberOfTPOutOfWindowIncreased</i>).		
Bit 27: System ready for print-mark control (<i>g_bSystemReadyToUseTP</i>)		
Bit 28: Using print-mark register (<i>g_bUseShiftedTP</i>)		
Bit 29: Knife in material (NOT <i>g_bKnifeNotInMaterial</i>)		
Bit 30: (not assigned)		
Bit 31: All calculation sequences for new cross-cutter motion profiles have been completed successfully. Starting a new calculation sequence will reset this status output to FALSE (<i>g_bAllCalcDone</i>).		

Cross Cutter Appendix

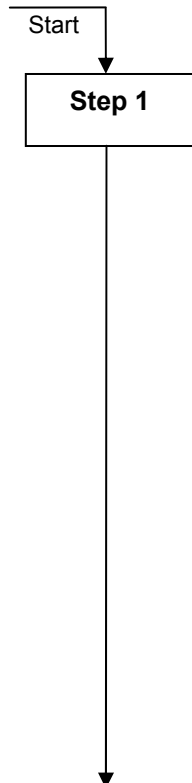
The individual status words are output to system variables (designated *Word 0*, *Word 1*, *Word 2* and *Word 4*) dependent on the operating mode in C4000/000:

Signal source	Word 0	Word 1	Word 2	Word 3
C4000/000 = 0	C0135/000	C4152/000	C4150/000	
C4000/000 = 1	CAN1_wDctrlStat	CAN1_nOutW1_a	CAN1_nOutW2_a	CAN1_nOutW3_a
C4000/000 = 2	CAN2_nOutW1_a	CAN2_nOutW2_a	CAN2_nOutW3_a	CAN2_nOutW4_a
C4000/000 = 3	CAN3_nOutW1_a	CAN3_nOutW2_a	CAN3_nOutW3_a	CAN3_nOutW4_a
C4000/000 = 4	AIF1_wDctrlStat	AIF1_nOutW1_a	AIF1_nOutW2_a	AIF1_nOutW3_a
C4000/000 = 5	AIF2_nOutW1_a	AIF2_nOutW2_a	AIF2_nOutW3_a	AIF2_nOutW4_a
C4000/000 = 6	AIF3_nOutW1_a	AIF3_nOutW2_a	AIF3_nOutW3_a	AIF3_nOutW4_a

The illustration on the previous page shows the factory-set default configuration of the 32-bit application status word. Users can also adapt the configuration of the application status word in accordance with their individual needs by setting the values of codes C4012/001 to C4012/032 accordingly.

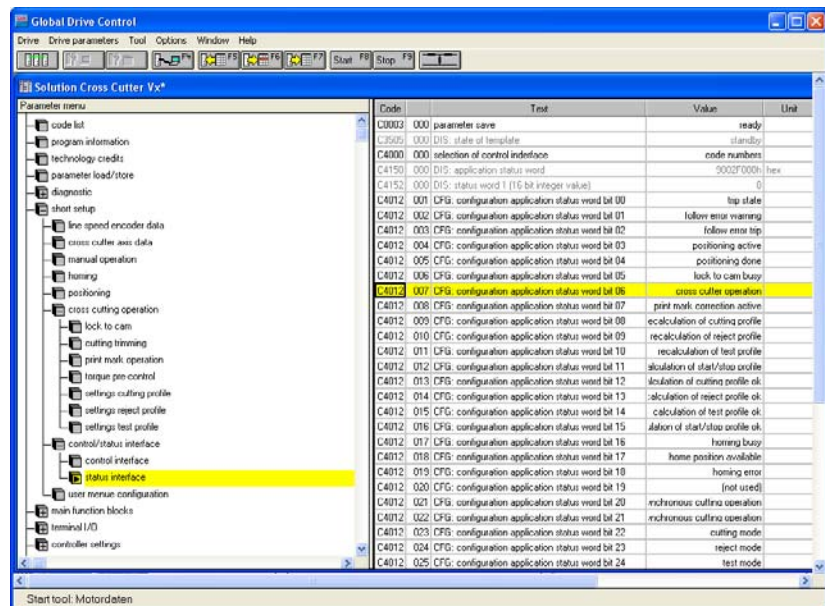
An example of how the **GDC** parameterisation software can be used to reconfigure the application status word bit by bit appears below:

You will find the code in the complete code list or in the menu Short Setup → Control/Status Interface → Status Interface:

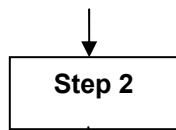


Select the application status word bit you wish to reconfigure.

Example: To assign bit 6 of the application status word, select code C4012/007:

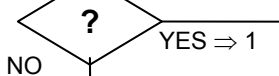
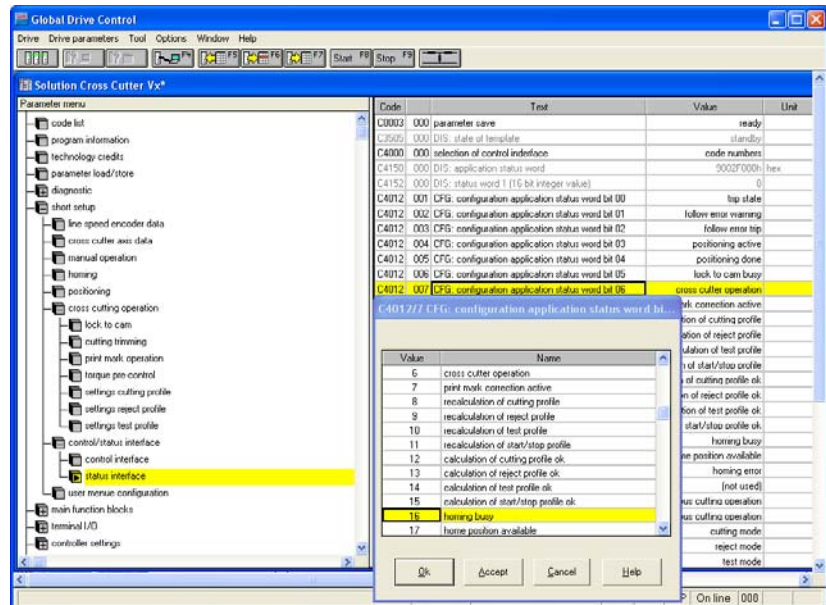


Cross Cutter Appendix

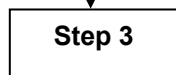


Click on the code and select the status information you wish to assign to this status bit for display.

Example: If you wish to see the "Homing in progress" signal, select C4012/007 = 16 (Homing in progress) and confirm with Ok.



To reconfigure other status bits, proceed as described in Step 1.



Save the modified settings to non-volatile memory with C0003/000 = 1.

End



TIP!

Regardless of the operating mode you select via C4000/000, the application status word will be displayed via code C4150/000.

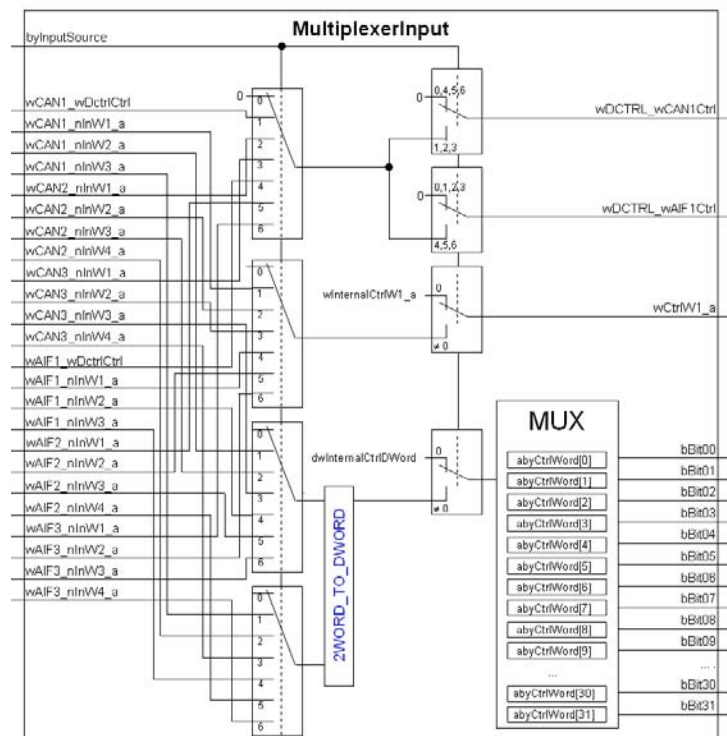
6.6 Description of the function blocks

6.6.1 MultiplexerInput function block

The *MultiplexerInput* function block can be used to select a data interface to control the output bits. The following sources can be selected via the *byInputSource* input byte:

- Internal source in the form of control words (3 process data items: words 1 ... 3)
- CAN1 (4 process data items: words 0 ... 3)
- CAN2 (4 process data items: words 0 ... 3)
- CAN3 (4 process data items: words 0 ... 3)
- AIF1 (4 process data items: words 0 ... 3)
- AIF2 (4 process data items: words 0 ... 3)
- AIF3 (4 process data items: words 0 ... 3)

Data words 2 and 3 are interpreted in bits and can be mapped to the output signals (*bBit00* ... *bBit31*) as required using the internal multiplexer. An element of the internal index array *abyCtrlWord[0..31]* is available for each of the output signals. This array is used to define, for each output bit (*bBit00* ... *bBit31*), the source in words 2 and 3 from which the information for this output signal has been taken (multiplexer).



Task information

Can be called in:	☒ Cyclic task	☒ Time-controlled task (INTERVAL)	☒ Event-controlled task (EVENT)	☒ Interrupt task
-------------------	---	---	---	--

Variable name

Inputs (Variable type: VAR_INPUT)		
Name	Data type	Meaning
byInputSource	BYTE	Selection of control source for application:
		0 Control code
		1 CAN1
		2 CAN2
		3 CAN3
		4 AIF1
		5 AIF2
		6 AIF3
wCAN1_wDctrlCtrl	WORD	Input for system variable <i>CAN1_wDctrlCtrl</i>
wCAN1_nlnW1_a	WORD	Input for system variable <i>CAN1_nlnW1_a</i>
wCAN1_nlnW2_a	WORD	Input for system variable <i>CAN1_nlnW2_a</i>
wCAN1_nlnW3_a	WORD	Input for system variable <i>CAN1_nlnW3_a</i>
wCAN2_nlnW1_a	WORD	Input for system variable <i>CAN2_nlnW1_a</i>
wCAN2_nlnW2_a	WORD	Input for system variable <i>CAN2_nlnW2_a</i>
wCAN2_nlnW3_a	WORD	Input for system variable <i>CAN2_nlnW3_a</i>
wCAN2_nlnW4_a	WORD	Input for system variable <i>CAN2_nlnW4_a</i>
wCAN3_nlnW1_a	WORD	Input for system variable <i>CAN3_nlnW1_a</i>
wCAN3_nlnW2_a	WORD	Input for system variable <i>CAN3_nlnW2_a</i>
wCAN3_nlnW3_a	WORD	Input for system variable <i>CAN3_nlnW3_a</i>
wCAN3_nlnW4_a	WORD	Input for system variable <i>CAN3_nlnW4_a</i>
wAIF1_wDctrlCtrl	WORD	Input for system variable <i>AIF1_wDctrlCtrl</i>
wAIF1_nlnW1_a	WORD	Input for system variable <i>AIF1_nlnW1_a</i>
wAIF1_nlnW2_a	WORD	Input for system variable <i>AIF1_nlnW2_a</i>
wAIF1_nlnW3_a	WORD	Input for system variable <i>AIF1_nlnW3_a</i>
wAIF2_nlnW1_a	WORD	Input for system variable <i>AIF2_nlnW1_a</i>
wAIF2_nlnW2_a	WORD	Input for system variable <i>AIF2_nlnW2_a</i>
wAIF2_nlnW3_a	WORD	Input for system variable <i>AIF2_nlnW3_a</i>
wAIF2_nlnW4_a	WORD	Input for system variable <i>AIF2_nlnW4_a</i>
wAIF3_nlnW1_a	WORD	Input for system variable <i>AIF3_nlnW1_a</i>
wAIF3_nlnW2_a	WORD	Input for system variable <i>AIF3_nlnW2_a</i>
wAIF3_nlnW3_a	WORD	Input for system variable <i>AIF3_nlnW3_a</i>
wAIF3_nlnW4_a	WORD	Input for system variable <i>AIF3_nlnW4_a</i>

Outputs (Variable type: VAR_OUTPUT)		
Name	Data type	Meaning
wDCTRL_wCAN1Ctrl	WORD	Output of control word for system variable <i>DCTRL_wCAN1Ctrl</i>
wDCTRL_wAIF1Ctrl	WORD	Output of control word for system variable <i>DCTRL_wAIF1Ctrl</i>
wCtrlW1_a	INT	Output of the first data word (derived from the <i>wInternalCtrlW1_a</i> , <i>wCAN1_nlnW1_a</i> , <i>wCAN2_nlnW2_a</i> , <i>wCAN3_nlnW2_a</i> , <i>wAIF1_nlnW1_a</i> , <i>wAIF2_nlnW2_a</i> or <i>wAIF3_nlnW2_a</i> variable in accordance with the value at <i>byInputSource</i>)

Cross Cutter Appendix

Outputs (Variable type: VAR_OUTPUT)		
Name	Data type	Meaning
bBit00 bBit01 bBit02 bBit03 ... bBit30 bBit31	BOOL	Control bits 0 to 31: The individual bits are generated from the internal control double word <i>dwCtrlWord</i> . Any bit from the control double word (<i>dwCtrlWord.0</i> ... <i>dwCtrlWord.31</i>) can be assigned to each output bit <i>bBit00</i> , <i>bBit01</i> ... <i>bBit31</i> . The internal index variable <i>abyCtrlWord[i]</i> (<i>i</i> = 0 ... 31) is used for assignment, whereby the index <i>i</i> addresses the required output bit <i>bBit00</i> , <i>bBit01</i> ... <i>bBit31</i> and the value of <i>abyCtrlWord[i]</i> the required source bit in the control double word (<i>dwCtrlWord.0</i> ... <i>dwCtrlWord.31</i>). The permissible value range for <i>abyCtrlWord[i]</i> is 0 ... 31.

