FAST Application Software

Technology module

Flying Saw

Reference Manual EN

13531728

Lenze

Contents

1 About this documentation

This documentation ...

- contains detailed information on the functionalities of the "Flying Saw" technology module;
- is part of the "Controller-based Automation" manual collection. It consists of the following sets of documentation:

Documentation type	Subject
Product catalogue	Controller-based Automation (system overview, sample topologies) Lenze Controller (product information, technical data)
System manuals	Visualisation (system overview/sample topologies)
Communication manuals **Online helps**	Bus systems • Controller-based Automation EtherCAT® • Controller-based Automation CANopen® • Controller-based Automation PROFIBUS® • Controller-based Automation PROFINET®
Reference manuals **Online helps**	Lenze Controllers: • Controller 3200 C • Controller c300 • Controller p300 • Controller p500
Software manuals **Online helps**	Lenze Engineering Tools: • »PLC Designer« (programming) • »Engineer« (parameter setting, configuration, diagnostics) • »VisiWinNET® Smart« (visualisation) • »Backup & Restore« (data backup, recovery, update)

More technical documentation for Lenze components

Further information on Lenze products which can be used in conjunction with Controller-based Automation can be found in the following sets of documentation:

Planning / configuration / technical data	
☐	**Product catalogues** • Controller-based Automation • Controllers • Inverter Drives/Servo Drives
Mounting and wiring	
🕮	**Mounting instructions** • Controllers • Communication cards (MC-xxx) • I/O system 1000 (EPM-Sxxx) • Inverter Drives/Servo Drives • Communication modules
☐	**Hardware manuals** • Inverter Drives/Servo Drives
Parameter setting / configuration / commissioning	
☐	**Online help/reference manuals** • Controllers • Inverter Drives/Servo Drives • I/O system 1000 (EPM-Sxxx)
☐	**Online help/communication manuals** • Bus systems • Communication modules
Sample applications and templates	
☐	**Online help / software and reference manuals** • i700 application sample • Application Samples 8400/9400 • FAST Application Template Lenze/PackML • FAST technology modules

Symbols:

🕮 Printed documentation
☐ PDF file / online help in the Lenze engineering tool

Tip!

Current documentation and software updates with regard to Lenze products can be found in the download area at:

www.lenze.com

Target group

This documentation is intended for all persons who plan, program and commission a Lenze automation system on the basis of the Lenze FAST Application Software.

1.1 Document history

Version			Description
5.1	05/2017	TD17	• Content structure has been changed. • General revisions
5.0	03/2016	TD17	• New: Deleting the mark stack (📖 43) • General revisions
4.0	09/2015	TD17	• General revisions • New outputs: xInPosition, xPositioning • New parameter: lrBladeWidth • Chapter Moving to the parking position (📖 45) has been revised. • Fig. State machine (📖 45) has been revised
3.0	07/2015	TD17	• Corrections and additions • Content structure has been changed.
2.0	05/2015	TD17	• General editorial revision • Modularisation of the contents for the »PLC Designer« online help
1.0	04/2014	TD00	First edition

1.2 Conventions used

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

Type of information	Highlighting	Examples/notes
Spelling of numbers		
Decimal separator	Point	The decimal point is always used. For example: 1234.56
Text		
Program name	» «	»PLC Designer« ...
Variable names	*italics*	By setting *bEnable* to TRUE...
Function blocks	**bold**	The **L_MC1P_AxisBasicControl** function block ...
Function libraries		The **L_TT1P_TechnologyModules** function library ...

Source code	Font
	`"Courier new"`
		`...`
		`dwNumerator := 1;`
		`dwDenominator := 1;`
		`...`

Icons		
Page reference	(📖 6)	Reference to further information: Page number in PDF file.

Variable names

The conventions used by Lenze for the variable names of Lenze system blocks, function blocks, and functions are based on the "Hungarian Notation". This notation makes it possible to identify the most important properties (e.g. the data type) of the corresponding variable by means of its name, e.g. xAxisEnabled.

1.3 Definition of the notes used

The following signal words and symbols are used in this documentation to indicate dangers and important information:

Safety instructions

Layout of the safety instructions:

Pictograph and signal word!

(characterise the type and severity of danger)

Note

(describes the danger and gives information about how to prevent dangerous situations)

Pictograph	Signal word	Meaning
	Danger!	**Danger of personal injury through dangerous electrical voltage** Reference to an imminent danger that may result in death or serious personal injury if the corresponding measures are not taken.
	Danger!	**Danger of personal injury through a general source of danger** Reference to an imminent danger that may result in death or serious personal injury if the corresponding measures are not taken.
STOP	**Stop!**	**Danger of property damage** Reference to a possible danger that may result in property damage if the corresponding measures are not taken.