Internal (Variable type: VAR)		
Name	Data type	Meaning
<i>wInternalCtrlW1_a</i>	INT	Internal variable for the entry of the data word on the output signal <i>wCtrlW1_a</i> : The value is output at the output signal <i>wCtrlW1_a</i> if internal control codes (<i>byInputSource</i> = 0) have been preselected as the signal source.
<i>dwInternalCtrlDWord</i>	DWORD	Internal variable for the entry of the control double word: The control double word is linked to the output signals <i>bBit00</i> , <i>bBit01</i> ... <i>bBit31</i> via the multiplexer, if internal control codes (<i>byInputSource</i> = 0) have been preselected as the signal source.
<i>abyCtrlWord</i>	ARRAY [0..31] OF BYTE	Index variables for linking the internal bits in the control double word to the output bits <i>bBit00</i> , <i>bBit01</i> ... <i>bBit31</i> Example: Index variable <i>abyCtrlWord[2]</i> defines the signal source selected for output bit <i>bBit02</i> . Setting <i>abyCtrlWord[2]</i> = 2 selects the third bit in the control double word (<i>dwCtrlWord.2</i>) as the signal source for output <i>bBit02</i> .
<i>dwCtrlWord</i>	DWORD	Internal mapping of control bits 0 to 31: This internal variable is mapped via the corresponding input signals in accordance with the signal source preselected via the <i>byInputSource</i> input.

(* selection of control interface via data bus system/control codes and multiplexing of input signals *)

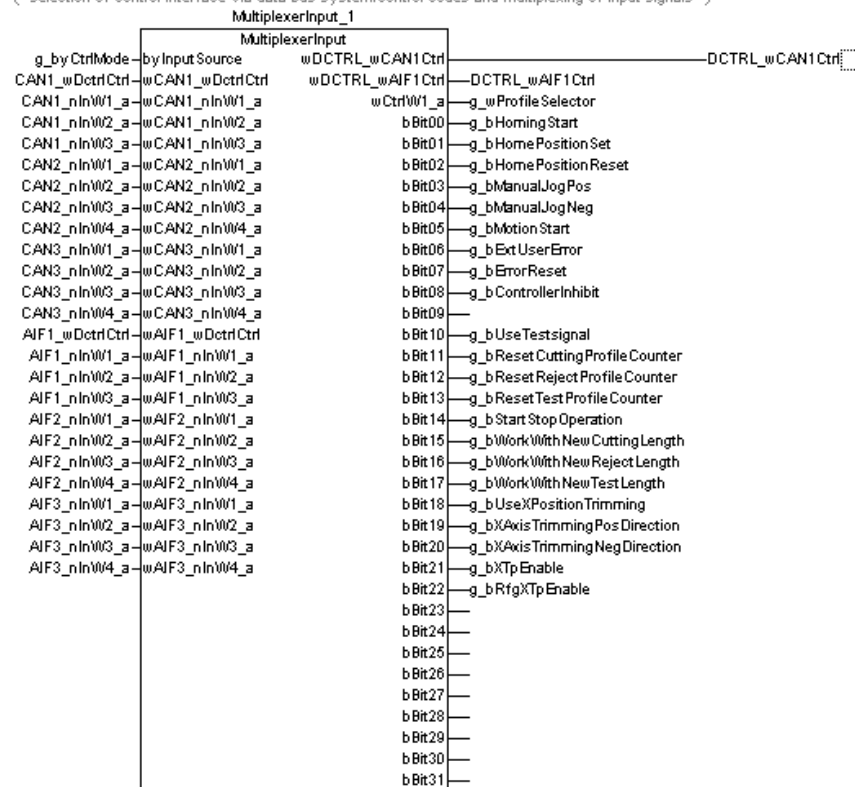


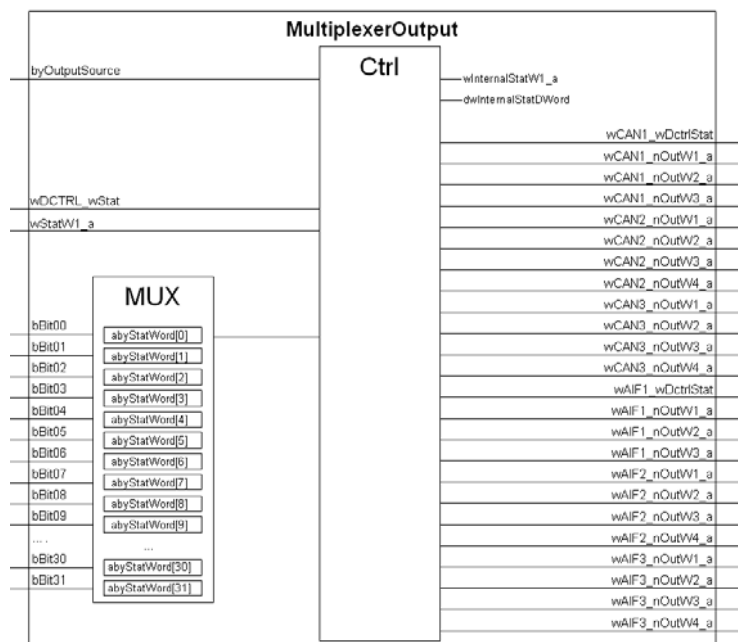
Figure45: Example wiring of the MultiplexerInput FB in the prepared solution

6.6.2 MultiplexerOutput function block

The *MultiplexerOutput* function block can be used to select a data interface to output status information. The following data interfaces can be selected via the *byOutputSource* input byte:

- Internal source in the form of status words (3 process data items: words 1 ... 3)
- CAN1 (4 process data items: words 0 ... 3)
- CAN2 (4 process data items: words 0 ... 3)
- CAN3 (4 process data items: words 0 ... 3)
- AIF1 (4 process data items: words 0 ... 3)
- AIF2 (4 process data items: words 0 ... 3)
- AIF3 (4 process data items: words 0 ... 3)

Data words 2 and 3 are interpreted in bits and can be mapped to the input signals (*bBit00* ... *bBit31*) as required using the internal multiplexer. An element of the internal index array *abyStatWord[0..31]* is available for each of the output signals. This array is used to define, for each bit in this status double word, the source (*bBit00* ... *bBit31*) from which the information for this status double word has been taken (multiplexer).



Task information

Can be called in:	☒ Cyclic task	☒ Time-controlled task (INTERVAL)	☒ Event-controlled task (EVENT)	☒ Interrupt task
-------------------	---	---	---	--

Variable name

Inputs (Variable type: VAR_INPUT)			
Name	Data type	Meaning	
byOutputSource	BYTE	Selection of the target interface for the application's status information:	
		0	Status code
		1	CAN1
		2	CAN2
		3	CAN3
		4	AIF1
		5	AIF2
		6	AIF3
wDCTRL_wStat	WORD	Input for the system status word (system variable <i>DCTRL_wStat</i>)	
wStatW1_a	INT	Free entry of the first data word (this value is then output at the <i>wInternalStatW1_a</i> , <i>wCAN1_nOutW1_a</i> , <i>wCAN2_nOutW2_a</i> , <i>wCAN3_nOutW2_a</i> , <i>wAIF1_nOutW1_a</i> , <i>wAIF2_nOutW2_a</i> or <i>wAIF3_nOutW2_a</i> variable in accordance with the value at <i>byOutputSource</i>)	
bBit00 bBit01 bBit02 bBit03 ... bBit30 bBit31	BOOL	Bit-by-bit entry of status information (bits 0 to 31): The internal status double word <i>dwStatWord</i> is generated from the individual bits. Each input bit <i>bBit00</i> , <i>bBit01</i> ... <i>bBit31</i> can be assigned to any bit in the status double word (<i>dwStatWord.0</i> ... <i>dwStatWord.31</i>). The internal index variable <i>abyStatWord[i]</i> (<i>i</i> = 0 ... 31) is used for assignment, whereby the index <i>i</i> addresses the required bit in the status double word (<i>dwStatWord.0</i> ... <i>dwStatWord.31</i>) and the value of <i>abyStatWord[i]</i> defines the required input bit (<i>bBit00</i> , <i>bBit01</i> ... <i>bBit31</i>) for this output bit. The permissible value range for <i>abyStatWord[i]</i> is 0 ... 31.	

Outputs (Variable type: VAR_OUTPUT)		
Name	Data type	Meaning
wCAN1_wDctrlStat	WORD	Output for system variable <i>CAN1_wDctrlStat</i>
wCAN1_nOutW1_a	WORD	Output for system variable <i>CAN1_nOutW1_a</i>
wCAN1_nOutW2_a	WORD	Output for system variable <i>CAN1_nOutW2_a</i>
wCAN1_nOutW3_a	WORD	Output for system variable <i>CAN1_nOutW3_a</i>
wCAN2_nOutW1_a	WORD	Output for system variable <i>CAN2_nOutW1_a</i>
wCAN2_nOutW2_a	WORD	Output for system variable <i>CAN2_nOutW2_a</i>
wCAN2_nOutW3_a	WORD	Output for system variable <i>CAN2_nOutW3_a</i>
wCAN2_nOutW4_a	WORD	Output for system variable <i>CAN2_nOutW4_a</i>
wCAN3_nOutW1_a	WORD	Output for system variable <i>CAN3_nOutW1_a</i>
wCAN3_nOutW2_a	WORD	Output for system variable <i>CAN3_nOutW2_a</i>
wCAN3_nOutW3_a	WORD	Output for system variable <i>CAN3_nOutW3_a</i>
wCAN3_nOutW4_a	WORD	Output for system variable <i>CAN3_nOutW4_a</i>
wAIF1_wDctrlStat	WORD	Output for system variable <i>AIF1_wDctrlStat</i>
wAIF1_nOutW1_a	WORD	Output for system variable <i>AIF1_nOutW1_a</i>
wAIF1_nOutW2_a	WORD	Output for system variable <i>AIF1_nOutW2_a</i>
wAIF1_nOutW3_a	WORD	Output for system variable <i>AIF1_nOutW3_a</i>
wAIF2_nOutW1_a	WORD	Output for system variable <i>AIF2_nOutW1_a</i>
wAIF2_nOutW2_a	WORD	Output for system variable <i>AIF2_nOutW2_a</i>
wAIF2_nOutW3_a	WORD	Output for system variable <i>AIF2_nOutW3_a</i>
wAIF2_nOutW4_a	WORD	Output for system variable <i>AIF2_nOutW4_a</i>
wAIF3_nOutW1_a	WORD	Output for system variable <i>AIF3_nOutW1_a</i>
wAIF3_nOutW2_a	WORD	Output for system variable <i>AIF3_nOutW2_a</i>
wAIF3_nOutW3_a	WORD	Output for system variable <i>AIF3_nOutW3_a</i>

Cross Cutter Appendix

Outputs (Variable type: VAR_OUTPUT)		
wAIF3_nOutW4_a	WORD	Output for system variable AIF3_nOutW4_a

Internal (Variable type: VAR)		
Name	Data type	Meaning
wInternalStatW1_a	INT	Internal status word for displaying the current data word (input signal wCtrlW1_a): If byOutputSource = 0, the value at input wCtrlW1_a is assigned to internal status word wInternalStatW1_a.
dwInternalStatDWord	DWORD	Internal variable for displaying the status double word: The status double word is generated from input signals bBit00, bBit01 ... bBit31 via the multiplexer if byOutputSource = 0 is preselected.
abyStatWord	ARRAY [0..31] OF BYTE	Index variables for linking the input bits bBit00, bBit01 ... bBit31 to the individual bits in status double word dwStatWord Example: Index variable abyStatWord[2] defines the input bit selected for the second data bit of the status double word. Setting abyStatWord[2] = 2 selects the second data bit of the status double word (dwStatWord.2) as the signal source for the third input bit bBit02.
dwStatWord	DWORD	Internal mapping of status bits 0 to 31 as status double word

(* selection of status interface via data bus system/status codes and multiplexing of output signals *)

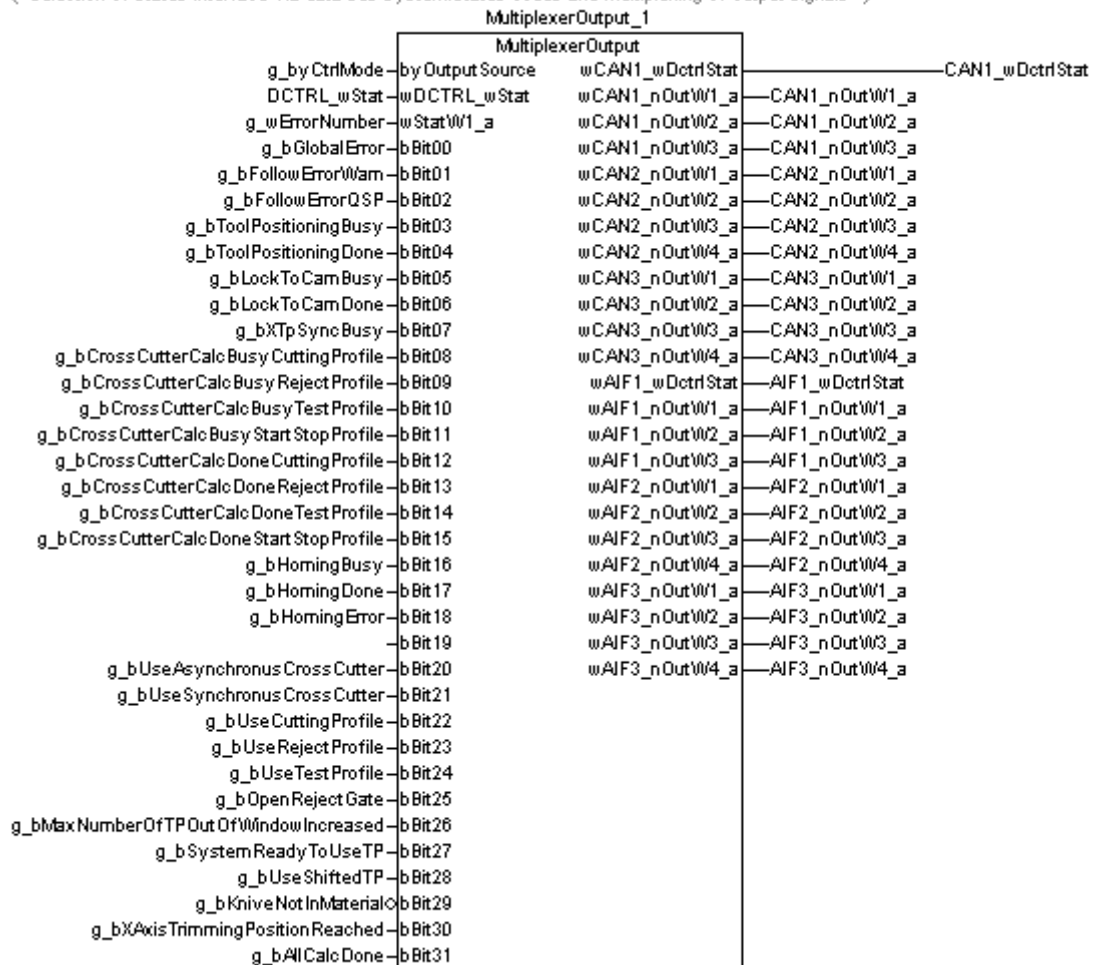


Figure46: Example wiring of the MultiplexerOutput FB in the prepared solution

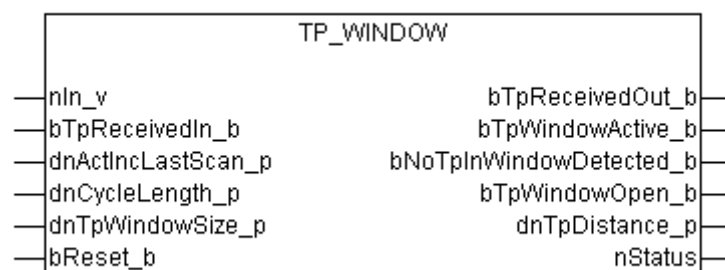
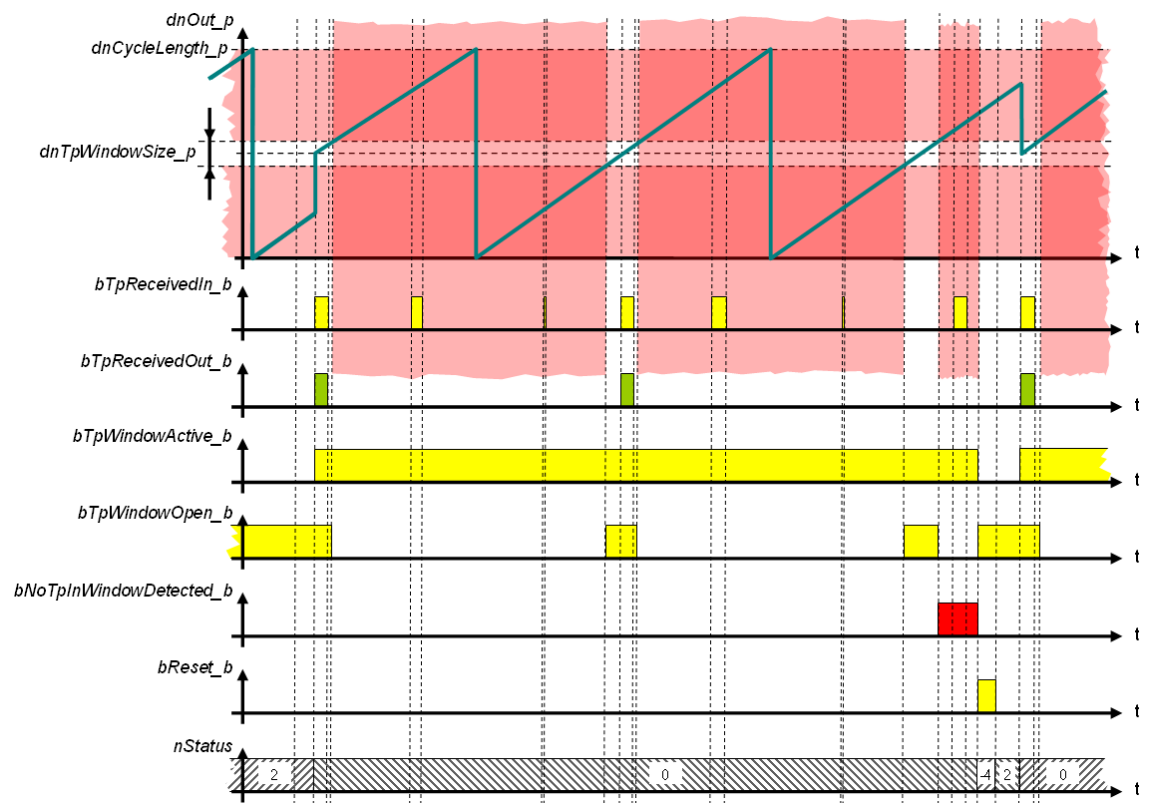
6.6.3 TP_Window – Masking out mark pulses

If the setpoint distance (*dnCycleLength_p*) between two valid print marks is known, the TP_Window function block can be used to mask out invalid print marks between two valid print marks. For this purpose, a tolerance window *dnTpWindowSize_p* can be defined in which the valid print-mark pulse must be located. Print-mark pulses are evaluated using the last-scan value on a touch-probe-specific basis. Every time a new valid print-mark pulse is detected, the centre of the tolerance window is adjusted in line with the print-mark pulse detected, in order to prevent the mark pulse "running away". The assignment between the current print-mark pulse and the tolerance window is deleted by setting control input *bReset_b* = TRUE. The next print-mark pulse in the sequence defines the new tolerance window position and also serves as the new home position for subsequent print-mark signals.

The function block has a comprehensive selection of status outputs:

- A valid print-mark pulse is output via output *bTpReceivedOut_b* and can be processed in subsequent function blocks (e.g. the print-mark register, *TP_Register* function block). The associated last-scan value corresponds to the value at input *dnActInLastScanIn_p*.
- If the mask-out function has been activated for mark signals, the status of output bit *bTpWindowActive_b* will change from FALSE to TRUE.
- Output *bTpWindowOpen_b* will be TRUE if the tolerance window is open: When *bTpWindowOpen_b* = TRUE, a print-mark signal at input *bTpReceivedIn_b* will be switched through to output *bTpReceivedOut_b*.
- If no print-mark pulses are detected at input *bTpReceivedIn_b* during the tolerance window, when the print-mark window closes (*bTpWindowOpen_b* = TRUE => FALSE) status output *bNoTpInWindowDetected_b* will be set to TRUE until the tolerance window opens again (*bTpWindowOpen_b* = FALSE => TRUE).
- Output signal *dnTpDistance_p* outputs the distance between the last two valid mark signals on a touch-probe-specific basis.

The temporal characteristics show, for example, the various states of *TP_Window* function blocks at constant material speed ($n/n_v = \text{const.}$):



Task information

Can be called in:	☐ Cyclic task	☒ Time-controlled task (INTERVAL)	☐ Event-controlled task (EVENT)	☐ Interrupt task
-------------------	--------------------------------------	---	--	---

Cross Cutter

Appendix

Variable name

Inputs (Variable type: VAR_INPUT)			
Name	Data type	Value/Meaning	
nIn_v	INT	Material speed in [incr./ms]: This provides the basis for the opening/closing of the tolerance window (16384 = 15000[rpm]). When the print-mark pulses are evaluated, system variable <i>DFIN_nIn_v</i> is typically evaluated at this input.	
bTpReceivedIn_b	BOOL	Input bit for the print-mark signal: A FALSE => TRUE edge tells the <i>Tp_Window</i> function block that a print-mark signal has been detected. When the print-mark pulses are evaluated, system variable <i>DFIN_bTPReceived_b</i> is typically evaluated at this input.	
dnActInLastScan_p	DINT	Last-scan information: This value is made available by the touch-probe interface and indicates the material feed in [incr.] between the point at which the print-mark signal occurred and the start of the calling task. When the print-mark pulses are evaluated, system variable <i>DFIN_dnInLastScan_p</i> is typically evaluated at this input.	
dnCycleLength_p	DINT	Entry indicating the setpoint distance between two valid print marks in [incr.]: On the basis of the setpoint distance, the <i>TP_Window</i> function block can open the tolerance window in advance, before the next valid print mark (touch probe) is detected at input <i>bTpReceivedIn_b</i> (see also model connection).	
dnTpWindowSize_p	DINT	Width of the tolerance window in [incr.]: Only enter positive values between 2 and ... (<i>dnCycleLength_p</i> – 1) at this input. You will find an example connection in the next section. <u>Note:</u> If the width of the tolerance window is outside the permissible range, status output <i>nStatus</i> will be set to -2. Furthermore, all incoming print-mark pulses at input <i>bTpReceivedIn_b</i> will be connected to output <i>bTpReceivedOut_b</i> . Status signal <i>bTpWindowOpen_b</i> is set statically to TRUE, status signal <i>bTpWindowActive_b</i> is set statically to FALSE.	
bReset_b	BOOL	Input to delete the position of the tolerance window:	
		FALSE	The first print-mark pulse detected at input <i>bTpReceivedIn_b</i> defines the centre of the tolerance window and sets its position. All subsequent print-mark signals at <i>bTpReceivedIn_b</i> will be checked in respect of their position inside the tolerance window and, if they are located inside the tolerance window, output to output <i>bTpReceivedOut_b</i> .
		TRUE	Resets the tolerance window: • All print-mark signals detected are blocked (output <i>bTpReceivedOut_b</i> is set statically to FALSE). • The status is set to <i>nStatus</i> = -4. • Status signal <i>bTpWindowOpen_b</i> is set statically to TRUE. • Status signal <i>bTpWindowActive_b</i> is set statically to FALSE.