Application notes

Pictograph	Signal word	Meaning
	Note!	Important note to ensure trouble-free operation
	Tip!	Useful tip for easy handling
		Reference to another document

2 Safety instructions

Please observe the safety instructions in this documentation when you want to commission an automation system or a plant with a Lenze Controller.

The device documentation contains safety instructions which must be observed!

Read the documentation supplied with the components of the automation system carefully before you start commissioning the Controller and the connected devices.

Danger!

High electrical voltage

Injury to persons caused by dangerous electrical voltage

Possible consequences

Death or severe injuries

Protective measures

Switch off the voltage supply before working on the components of the automation system.

After switching off the voltage supply, do not touch live device parts and power terminals immediately because capacitors may be charged.

Observe the corresponding information plates on the device.

Danger!

Injury to persons

Risk of injury is caused by ...

- unpredictable motor movements (e.g. unintended direction of rotation, too high velocities or jerky movement);
- impermissible operating states during the parameterisation while there is an active online connection to the device.

Possible consequences

Death or severe injuries

Protective measures

- If required, provide systems with installed inverters with additional monitoring and protective devices according to the safety regulations valid in each case (e.g. law on technical equipment, regulations for the prevention of accidents).
- During commissioning, maintain an adequate safety distance to the motor or the machine parts driven by the motor.

Stop!

Damage or destruction of machine parts

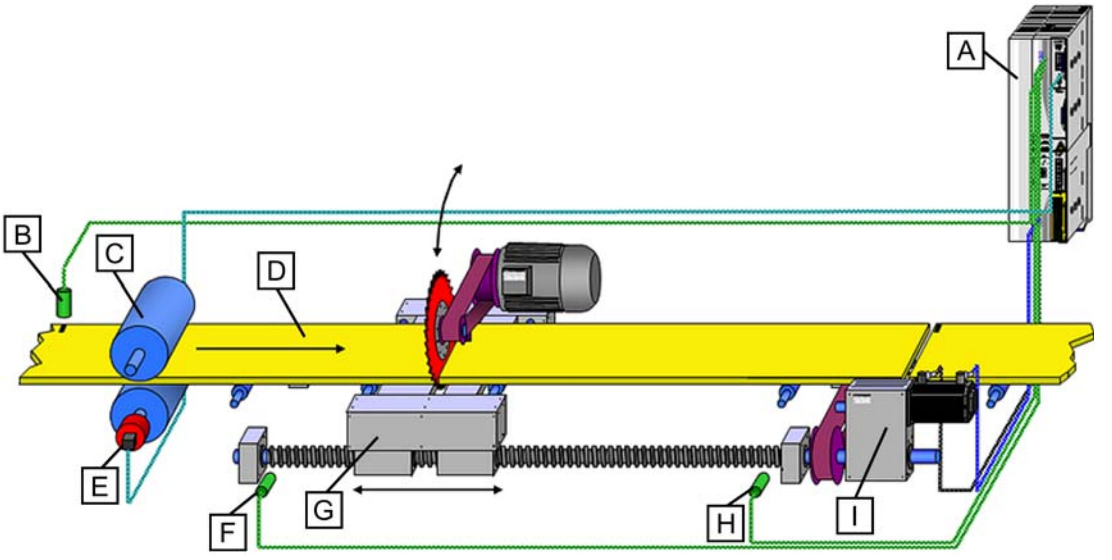
Damage or destruction of machine parts can be caused by ...

- Short circuit or static discharges (ESD);
- unpredictable motor movements (e.g. unintended direction of rotation, too high velocities or jerky movement);
- impermissible operating states during the parameterisation while there is an active online connection to the device.

Protective measures

- Always switch off the voltage supply before working on the components of the automation system.
- Do not touch electronic components and contacts unless ESD measures were taken beforehand.
- If required, provide systems with installed inverters with additional monitoring and protective devices according to the safety regulations valid in each case (e.g. law on technical equipment, regulations for the prevention of accidents).