Cross Cutter Appendix

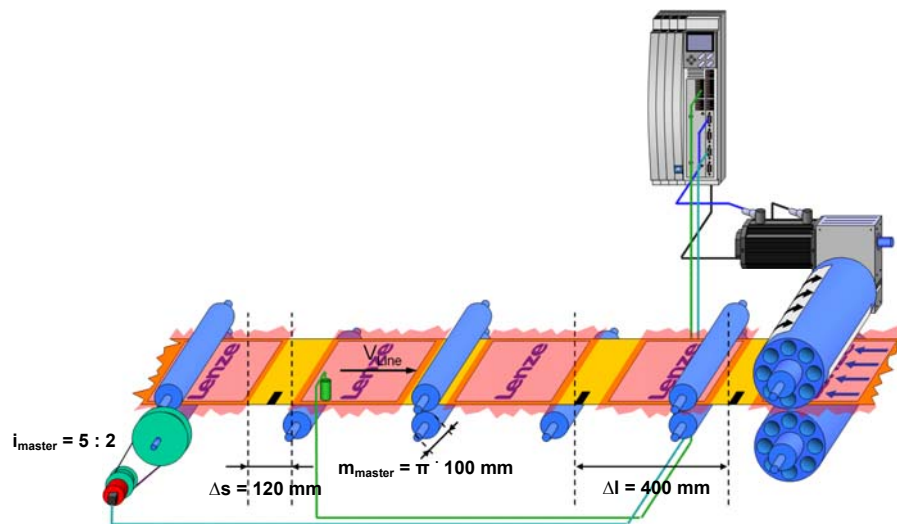
Outputs (Variable type: VAR_OUTPUT)											
Name	Data type	Value/Meaning									
bTpReceivedOut_b	BOOL	Display of a valid print-mark pulse (print-mark pulse inside the tolerance window): A FALSE-TRUE edge indicates a valid print-mark signal. <u>Note:</u> The associated last-scan value for a valid print mark corresponds to the value at input <i>dnActInLastScan_p</i> . This makes it possible to evaluate the print-mark signal on a touch-probe-specific basis.									
bTpWindowActive_b	BOOL	Indicates whether a tolerance window has been defined by an initial print-mark pulse at input <i>bTpReceivedIn_b</i> : <table><tr><td>FALSE</td><td>The tolerance window has not yet been defined (the position of the tolerance window will be defined by the first print-mark pulse detected at input <i>bTpReceivedIn_b</i>).</td></tr><tr><td>TRUE</td><td>The position of the tolerance window is known – all subsequent print-mark pulses are checked in respect of their position inside the tolerance window and output to <i>bTpReceivedOut_b</i> if applicable.</td></tr></table>		FALSE	The tolerance window has not yet been defined (the position of the tolerance window will be defined by the first print-mark pulse detected at input <i>bTpReceivedIn_b</i>).	TRUE	The position of the tolerance window is known – all subsequent print-mark pulses are checked in respect of their position inside the tolerance window and output to <i>bTpReceivedOut_b</i> if applicable.				
FALSE	The tolerance window has not yet been defined (the position of the tolerance window will be defined by the first print-mark pulse detected at input <i>bTpReceivedIn_b</i>).										
TRUE	The position of the tolerance window is known – all subsequent print-mark pulses are checked in respect of their position inside the tolerance window and output to <i>bTpReceivedOut_b</i> if applicable.										
bNoTpInWindowDetected_b	BOOL	Display bit for error pulses: <table><tr><td>FALSE => TRUE</td><td>No print-mark pulses have been detected in the tolerance window. The state of the output will not change until the tolerance window opens again.</td></tr></table>		FALSE => TRUE	No print-mark pulses have been detected in the tolerance window. The state of the output will not change until the tolerance window opens again.						
FALSE => TRUE	No print-mark pulses have been detected in the tolerance window. The state of the output will not change until the tolerance window opens again.										
bTpWindowOpen_b	BOOL	Display bit for the open tolerance window: <table><tr><td>FALSE</td><td>The tolerance window is closed: A print-mark pulse at input <i>bTpReceivedIn_b</i> will be blocked and not output to output <i>bTpReceivedOut_b</i>.</td></tr><tr><td>TRUE</td><td>The tolerance window is open: A print-mark pulse at input <i>bTpReceivedIn_b</i> will be output to output <i>bTpReceivedOut_b</i>. <u>Note:</u> The output will be TRUE for at least one cycle, even if the tolerance window opens and closes again within a task cycle.</td></tr></table>		FALSE	The tolerance window is closed: A print-mark pulse at input <i>bTpReceivedIn_b</i> will be blocked and not output to output <i>bTpReceivedOut_b</i> .	TRUE	The tolerance window is open: A print-mark pulse at input <i>bTpReceivedIn_b</i> will be output to output <i>bTpReceivedOut_b</i> . <u>Note:</u> The output will be TRUE for at least one cycle, even if the tolerance window opens and closes again within a task cycle.				
FALSE	The tolerance window is closed: A print-mark pulse at input <i>bTpReceivedIn_b</i> will be blocked and not output to output <i>bTpReceivedOut_b</i> .										
TRUE	The tolerance window is open: A print-mark pulse at input <i>bTpReceivedIn_b</i> will be output to output <i>bTpReceivedOut_b</i> . <u>Note:</u> The output will be TRUE for at least one cycle, even if the tolerance window opens and closes again within a task cycle.										
dnTpDistance_p	DINT	Distance between two valid print-mark pulses in [incr.]: This output can be used for control and monitoring, e.g. in order to: - Identify inaccuracies or diffusions affecting the application of print marks. - Set the print-mark distance as a reference value for the setpoint distance (input signal <i>dnCycleLength_p</i>).									
nStatus	INT	Status of the <i>Tp_Window</i> function block: <table><tr><td>2</td><td>The first print mark is expected at input <i>bTpReceivedIn_b</i> to define the position of the tolerance window.</td></tr><tr><td>0</td><td>Normal operation: Incoming print-mark pulses at input <i>bTpReceivedIn_b</i> are checked in respect of their position inside the tolerance window and output to output <i>bTpReceivedOut_b</i> if applicable.</td></tr><tr><td>-2</td><td>The width of the tolerance window (input <i>dnTpWindowSize_p</i>) is outside the permitted range 2 ... (<i>dnCycleLength_p</i> – 1).</td></tr><tr><td>-4</td><td>The function block is reset by input <i>bReset_b</i> = TRUE.</td></tr></table>		2	The first print mark is expected at input <i>bTpReceivedIn_b</i> to define the position of the tolerance window.	0	Normal operation: Incoming print-mark pulses at input <i>bTpReceivedIn_b</i> are checked in respect of their position inside the tolerance window and output to output <i>bTpReceivedOut_b</i> if applicable.	-2	The width of the tolerance window (input <i>dnTpWindowSize_p</i>) is outside the permitted range 2 ... (<i>dnCycleLength_p</i> – 1).	-4	The function block is reset by input <i>bReset_b</i> = TRUE.
2	The first print mark is expected at input <i>bTpReceivedIn_b</i> to define the position of the tolerance window.										
0	Normal operation: Incoming print-mark pulses at input <i>bTpReceivedIn_b</i> are checked in respect of their position inside the tolerance window and output to output <i>bTpReceivedOut_b</i> if applicable.										
-2	The width of the tolerance window (input <i>dnTpWindowSize_p</i>) is outside the permitted range 2 ... (<i>dnCycleLength_p</i> – 1).										
-4	The function block is reset by input <i>bReset_b</i> = TRUE.										

Internal (Variable type: VAR)		
Name	Data type	Meaning
dnTpPosition_p	DINT	Position of the tolerance window: The variable indicates the centre of the tolerance window in the cycle. The value is calculated by dividing <i>dnCycleLength_p</i> by 2.
dnLowerWindowLimit_p dnUpperWindowLimit_p	DINT	Upper/lower limit of the tolerance window: The variables are calculated as follows: $\begin{aligned} dnLowerWindowLimit_p &= dnTpPosition_p - dnTpWindowSize_pl/2 \\ dnUpperWindowLimit_p &= dnTpPosition_p + dnTpWindowSize_pl/2. \end{aligned}$
dnOut_p	DINT	Control variable: This position value is derived by integrating the speed input <i>nIn_v</i> and runs in the value range 0 ... (<i>dnCycleLength_p</i> – 1), thereby mapping a cycle (setpoint distance between two valid print marks). It combines with the <i>dnLowerWindowLimit_p</i> and <i>dnUpperWindowLimit_p</i> variables to generate the <i>bTpWindowOpen_b</i> output variable.

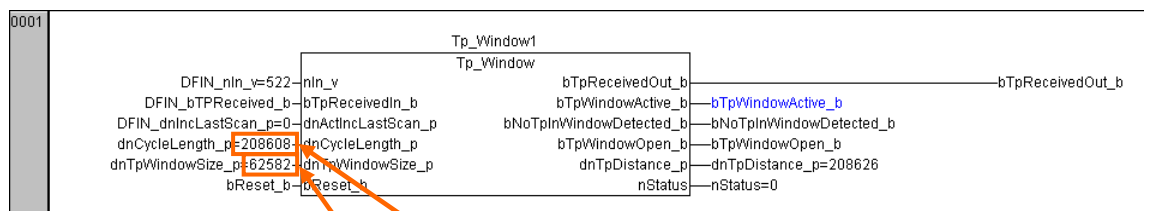
Connection example

The following structure will be used to provide a practical illustration of the application of the *Tp_Window* function block:

A web with print marks at distances of $\Delta l = 400$ mm is moving forwards at approx. 60 m/min. The feed motion is detected via an incremental encoder driven via a belt ratio $i_{\text{master}} = 5 : 2$ by one of the feed rolls (diameter $\varnothing_{\text{master}} = 100$ mm). A valid print-mark tolerance window of $\Delta s = 100$ mm wide is to be set in order to prevent any additional print image on the web affecting the mark sensor.



The required behaviour can be achieved in the following connection using the *Tp_Window* function block:



The incremental value for the setpoint print-mark distance (input signal *dnCycleLength_p*) is calculated using the following formula:

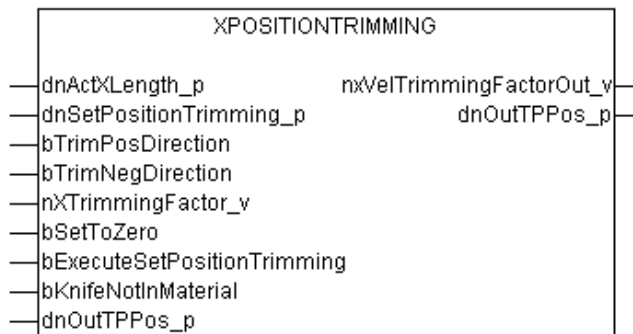
$$\begin{aligned} dnCycleLength_p &= 65536[\text{incr./rev}] \cdot i_{\text{Master}} \cdot \frac{\Delta l}{m_{\text{Master}}} = \\ &= 65536[\text{incr./rev}] \cdot \frac{5}{2} \cdot \frac{400[\text{mm}]}{\pi \cdot 100[\text{mm/rev}]} = 208607,567... \end{aligned}$$

The value for the width of the window (input signal *dnTpWindowSize_p*) is calculated using the following formula:

$$\begin{aligned} dnTpWindowSize_p &= 65536[\text{incr./rev}] \cdot i_{\text{Master}} \cdot \frac{\Delta s}{m_{\text{Master}}} = \\ &= 65536[\text{incr./rev}] \cdot \frac{5}{2} \cdot \frac{120[\text{mm}]}{\pi \cdot 100[\text{mm/rev}]} = 62582,270... \end{aligned}$$

6.6.4 XPositionTrimming function block

The XPositionTrimming function block can be used to move the x position in a negative or positive direction by means of inching. Furthermore, a position value can be defined for an absolute positioning operation to which the function block is to move the x position. To move the x position, the function block generates a speed value at the output to move the x axis directly. A print-mark-sensor distance is also defined at the output by the block.



Task information

Can be called in:	☐ Cyclic task	☒ Time-controlled task (INTERVAL)	☐ Event-controlled task (EVENT)	☐ Interrupt task
-------------------	--------------------------------------	---	--	---

Variable names

Inputs (Variable type: VAR_INPUT)		
Name	Data type	Value/Meaning
dnActXLength_p	DINT	Cycle of the x axis. Connect this input to output <i>dnActXLength_p</i> of the L_CamData FB.
dnSetPositionTrimming_p	DINT	Setpoint for absolute positioning. Connect this variable to the position value you wish to use for correction. As soon as this value changes, absolute positioning will be performed to the specified value.
bTrimPosDirection	BOOL	Setting this variable to TRUE will cause a positive trimming speed (<i>nXTrimmingFactor_v</i>) to be output to output <i>nxVelTrimmingFactorOut_v</i> and a corresponding position value to be output to output <i>dnOutTpPos_p</i> .
bTrimNegDirection	BOOL	Setting this value to TRUE will cause a negative trimming speed to be output to output <i>nxVelTrimmingFactorOut_v</i> and a corresponding position value to be output to output <i>dnOutTpPos_p</i> .
nXTrimmingFactor_v	INT	Entry indicating the trimming speed for trimming in a positive and negative direction (<i>bTrimPosDirection</i> and <i>bTrimNegDirection</i>)
bSetToZero	BOOL	Reset Variable <i>dnOutTpPos_p</i> to zero.
		TRUE Set the value of variable <i>dnOutTpPos_p</i> to zero.
		FALSE Value of variable <i>dnOutTpPos_p</i> will remain its value.
bExecuteSetPositionTrimming	BOOL	Perform absolute positioning.
		TRUE Perform absolute positioning in accordance with the value of variable <i>dnSetPositionTrimming_p</i> .
		FALSE Abort absolute positioning.

Cross Cutter Appendix

Inputs (Variable type: VAR_INPUT)		
Name	Data type	Value/Meaning
bKnifeNotInMaterial	BOOL	Indicates that the cross cutter knife is not in Material. Connect this value to variable <i>g_bKnifeNotInMaterial</i> .
		TRUE The cross cutter knife is not in material.
		FALSE The cross cutter knife is in material.

Outputs (Variable type: VAR_OUTPUT)		
Name	Data type	Value/Meaning
nxVelTrimmingFactorOut_v	INT	Speed signal which can be used to perform manual x trimming. This value must be added to the master speed of the x axis.

Inputs/Outputs (Variable type: VAR_IN_OUT)		
Name	Data type	Value/Meaning
dnOutTpPos_p	DINT	Position signal used for manual x trimming. Connect this value to variable <i>dnTpPos_p</i> of the <i>L_CamSetXAxisVelocity</i> function block. Note: Assign a retain variable to this input/output!

In order to be able to use the full functionality of the *XPositionTrimming* function block, you need to create 7 codes.

The specific variables used for this purpose are:

Variable	Value/Meaning			
Name	Data type	Value/Meaning		
xxx.L_RFG1.dnTir	DINT	Variable for acceleration time when using <i>bTrimPosDirection</i> and <i>bTrimNegDirection</i> . The acceleration and deceleration times relate to a change in the output value from 0 to 100% (100% = 16384).		
		Possible settings		
		Default	Selection	
		0.000 s	0.000	{0.001 s} 999.999
xxx.L_RFG1.dnTif	DINT	Variable for deceleration time when using <i>bTrimPosDirection</i> and <i>bTrimNegDirection</i> . The acceleration and deceleration times relate to a change in the output value from 0 to 100% (100% = 16384).		
		Possible settings		
		Default	Selection	
		0.000 s	0.000	{0.001 s} 999.999
xxx.L_CamPosCtrlLin1.adnTiPos[1]	ARRAY OF DINT	Variable for deceleration ramp to stop before reaching the set position (<i>bExecuteSetPositionTrimming</i> = FALSE) when using variables <i>dnSetPositionTrimming_p</i> and <i>bExecuteSetPositionTrimming</i> . The ramp assumes 4000 rpm.		
		Possible settings		
		Default	Selection	
		1.000 s	0.010	{0.001 s} 130.000

Cross Cutter Appendix

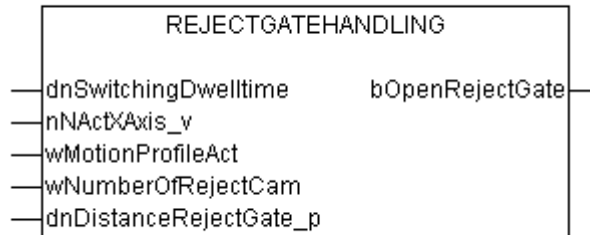
Variable	Value/Meaning			
Name	Data type	Value/Meaning		
xxx.L_CamPosCtrlLin1.adnTiPos[2]	ARRAY OF DINT	Variable for acceleration ramp for positioning when using variables dnSetPositionTrimming_p and bLoadSetPositionTrimming. The ramp assumes 4000 rpm.		
	Possible settings			
	Default	Selection		
	1.000 s	0.010	{0.001 s}	130.000
xxx.L_CamPosCtrlLin1.adnTiPos[3]	ARRAY OF DINT	Variable for deceleration ramp for positioning when using variables dnSetPositionTrimming_p and bLoadSetPositionTrimming. The ramp assumes 4000 rpm.		
	Possible settings			
	Default	Selection		
	1.000 s	0.010	{0.001 s}	130.000
xxx.L_CamPosCtrlLin1.anSpeed[1]	ARRAY OF INT	Variable for forward motion (positive speed) for positioning when using variables dnSetPositionTrimming_p and bLoadSetPositionTrimming.		
	Possible settings			
	Default	Selection		
	200 rpm	-15000	{1 rpm}	15000
xxx.L_CamPosCtrlLin1.anSpeed[2]	ARRAY OF INT	Variable for return motion (negative speed) for positioning when using variables dnSetPositionTrimming_p and bLoadSetPositionTrimming.		
	Possible settings			
	Default	Selection		
	200 rpm	-15000	{1 rpm}	15000

* xxx stands for the instance name of the block.

User-specific codes can be created starting at value C3000. 7 codes must be created for each function block call.

6.6.5 RejectGateHandling function block

The RejectGateHandling function block can be used to open a reject gate. A switching delay time is entered to compensate reject gate switching delays.



Task information

Can be called in:	☐ Cyclic task	☒ Time-controlled task (INTERVAL)	☐ Event-controlled task (EVENT)	☐ Interrupt task
-------------------	--------------------------------------	---	--	---

Variable names

Inputs (Variable type: VAR_INPUT)		
Name	Data type	Value/Meaning
dnSwitchingDwelltime	DINT	Entry indicating the switching delay time of the reject gate in ms
nNActXAxis_v	INT	Line speed of the x axis. The value can be positive or negative. Positive values are interpreted as clockwise rotation and negative values as counter-clockwise rotation.
wMotionProfileAct	WORD	Motion profile currently in use. Connect this input to variable <i>g_wMotionProfileAct</i> .
wNumberOfRejectCam	WORD	Number of reject cam.
dnDistanceRejectGate_p	DINT	Distance from reject gate to knife drum (curve zero of x axis) in increments (incr.).

Outputs (Variable type: VAR_OUTPUT)		
Name	Data type	Value/Meaning
bOpenRejectGate	BOOL	Control of reject gate. The output is activated dependent upon the line speed, the reject gate switching delay time and the distance from the reject gate to the knife drum if the motion profile currently selected is the reject cam.
		TRUE Reject gate open
		FALSE Reject gate closed

(*Implementation of the Reject Gate Handling*)

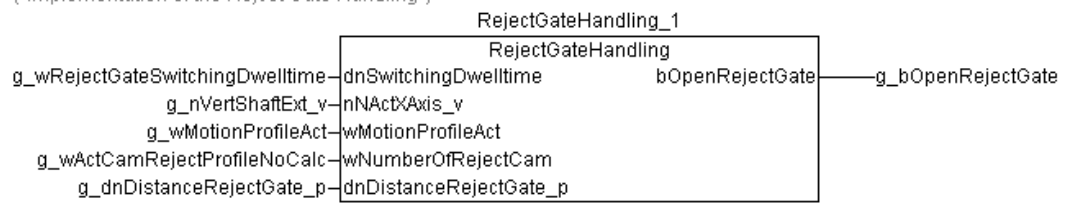
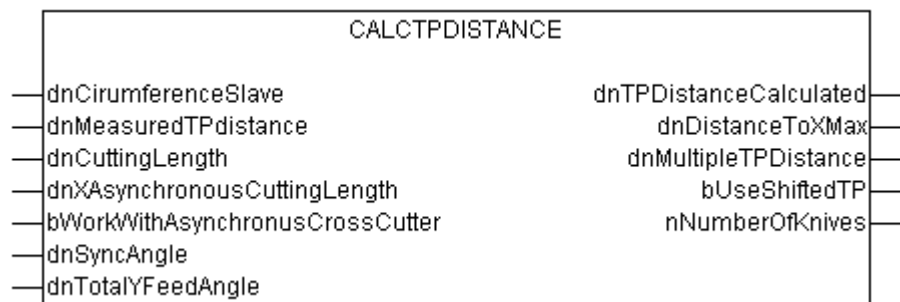


Figure47: Example wiring of the RejectGateHandling FB in the prepared solution

6.6.6 CalcTpDistance function block

The CalcTpDistance function block uses the cross-cutter application's input variables to calculate the following output values:

- The print-mark-sensor distance
- The distance from the 6 o'clock position of the knife to the x max. value of the profile
- The value indicated by the distance of the touch-probe sensor in integer multiples of the cutting length
- The number of knives mounted on the cross-cutter drum are also calculated.



Task information

Can be called in:	☒ Cyclic task	☒ Time-controlled task (INTERVAL)	☐ Event-controlled task (EVENT)	☐ Interrupt task
-------------------	---	---	--	---

Variable names

Inputs (Variable type: VAR_INPUT)		
Name	Data type	Value/Meaning
dnCircumferenceSlave	DINT	Cutting circle circumference of the knife drum in 0.0001[m_units]. <u>Note:</u> Do not enter the synchronous cutting length here (on knife drums with more than one knife, this may not be the same as the cutting circle circumference). You should always enter the entire circumference of the knife's cutting circle.
dnMeasuredTPdistance	DINT	Distance measured from TP sensor to 6 o'clock position on the knife drum. The value is entered in 0.0001[m_units].
dnCuttingLength	DINT	Cutting length of the material. The value is entered in 0.0001[m_units]. You can also link this input variable directly to output variable <i>dnActCuttingLength</i> in the <i>CrossCutterMotionProfileCalculation FB</i> .
dnXAsynchronousCuttingLength	DINT	Length of the profile in which the cross cutter moves asynchronously in relation to the material;. The value is entered in 0.0001[m_units]. You can also link this input variable directly to output variable <i>dnXAsynchronousLength</i> in the <i>CrossCutterMotionProfileCalculation FB</i> .
bWorkWithAsynchronousCrossCutter	BOOL	Change-over of calculation formula in accordance with the type of cross cutter being used.
		TRUE Using the calculation for the asynchronous cross cutter.
		FALSE Using the calculation for the synchronous cross cutter.
dnSyncAngle	DINT	Cutting angle for which the knife has to run in synchronism with the material (y_{sync} , the value is entered in 0.0001[s_units]).
dnTotalYFeedAngle	DINT	Y final value of the selected motion profile in 0.0001[s_units].

Cross Cutter Appendix

Outputs (Variable type: VAR_OUTPUT)		
Name	Data type	Value/Meaning
dnTPDistanceCalculated	DINT	Touch-probe-sensor position calculated. A modulo arithmetic operation is used to adjust this value by multiples of the cutting length so that a virtual touch-probe-sensor position can be calculated as close as possible to the knife. The value is scaled in 0.0001[m_units].
dnDistanceToXMax	DINT	Distance calculated from the 6 o'clock position of the knife to the x max. value of the profile. The value is scaled in 0.0001[m_units].
dnMultipleTPDistance	DINT	Calculated value indicating the touch-probe-sensor distance in integer multiples of the cutting length. The value is scaled in 0.0001[m_units].
bUseShiftedTP	BOOL	Status signal indicating whether the touch-probe-sensor position measured plus the 6 o'clock position of the knife calculated in relation to the x max. value of the profile (<i>dnDistanceToXMax</i>) permits touch-probe correction with or without touch-probe register control.
		TRUE The touch-probe-sensor distance measured plus <i>dnDistanceToXMax</i> is greater than the cutting length. Touch-probe correction with a touch-probe register is required.
		FALSE The touch-probe-sensor distance measured plus <i>dnDistanceToXMax</i> is less than the cutting length. Touch-probe correction without a touch-probe register is required.
nNumberOfKnives	INT	Number of knives on the cross-cutter drum

("Calculating the position of the TP sensor")

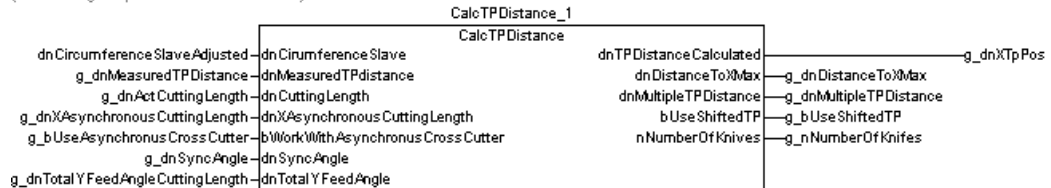
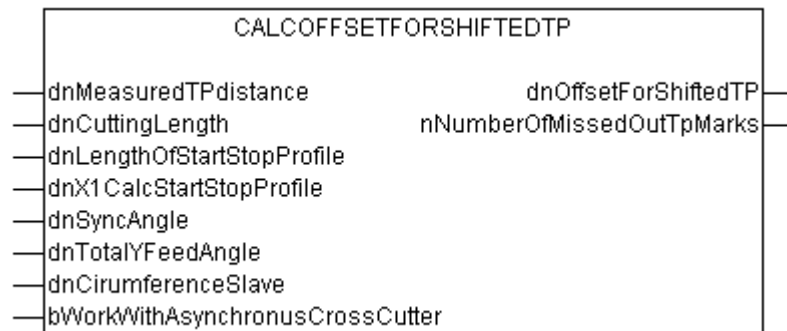


Figure48: Example wiring of the CalcTPDistance FB in the prepared solution

6.6.7 CalcOffsetForShiftedTP function block

The CalcOffsetForShiftedTP function block uses the cross-cutter application's input variables to calculate the offset value by which the first print-mark signal will be delayed to ensure that the first cut is made at a valid print mark. The block also indicates how many print marks will pass before the first cut is made.