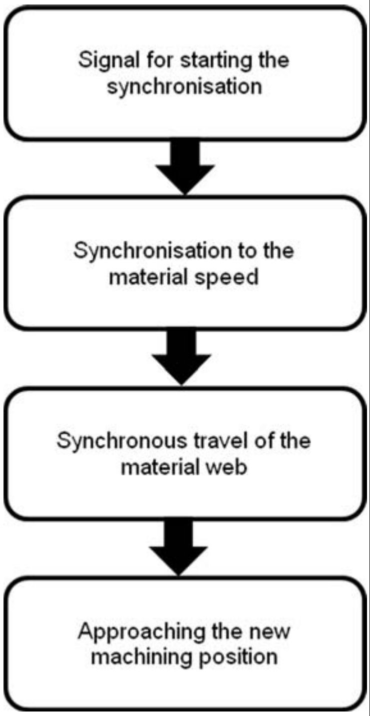
3 Functional description for the "Flying Saw"



A Inverter Drive
B Mark sensor (touch probe sensor)
C Feeding rollers
D Material web
E Detection of the master speed
F Negative hardware limit switch
G Saw carriage
H Positive hardware limit switch
I Motor and gearbox

[3-1] Typical mechanics of the technology module

The flying saw provides the application of a machining process (e.g. sawing process, glue application) to material that is continuously passing through. A typical flying saw process is divided into the following motion stages:



1. Waiting for the start signal for synchronisation:
   The drive is in the home position and is waiting for the start signal for synchronisation with the material passing through.
2. Synchronisation with material speed:
   The drive is synchronised with the speed of the web at the acceleration ramp specified. When the acceleration path has been passed through, the tool slide reaches position synchronicity with the material. At the machining unit, a "Synchronicity reached" status signal is set.
3. Synchronous run of the web:
   During the synchronous run with the material web, the product is machined (e.g. sawing process, glue application). When the machining process has been completed, the synchronous movement to the material web can be cancelled.
4. Approaching the new machining position:
   Completion of synchronous operation is initiated by an external "Machining process completed" control signal. The drive stops the synchronous movement to the material web and initiates the reversing process to the next machining position.

The "Flying saw" technology module offers the following functionalities:

- In the "Base" version, the technology module can only operate in a length-controlled manner. The saw carriage is synchronised to the material speed.
- The "State" version provides an extended function range of the "Base" version:
  The State version additionally provides mark-controlled operation, the recovery of the target position, travelling to a parking position, and following error monitoring.

▶ Overview of the functions (📖 12)

3.1 Overview of the functions

In addition to the basic functions for operating the **L_MC1P_AxisBasicControl** function block, the **stop function** and the **holding function**, the technology module offers the following functionalities which are assigned to the "Base" and "State" versions:

Functionality	Versions	
	Base	State
Manual jog (jogging) (🕮 28)	●	●
Homing (🕮 29)	●	●
Setting motion profiles (linear/S-ramps) (🕮 30)	●	●
First synchronisation of the saw carriage to the material speed (🕮 31)	●	●
Length-controlled operation (cutting of relative product lengths) (🕮 34)	●	●
Mark-controlled operation (cutting to position markers) (🕮 37)		●
Restoring the target position (🕮 40)		●
Deleting the mark stack (🕮 43)		●
Moving to the parking position (🕮 45)		●
Window for following error monitoring (🕮 47)		●