Task information

Can be called in:	☒ Cyclic task	☒ Time-controlled task (INTERVAL)	☐ Event-controlled task (EVENT)	☐ Interrupt task
-------------------	---	---	--	---

Variable names

Inputs (Variable type: VAR_INPUT)		
Name	Data type	Value/Meaning
dnMeasuredTPdistance	DINT	Distance measured from TP sensor to 6 o'clock position on the knife drum. The value is entered in 0.0001[m_units].
dnCuttingLength	DINT	Cutting length of the material. The value is entered in 0.0001[m_units].
dnLengthOfStartStopProfile	DINT	Length of the start/stop motion profile.
dnX1CalcStartStopProfile	DINT	Position value for the starting point of the pause on the start/stop motion profile in 0.0001[m_units].
dnSyncAngle	DINT	Cutting angle for which the knife has to run in synchronism with the material (y_{sync} , the value is entered in 0.0001[s_units]).
dnTotalYFeedAngle	DINT	Y final value of the selected motion profile in 0.0001[s_units].
dnCircumferenceSlave	DINT	Cutting circle circumference of the knife drum in 0.0001[m_units]. Note: Do not enter the synchronous cutting length here (on knife drums with more than one knife, this may not be the same as the cutting circle circumference). You should always enter the entire circumference of the knife's cutting circle.
bWorkWithAsynchronousCrossCutter	BOOL	Change-over of calculation formula in accordance with the type of cross cutter being used.
		TRUE Using the calculation for the asynchronous cross cutter.
		FALSE Using the calculation for the synchronous cross cutter.

Cross Cutter Appendix

Outputs (Variable type: VAR_OUTPUT)		
Name	Data type	Value/Meaning
dnOffsetForShiftedTP	DINT	Calculated position value by which the first print-mark signal must be delayed in order for the first cut to be made at a print mark. The value is scaled in 0.0001[m_units].
nNumberOfMissedOutTpMarks	INT	Number of print marks which pass the cross-cutter knife drum before the first cut is made.

("Calculate the position value for shifting the TP signals for starting the cross cutter with printing marks")

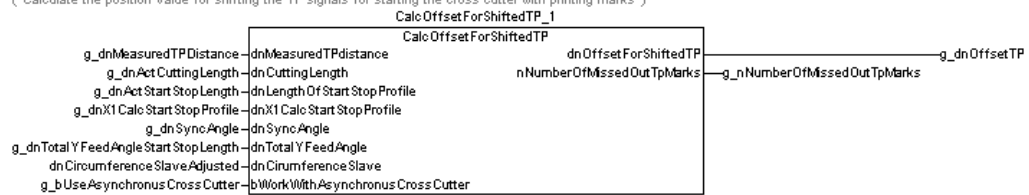
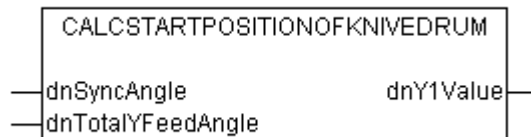


Figure49: Example wiring of the CalcOffsetForShiftedTP FB in the prepared solution

6.6.8 CalcStartPositionOfKnifeDrum function block

The CalcStartPositionOfKnifeDrum function block uses the cross-cutter application's input variables to calculate the starting position for the cross-cutter knife.



Task information

Can be called in:	☒ Cyclic task	☒ Time-controlled task (INTERVAL)	☐ Event-controlled task (EVENT)	☐ Interrupt task
-------------------	---	---	--	---

Variable names

Inputs (Variable type: VAR_INPUT)		
Name	Data type	Value/Meaning
dnSyncAngle	DINT	Cutting angle for which the knife has to run in synchronism with the material (y_{sync} , the value is entered in 0.0001[s_units]).
dnTotalYFeedAngle	DINT	Y final value of the selected motion profile in 0.0001[s_units].

Outputs (Variable type: VAR_OUTPUT)		
Name	Data type	Value/Meaning
dnY1Value	DINT	Cross-cutter knife drum starting and stopping position. The value is scaled in 0.0001[s_units].

(*Calc the start Position of the knife drum*)

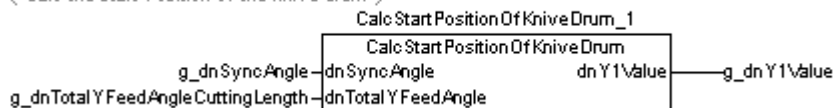
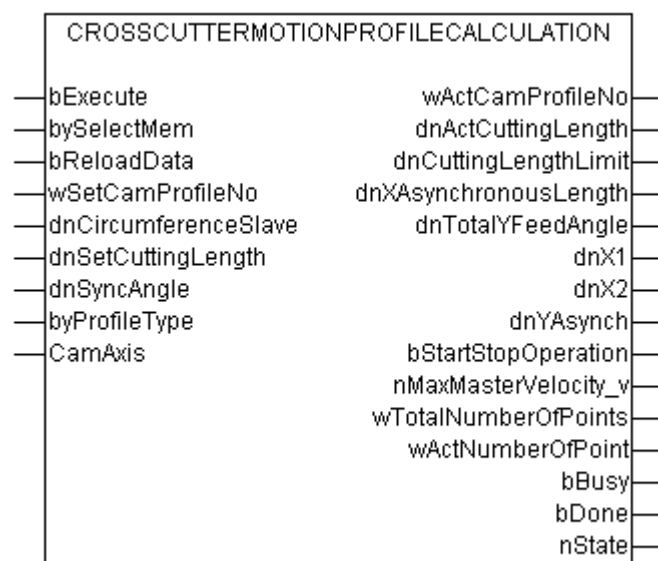


Figure50: Example wiring of the CalcStartPositionOfKnifeDrum FB in the prepared solution

6.6.9 CrossCutterMotionProfileCalculation function block

The CrossCutterMotionProfileCalculation function block calculates motion profiles for cross-cutter applications with second-order polynomials (linear speed ramps), fifth-order polynomials and sloping sine line for the motion function of the asynchronous section. The motion profile is recalculated and saved in the defined area of the profile data field on the ServoPLC. The calculation will only be made correctly if a basic profile with appropriate boundary parameters is available in the profile data memory on the target system.

The function block also calculates an appropriate torque precontrol profile.



Task information

Can be called in:	☒ Cyclic task	☐ Time-controlled task (INTERVAL)	☐ Event-controlled task (EVENT)	☐ Interrupt task
-------------------	---	--	--	---



Caution!

The *CrossCutterMotionProfileCalculation* function block must only be processed in the cyclic task (PLC_PRG or a subroutine called there) because it uses floating-point arithmetic.

Cross Cutter

Appendix

Variable name

Inputs (Variable type: VAR_INPUT)								
Name	Data type	Meaning						
bExecute	BOOL	Start the calculation sequence for a new cross-cutter motion profile. <table><tr><td>FALSE => TRUE</td><td>Starts a new calculation sequence for a cross-cutter motion profile. The state of the function block changes from <i>byOperationMode</i> = 0 to <i>byOperationMode</i> = 10. Status signal <i>bBusy</i> is set to TRUE until the calculation is complete or is aborted.</td></tr></table>	FALSE => TRUE	Starts a new calculation sequence for a cross-cutter motion profile. The state of the function block changes from <i>byOperationMode</i> = 0 to <i>byOperationMode</i> = 10. Status signal <i>bBusy</i> is set to TRUE until the calculation is complete or is aborted.				
FALSE => TRUE	Starts a new calculation sequence for a cross-cutter motion profile. The state of the function block changes from <i>byOperationMode</i> = 0 to <i>byOperationMode</i> = 10. Status signal <i>bBusy</i> is set to TRUE until the calculation is complete or is aborted.							
bySelectMem	BYTE	Selection of profile data field: Two RAM storage blocks (application RAM) are available for this purpose and can be selected as follows: <table><tr><td>1:</td><td>RAM block 1</td></tr><tr><td>2:</td><td>RAM block 2</td></tr></table> The signal is limited to the values 1 or 2 inside the function block. <u>Note:</u> If you change motion profiles in the RAM memory area which are also accessed by the <i>L_CamData</i> function block for drive control, make sure that <i>L_CamData</i> is using a motion profile other than the one changed via the <i>CrossCutterMotionProfileCalculation</i> function block.	1:	RAM block 1	2:	RAM block 2		
1:	RAM block 1							
2:	RAM block 2							
bReloadData	BOOL	Input signal for data reinitialisation (e.g. after profile data has been reloaded via the CamDesigner online download): <table><tr><td>FALSE => TRUE</td><td>Reinitialisation of profile data in the <i>L_CamProfileData1</i> function block used internally</td></tr></table> <u>Note:</u> During a calculation sequence (<i>bBusy</i> = TRUE) a state change at input signal <i>bReloadData</i> will be ignored.	FALSE => TRUE	Reinitialisation of profile data in the <i>L_CamProfileData1</i> function block used internally				
FALSE => TRUE	Reinitialisation of profile data in the <i>L_CamProfileData1</i> function block used internally							
wSetCamProfileNo	WORD	Selection of the motion profile to be changed <u>Note:</u> Make sure that you never change motion profiles currently being used for setpoint generation for drive control (e.g. in the <i>L_CamData</i> function block)!						
dnCircumferenceSlave	DINT	Cutting circle circumference <i>X</i>_{circumference} of the knife drum in 0.0001[m_units]: <u>Example:</u> The cutting circle diameter of the knife drum is 127.3240[mm]. The circumference is calculated by multiplying the diameter by π (= 3.141592654 ...) and results in a value of 400.0000[m_units]. You should therefore assign a value of 4000000 to input variable <i>dnCircumferenceSlave</i>. <u>Note:</u> Do not enter the synchronous cutting length here (this may not be the same as the cutting circle circumference on knife drums with multiple knives). Always enter the full circumference of the knife cutting circle here.						
dnSetCuttingLength	DINT	Entry indicating the required cutting length <i>x</i>_{max} in 0.0001[m_units]						
dnSyncAngle	DINT	Cutting angle for which the knife has to run in synchronism with the material (<i>y</i>_{sync}, the value is entered in 0.0001[s_units]). <u>Example:</u> A <i>dnSyncAngle</i> value = 300000 corresponds to a cutting angle of 30.0000[s_units] = 30.0000 [°].						
byProfileType	BYTE	Selection of type of motion for the asynchronous motion phase: <table><tr><td>0:</td><td>Second order polynomial (linear speed ramps)</td></tr><tr><td>1:</td><td>Fifth order polynomial</td></tr><tr><td>2:</td><td>Sloping sine line</td></tr></table> The input values at variable <i>byProfileType</i> are limited to this value range!	0:	Second order polynomial (linear speed ramps)	1:	Fifth order polynomial	2:	Sloping sine line
0:	Second order polynomial (linear speed ramps)							
1:	Fifth order polynomial							
2:	Sloping sine line							

Cross Cutter

Appendix

Inputs (Variable type: VAR_INPUT)			
CamAxis	CAM_AXIS_REF	Axis data for the master and cross-cutter axis (profile axis):	
		wXGearNum	Gearbox ratio master value (numerator)
		wXGearDenom	Gearbox ratio master value (denominator)
		dwXFeedConst	Feed constant of master drive
		wYGearNum	Gearbox ratio cross-cutter axis (numerator)
		wYGearDenom	Gearbox ratio cross-cutter axis (denominator)
		dwYFeedConst	Feed constant of cross-cutter axis (must always be set to 3600000 = 360.0000[s_units])
		wXCycles	Master speed in [incr./ms] at which the cross-cutter drive will reach its rated number of cycles
		<u>Note:</u> When using the prepared solution you can link global variable structure <i>g_CamAxis</i> to this input.	

Outputs (Variable type: VAR_OUTPUT)			
Name	Data type	Meaning	
wActCamProfileNo	WORD	Number of the last motion profile changed <u>Note:</u> The state of this output variable only changes when a new calculation sequence is started via a positive signal edge at input signal <i>bExecute</i> .	
dnActCuttingLength	DINT	Cutting length x_{cut} used to calculate the motion profile with number <i>wActCamProfileNo</i> (display in 0.0001[m_units]). <u>Note:</u> The state of this output variable only changes when a new calculation sequence is started via a positive signal edge at input signal <i>bExecute</i> .	
dnCuttingLengthLimit	DINT	Maximum cutting length in 0.0001[m_units] possible when the cross cutter is in continuous duty. Selecting a longer cutting length will lead to a motion profile with a break in the asynchronous motion phase. <u>Note:</u> The value of this output signal is only recalculated during a calculation sequence (<i>bBusy</i> = TRUE).	
dnXAsynchronousLength	DINT	Material length in 0.0001[m_units] to which the cross cutter is set in the asynchronous motion phase. The value is the difference between the cutting length x_{cut} and the synchronous x length x_{sync} . <u>Note:</u> The value of this output signal is only recalculated during a calculation sequence (<i>bBusy</i> = TRUE).	
dnTotalYFeedAngle	DINT	Y final value of the motion profile selected via <i>wActCamProfileNo</i> in 0.0001[s_units]. The value is taken from the y final value of the basic cam defined in CamDesigner. <u>Note:</u> The value of this output signal is only recalculated during a calculation sequence (<i>bBusy</i> = TRUE).	
dnX1	DINT	Starting point of the pause in 0.0001[m_units] only if <i>bStartStopOperation</i> = TRUE.	
dnX2	DINT	Finishing point of the pause in 0.0001[m_units] only if <i>bStartStopOperation</i> = TRUE.	
dnYAsynch	DINT	Y position in 0.0001[s_units] at which the cross-cutter knife drum will be in the initial position.	
bStartStopOperation	BOOL	Status signal: Newly calculated cam induces start/stop operation of cross cutter.	
		FALSE	The selected cutting length (<i>dnActCuttingLength</i>) is less than the limit cutting length for start/stop operation (<i>dnCuttingLengthLimit</i>).
		TRUE	The selected cutting length (<i>dnActCuttingLength</i>) is greater than the limit cutting length for start/stop operation (<i>dnCuttingLengthLimit</i>) => The length of the pause will be extended accordingly in the asynchronous motion phase.
nMaxMasterVelocity_v	INT	Maximum possible line speed for the set cutting length in [incr./ms].	
wTotalNumberOfPoints	WORD	Number of interpolation points in the motion profile selected via <i>wActCamProfileNo</i> <u>Note:</u> The value of this output signal is only reset during a calculation sequence (<i>bBusy</i> = TRUE).	
wActNumberOfPoint	WORD	Newly calculated interpolation point in the motion profile selected via <i>wActCamProfileNo</i> : This output can, for example, be used as a progress indicator. The value range is 0 ... (<i>wTotalNumberOfPoints</i> - 1).	
bBusy	BOOL	The calculation sequence for the new cross-cutter motion profile is in progress.	
		FALSE	The calculation sequence for the new cross-cutter motion profile has been completed successfully (<i>bDone</i> = TRUE) or aborted (<i>nState</i> < 0).
		TRUE	The calculation sequence for the cross-cutter motion profile (<i>wActCamProfileNo</i>) is in progress.

Cross Cutter Appendix

Outputs (Variable type: VAR_OUTPUT)		
Name	Data type	Meaning
bDone	BOOL	The calculation sequence for the new cross-cutter motion profile has been completed successfully.
		FALSE The calculation sequence for the new cross-cutter motion profile has not been completed.
		TRUE The calculation sequence for the new cross-cutter motion profile has been completed successfully. Starting a new calculation sequence will always reset this status output to FALSE (positive edge at <i>bExecute</i>).
nState	INT	Status of calculation sequence:
		0 Calculation ok.
		-1 No LC9 file found/no profile data in the absolute data model/read error when accessing profile data.
		-2 Motion profile selected via <i>wSetCamProfileNo</i> not found.
		-4 The angle of motion of the knife drum for the asynchronous motion phase is less than or equal to zero (check y final value of basic profile and cutting angle!).
		-8 The entry indicating the cutting circle circumference of the knife drum is zero (input variable <i>dnCircumferenceSlave</i>).
		-16 The preset cutting length <i>dnActCuttingLength</i> is less than the material length corresponding to the cutting angle.
		-32 The motion profile selected via <i>wActCamProfileNo</i> contains fewer than 20 interpolation points.
		-64 The selected motion profile does not start at zero (0.0000/0.0000).
		-128 Cutting angle too small (input signal <i>dnSyncAngle</i>).
		-256 The y final value of the basic profile for profile number <i>wActCamProfileNo</i> is not equal to a value $360.0000[^\circ]/n_{\text{cuts/rev}}$. ($n_{\text{cuts/rev}}$ = Number of knives on knife drum, values of 1, 2, 3, 4, 5 or 6 are possible) By way of confirmation, the actual y final value is output at variable <i>dnTotalYFeedAngle</i> in 0.0001[s_units].
		-512 Internal read access to code C0011 failed.
		If status variable <i>nState</i> is exhibiting negative values, the <i>dnCuttingLengthLimit</i> , <i>dnXAsynchronousLength</i> , <i>dnTotalYFeedAngle</i> , <i>bStartStopOperation</i> and <i>wTotalNumberOfPoints</i> output signals may take on invalid values, in which case it will not be possible to use them subsequently in the control program!

Internal (Variable type: VAR)		
Name	Data type	Meaning
byOperationMode	BYTE	State of the function block
		0 Function block is in standby mode (calculation not in progress).
		10 Recalculation in progress for a cross-cutter motion profile.
bySubState	BYTE	Internal cutting counter during recalculation of a cross-cutter profile (only if <i>byOperationMode</i> = 10): Inside the function block, this variable is used to distribute complex calculations and recalculations over a number of cyclic task runs.

6.6.10 TP_Register – Position tracking of mark pulses

The *TP_Register* function block detects print-mark pulses via a positive edge at input *bTpReceived_b* and tracks the mark pulses detected over an adjustable distance (input *dnOffset_p*). The mark pulses are read in and output (with delay) on a touch-probe-specific basis. The *Tp_Register* function block can track up to 32 mark pulses simultaneously. The function block uses the integration of the speed input *nIn_v* to calculate the movement of the web and therefore that of the print marks.

A reset function (input *bReset_b* = TRUE) is used to delete all registers assigned internally. The reset procedure takes 8 task cycles to complete and is displayed by status output *bResetState_b* = TRUE.

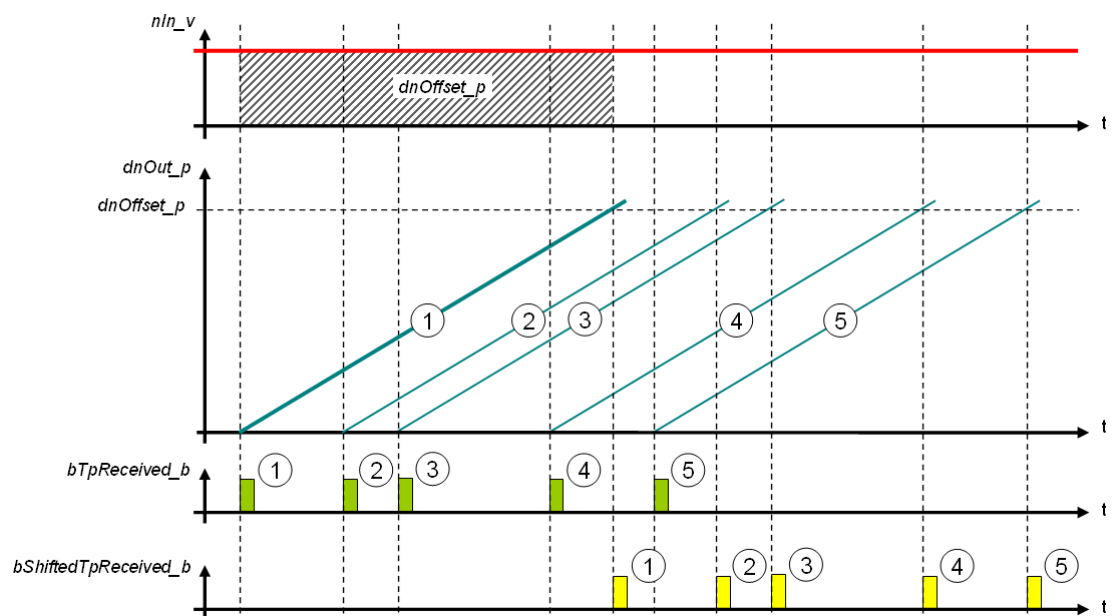
If more than 32 print marks need to be managed in the register simultaneously, an error message will be output via output *bOverflow_b* = TRUE. In this case, the last print-mark signal detected is always taken into account.



TIP!

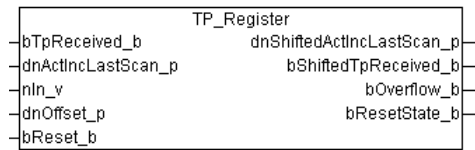
However, the *Tp_Register* function block can easily be expanded to accommodate more registers, as it has been programmed in the IEC Structured Text (ST) language.

The following diagram illustrates, for the *TP_Register* function block, example temporal characteristics for the print-mark pulses detected (input signal *bTpReceived_b*) and shifted via the offset distance (output signal *bShiftedTpReceived_b*) at constant material speed (*nIn_v* = const.):



Cross Cutter

Appendix



Task information

Can be called in:	☐ Cyclic task	☒ Time-controlled task (INTERVAL)	☐ Event-controlled task (EVENT)	☐ Interrupt task
-------------------	--------------------------------------	---	--	---

Variable names

Inputs (Variable type: VAR_INPUT)						
Name	Data type	Value/Meaning				
<i>bTpReceived_b</i>	BOOL	Input bit for the print-mark signal: A FALSE => TRUE edge tells the <i>Tp_Register</i> function block that a print-mark signal has been detected. When the print-mark pulses are evaluated, system variable <i>DFIN_bTpReceived_b</i> is typically evaluated at this input.				
<i>dnActInclLastScan_p</i>	DINT	Last-scan information: This value is made available by the touch-probe interface and indicates the material feed in [incr.] between the point at which the print-mark signal occurred and the start of the calling task. When the print-mark pulses are evaluated, system variable <i>DFIN_dnInclLastScan_p</i> is typically evaluated at this input.				
<i>nIn_v</i>	INT	Material speed in [incr./ms] (16384 = 15000[rpm]): As soon as a print-mark pulse appears at input <i>bTpReceived_b</i> , in the <i>Tp_Register</i> function block this value is integrated to a position (internal variable <i>dnOut_p</i>). If the integrated position exceeds the set offset distance <i>dnOffset_p</i> , a pulse will be output at output <i>bShiftedTpReceived_b</i> . When the print-mark pulses are evaluated, system variable <i>DFIN_nIn_v</i> is typically evaluated at this input.				
<i>dnOffset_p</i>	DINT	Entry indicating the offset distance in [incr.]: Once it has been detected at input <i>bTpReceived_b</i> , inside the function block the print-mark pulse will pass through this distance before being output at output <i>bShiftedTpReceived_b</i> as a FALSE-TRUE edge. When output at <i>bShiftedTpReceived_b</i> , the print-mark pulse is deleted from the register. <u>Note:</u> Only the value (and not the state) at input <i>dnOffset_p</i> is evaluated inside the function block: - At positive input speeds (<i>nIn_v</i> > 0) a pulse signal will be output at output <i>bShiftedTpReceived_b</i> if the integrated speed at <i>nIn_v</i> is greater than the offset value <i>dnOffset_p</i>. - At negative input speeds (<i>nIn_v</i> < 0) a pulse signal will be output at output <i>bShiftedTpReceived_b</i> if the integrated speed at <i>nIn_v</i> is less than the offset value <i>dnOffset_p</i>.				
<i>bReset_b</i>	BOOL	Input to reset all internal registers: <table><tr><td>FALSE</td><td>No effect – the <i>Tp_Register</i> function block works as described.</td></tr><tr><td>TRUE</td><td>Deletion of all internal registers: - All print-mark signals detected via <i>bTpReceived_b</i> but not yet output at <i>bShiftedTpReceived_b</i> are deleted from the internal registers (all elements of the internal data array <i>abEnable[0..31]</i> are set to FALSE). - The reset procedure takes 8 task cycles. During the reset procedure, status output <i>bResetState_b</i> = TRUE. </td></tr></table>	FALSE	No effect – the <i>Tp_Register</i> function block works as described.	TRUE	Deletion of all internal registers: - All print-mark signals detected via <i>bTpReceived_b</i> but not yet output at <i>bShiftedTpReceived_b</i> are deleted from the internal registers (all elements of the internal data array <i>abEnable[0..31]</i> are set to FALSE). - The reset procedure takes 8 task cycles. During the reset procedure, status output <i>bResetState_b</i> = TRUE.
FALSE	No effect – the <i>Tp_Register</i> function block works as described.					
TRUE	Deletion of all internal registers: - All print-mark signals detected via <i>bTpReceived_b</i> but not yet output at <i>bShiftedTpReceived_b</i> are deleted from the internal registers (all elements of the internal data array <i>abEnable[0..31]</i> are set to FALSE). - The reset procedure takes 8 task cycles. During the reset procedure, status output <i>bResetState_b</i> = TRUE.					