»PLC Designer« Online help

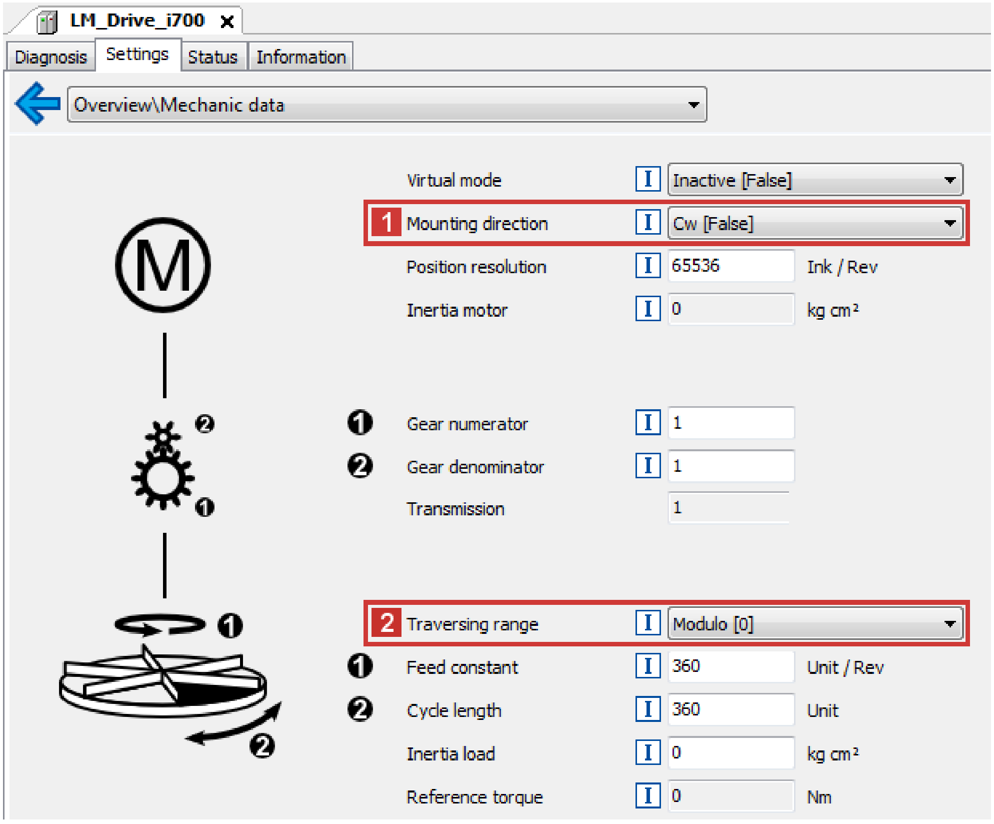
Here you will find detailed information on the **L_MC1P_AxisBasicControl** function block, the **stop function** and the **holding function**.

3.2 Important notes on how to operate the technology module

For the "Flying Saw" technology module, the axes have to be configured as follows:

- The master axis has to be a rotary axis.
- The slave axis has to be a linear axis.

Go to the »PLC Designer« and set the corresponding 2 machine measuring system for <u>each</u> axis under the **Settings** tab:



During the cutting process the master axis must only move in positive direction.

In the application, the mounting direction 1 must also be set so that the master axis only moves in positive direction.

Setting of the operating mode

The operating mode for the slave axis has to be set to "cyclically synchronous position" (csp) because the axis is led via the master position value.

Controlled start of the axes

Motion commands that are set in the inhibited axis state (*xAxisEnabled* = FALSE) after enable (*xRegulatorOn* = TRUE) must be activated again by a FALSE↗TRUE edge.

In this way it is prevented that the drive starts in an uncontrolled manner after controller enable.

Example Manual jog (jogging) (📖 28)**:**

1. In the inhibited axis state (*xAxisEnabled* = FALSE), *xJogPos* is set to TRUE.
   - *xRegulatorOn* = FALSE (axis is inhibited.)
     ==> "READY" state (*xAxisEnabled* = FALSE)
   - *xJogPos* = TRUE (manual jog is to be executed.)
2. Enable axis.
   - *xRegulatorOn* = TRUE
     ==> "READY" state (*xAxisEnabled* = TRUE)
3. Execute manual jog.
   - *xJogPos* = FALSE↗TRUE
     ==> "JOGPOS" state

3.3 L_TT1P_FlyingSaw[Base/State] function block

The figure shows the relation of the inputs and outputs to the "Base" and "State" versions.

The additional inputs and outputs of the "State" version are shaded.

L_TT1P_FlyingSaw[Base/State]

Type	Input	Output	Type
Base			
BOOL	xEnableInternalControl	xInternalControlActive	BOOL
BOOL	xEnable	eTMState	L_TT1P_States
scCtrl_ABC	scCtrlABC	scStatusABC	scStatus_ABC
BOOL	xResetError	xError	BOOL
BOOL	xRegulatorOn	xWarning	BOOL
BOOL	xStop	eErrorID	L_IE1P_Error
BOOL	xHalt	scErrorInfo	L_TT1P_scErrorInfo
L_TT1P_scPar_FlyingSaw[Base/State]	scPar	xAxisEnabled	BOOL
AXIS_REF	MasterAxis	scSignalFlow	L_TT1P_scSF_FlyingSaw[Base/State]
AXIS_REF	SlaveAxis	xDone	BOOL
L_TT1P_scAP_FlyingSaw[Base/State]	scAccessPoints	xBusy	BOOL
BOOL	xJogPos	xIsHomed	BOOL
BOOL	xJogNeg	xHwLimitSwitchNeg	BOOL
BOOL	xHomeExecute	xHwLimitSwitchPos	BOOL
BOOL	xHomeAbsSwitch	xSwLimitEnabled	BOOL
BOOL	xEnableHWLimit	xSwLimitSwitchActive	BOOL
BOOL	xHWLimitPos	xSynchronised	BOOL
BOOL	xHWLimitNeg	xAccDecSync	BOOL
BOOL	xStartFlyingSaw	xInPosition	BOOL
BOOL	xSyncInInstant	xPositioning	BOOL
BOOL	xExecuteNextCut	lrActPos	LREAL
BOOL	xExecuteGap	lrActVel	LREAL
LREAL	lrCuttingLength	lrActFollowingError	LREAL
BOOL	xSetMaterialCounter	lrMaterialCounter	LREAL
		lrTargetPos	LREAL
		lrCutGap	LREAL
State			
BOOL	xMoveToParkPos	byRemainMarkStack	BYTE
BOOL	xResumeTargetPos		
BOOL	xResetMarkStack		
BOOL	xMarkReceive		
LREAL	lrActMarkPosIn		