Cross Cutter Appendix

Outputs (Variable type: VAR_OUTPUT)								
Name	Data type	Value/Meaning						
<i>dnShiftedActIncLastScan_p</i>	DINT	Last-scan value assigned to the shifted print-mark pulse <i>bShiftedTpReceived_b</i> in [incr.]: This signal can be used for the touch-probe-accurate evaluation of the shifted print-mark pulse for additional function blocks (e.g. <i>L_DFSET.dnSetTPLastScan_p</i>).						
<i>bShiftedTpReceived_b</i>	BOOL	Output of the print-mark pulse shifted by offset distance <i>dnOffset_p</i> (FALSE-TRUE edge): When output at <i>bShiftedTpReceived_b</i> , the print-mark pulse is deleted from the register. <u>Note:</u> This signal can be used for the touch-probe-accurate evaluation of the shifted print-mark pulse for additional function blocks (e.g. <i>L_DFSET.bSetTPReceived_b</i>). The associated last-scan value for the shifted print-mark signal (shifted based on distance) is output at output <i>dnShiftedActIncLastScan_p</i> .						
<i>bOverflow_b</i>	BOOL	<table><tr><td colspan="2">Overflow indicator for the print-mark register:</td></tr><tr><td>FALSE</td><td>The number of print-mark pulses entered in the register simultaneously is less than or equal to 32. The print-mark register is running without errors.</td></tr><tr><td>TRUE</td><td>A further print-mark pulse has been detected via input <i>bTpReceived_b</i>, although 32 signals have already been entered in the print-mark register. The register structure is now implausible and this is resulting in incorrect output signals. Reinitialise the function block by setting <i>bReset_b</i> = TRUE to restore data plausibility.⁴</td></tr></table> The following measures can be taken to prevent a print-mark-register overflow: - Only use the <i>Tp_Register</i> function block to shift valid print-mark pulses which will later also be used for evaluation. You can connect the <i>Tp_Window</i> function block upstream of the print-mark register to mask out invalid print-mark pulses. - Mount the print-mark sensor nearer to the machine unit (this will reduce the offset distance <i>dnOffset_p</i> and allow register entries to be enabled more quickly).	Overflow indicator for the print-mark register:		FALSE	The number of print-mark pulses entered in the register simultaneously is less than or equal to 32. The print-mark register is running without errors.	TRUE	A further print-mark pulse has been detected via input <i>bTpReceived_b</i> , although 32 signals have already been entered in the print-mark register. The register structure is now implausible and this is resulting in incorrect output signals. Reinitialise the function block by setting <i>bReset_b</i> = TRUE to restore data plausibility. ⁴
Overflow indicator for the print-mark register:								
FALSE	The number of print-mark pulses entered in the register simultaneously is less than or equal to 32. The print-mark register is running without errors.							
TRUE	A further print-mark pulse has been detected via input <i>bTpReceived_b</i> , although 32 signals have already been entered in the print-mark register. The register structure is now implausible and this is resulting in incorrect output signals. Reinitialise the function block by setting <i>bReset_b</i> = TRUE to restore data plausibility. ⁴							
<i>bResetState_b</i>	BOOL	<table><tr><td colspan="2">Initialisation/reset active</td></tr><tr><td>FALSE</td><td>No effect – the <i>Tp_Register</i> function block works as described.</td></tr><tr><td>TRUE</td><td>Deletion of all internal registers active: All print-mark signals detected via <i>bTpReceived_b</i> but not yet output at <i>bShiftedTpReceived_b</i> are deleted from the internal registers (all elements of the internal data array <i>abEnable[0..31]</i> are set to FALSE).</td></tr></table>	Initialisation/reset active		FALSE	No effect – the <i>Tp_Register</i> function block works as described.	TRUE	Deletion of all internal registers active: All print-mark signals detected via <i>bTpReceived_b</i> but not yet output at <i>bShiftedTpReceived_b</i> are deleted from the internal registers (all elements of the internal data array <i>abEnable[0..31]</i> are set to FALSE).
Initialisation/reset active								
FALSE	No effect – the <i>Tp_Register</i> function block works as described.							
TRUE	Deletion of all internal registers active: All print-mark signals detected via <i>bTpReceived_b</i> but not yet output at <i>bShiftedTpReceived_b</i> are deleted from the internal registers (all elements of the internal data array <i>abEnable[0..31]</i> are set to FALSE).							

Internal (Variable type: VAR)		
Name	Data type	Meaning
<i>abEnable</i>	ARRAY [0..31] OF BOOL	Internal data field: Register entry active A print-mark signal detected at input <i>bTpReceived_b</i> increments the internal input index <i>byInIndex</i> and activates the associated register entry (<i>abEnable[byInIndex]</i> = TRUE). The register entry will be deactivated once the print-mark signal has been shifted (based on distance) by offset distance <i>dnOffset_p</i> or a reinitialisation of the <i>Tp_Register</i> function block has been activated (<i>bReset_b</i> = TRUE).
<i>adnBasis_p</i>	ARRAY [0..31] OF DINT	Internal data field: Basic position of print-mark signals The current positions of the print marks entered in the print-mark register will vary between 0 and the offset distance <i>dnOffset_p</i> . The current position of each of the 32 print marks which can be stored in the register can be calculated as follows using the basic position <i>adnBasis_p[i]</i> : $\text{Current position} = L_PHINT1.dnOut_p - adnBasis_p[i] \quad (\text{where } i = 0 \dots 31)$
<i>byInIndex</i>	BYTE	Input index: Indicates the data element of the print-mark register (value range 0 ... 31) with the last print mark detected via input <i>bTpReceived_b</i> .
<i>byOutIndex</i>	BYTE	Output index: Indicates the data element of the print-mark register (value range 0 ... 31) with the next print-mark pulse to be output at output <i>bShiftedTpReceived_b</i> .
<i>dnOut_p</i>	DINT	Internal position value: Position of the next print-mark pulse to be output at output <i>bShiftedTpReceived_b</i> , scaled in [incr.]

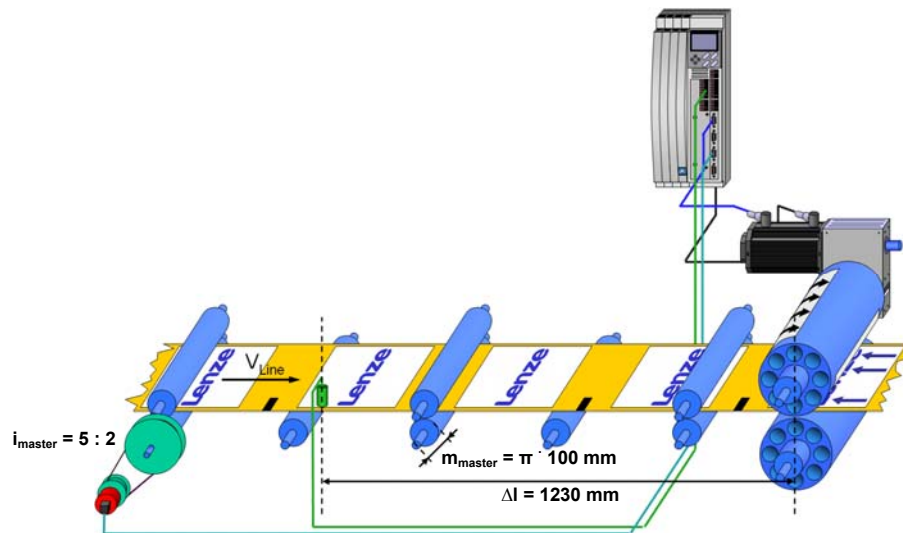
Connection example:

⁴ The *Tp_Register* function block is programmed in the IEC Structured Text (ST) language and the number of registers can easily be increased by the user.

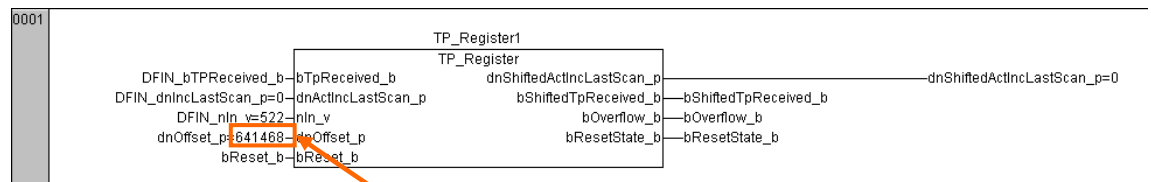
Cross Cutter Appendix

The following structure will be used to provide a practical illustration of the application of the *Tp_Register* function block:

Upstream of a printing unit, print marks are detected via a sensor mounted $\Delta l = 1230$ mm upstream of the printing unit. The web is moving forwards at approx. 60 m/min. The feed motion is detected via an incremental encoder driven via a belt ratio $i_{\text{master}} = 5 : 2$ by one of the feed rolls (diameter $\varnothing_{\text{master}} = 100$ mm).



The required shifting of the print-mark signals by the offset distance of $\Delta l = 1230$ mm can be achieved using the *Tp_Register* function block in the following connection:



The incremental value for the offset distance (input signal *dnOffset_p*) is calculated using the following formula:

$$\begin{aligned}
 dnOffset_p &= 65536[\text{incr./rev}] \cdot i_{\text{Master}} \cdot \frac{\Delta l}{m_{\text{Master}}} = \\
 &= 65536[\text{incr./rev}] \cdot \frac{5}{2} \cdot \frac{1230[\text{mm}]}{\pi \cdot 100[\text{mm/rev}]} = 641468,269.
 \end{aligned}$$

6.6.11 VersionHandling function block

The function block *VersionHandling* provides three versions for indication in the related codes:

- Version of the Prepared Solution
- Version of the application library (depending on the Prepared Solution)
- Version of the basic library (is used in all Prepared Solutions)

The function block calculates the three elements for the internal data array from the major and subversions using the following formula:

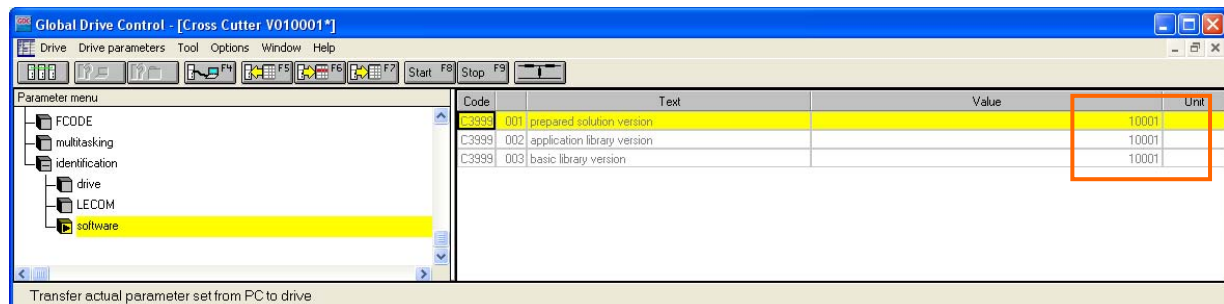
$adwVersion[1] := ((dwProRelease * 100 + dwProLevel)*100)+dwProServicePack;$

$adwVersion[2] := ((dwLib1Release * 100 + dwLib1Level)*100)+dwLib1ServicePack;$

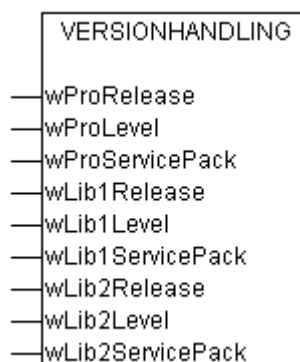
$adwVersion[3] := ((dwLib2Release * 100 + dwLib2Level)*100)+dwLib2ServicePack;$

The field variable *adwVersion* is intended for indication in display codes with two decimal positions. In this way the user can straightforwardly determine the version states for the project and libraries.

On the display of the code values the versions are indicated so that they are easy for the user to read (e.g. using GDC):



Interface



Task information

Can be called in:	☒ Cyclic task	☒ Time-controlled task (INTERVAL)	☒ Event-controlled task (EVENT)	☒ Interrupt task
-------------------	---	---	---	--

Variable names

Inputs (Variable type: VAR_INPUT)		
Identifier	Data type	Meaning
wProRelease	WORD	main version for the project:: the value entered must be in the range between 0 and 99.
wProLevel	WORD	subversion for the project:: the value entered must be in the range between 0 and 99.
wProServicePack	WORD	version of service-pack: the value entered must be in the range between 0 and 99.
wLib1Release	WORD	main version for library 1 (e.g. application library): the value entered must be in the range between 0 and 99.
wLib1Level	WORD	subversion for library 1 (e.g. application library): the value entered must be in the range between 0 and 99.
wLib1ServicePack	WORD	version of service-pack: the value entered must be in the range between 0 and 99.
wLib2Release	WORD	main version for library 2 (e.g. basic library): the value entered must be in the range between 0 and 99.
wLib2Level	WORD	subversion for library 2 (e.g. basic library): the value entered must be in the range between 0 and 99.
wLib2ServicePack	WORD	version of service-pack: the value entered must be in the range between 0 and 99.

Internal (Variable type: VAR)		
Identifier	Data type	Meaning
adwVersion	ARRAY [1..3] OF DWORD	Data field with indication of the three versions: the first and the second number is the main versio, the third and the fourth is the subversion and the fith and the sixth is the number of the service-pack. Example: main verison 1.0, subversion 1.0, service-pack 0.0, display: 10100 The values are calculated using the following formulas: $\text{adwVersion}[1] := ((\text{dwProRelease} * 100 + \text{dwProLevel}) * 100) + \text{dwProServicePack};$ $\text{adwVersion}[2] := ((\text{dwLib1Release} * 100 + \text{dwLib1Level}) * 100) + \text{dwLib1ServicePack};$ $\text{adwVersion}[3] := ((\text{dwLib2Release} * 100 + \text{dwLib2Level}) * 100) + \text{dwLib2ServicePack};$