3.3.1 Inputs and outputs

Designator / Data type	Description	Available in version: Base	State
MasterAxis AXIS_REF	Reference to the master axis (master axis)	●	●
SlaveAxis AXIS_REF	Reference to the slave axis	●	●

3.3.2 Inputs

Designator / Data type	Description		Available in version: Base	State
xEnableInternalControl BOOL	TRUE	In the visualisation, the internal control of the axis can be selected via the "Internal Control" axis.	●	●
xEnable BOOL	Execution of the function block		●	●
	TRUE	The function block is executed.		
	FALSE	The function block is not executed.		
scCtrlABC scCtrl_ABC	Input structure for the **L_MC1P_AxisBasicControl** function block • scCtrlABC can be used in "Ready" state. • If there is a request, the state changes to "Service". • The state change from "Service" back to "Ready" takes place if there are no more requests.		●	●
xResetError BOOL	TRUE	Reset axis error or software error.	●	●
xRegulatorOn BOOL	TRUE	Activate controller enable of the axis (via the **MC_Power** function block).	●	●
xStop BOOL	TRUE	Cancel the active movement and brake the axis to a standstill with the deceleration defined via the lrStopDec parameter. • The state changes to "Stop". • The technology module remains in the "Stop" state as long as xStop is set to TRUE (or xHalt = TRUE). • The input is also active with "Internal Control".	●	●
xHalt BOOL	TRUE	Cancel the active movement and brake the axis to a standstill with the deceleration defined via the lrHaltDec parameter. • The state changes to "Stop". • The technology module remains in the "Stop" state as long as xStop is set to TRUE (or xHalt = TRUE).	●	●
scPar L_TT1P_scPar_FlyingSaw[Base/State]	The parameter structure contains the parameters of the technology module. The data type depends on the version used (Base/State).		●	●
scAccessPoints L_TT1P_scAP_FlyingSaw[Base/State]	Structure of the access points The data type depends on the version used (Base/State).		●	●
xJogPos BOOL	TRUE	Traverse axis in positive direction (manual jog). If xJogNeg is also TRUE, the traversing direction selected first remains set.	●	●
xJogNeg BOOL	TRUE	Traverse axis in negative direction (manual jog). If xJogPos is also TRUE, the traversing direction selected first remains set.	●	●

Designator / Data type	Description		Available in version: Base	State
xHomeExecute BOOL	The input is edge-controlled and evaluates the rising edge.		●	●
	FALSE↗TRUE	Start homing. The function is aborted via the xStop input.		
xHomeAbsSwitch BOOL	TRUE	Connection for reference switch: For homing modes with a reference switch, connect this input to the digital signal which maps the state of the reference switch.	●	●
xEnableHWLimit BOOL	TRUE	The evaluation of the travel range limit switch (hardware limit switch) is activated.	●	●
xHWLimitPos BOOL	Positive hardware limit switch Connect this input to the corresponding digital input that is connected to the limit switch.		●	●
	TRUE	The positive hardware limit switch has been reached or approached. • The xHwLimitSwitchPos output is also set to TRUE. • The axis is brought to a standstill with the deceleration in the alrStopDec parameter. • The state changes to "ERROR" with the error message ’20500’ (HWLimitPos).		
xHWLimitNeg BOOL	Negative hardware limit switch Connect this input to the corresponding digital input that is connected to the limit switch.		●	●
	TRUE	The negative hardware limit switch has been reached or approached. • The xHwLimitSwitchNeg output is also set to TRUE. • The axis is brought to a standstill with the deceleration in the alrStopDec parameter. • The state changes to "ERROR" with the error message ’20501’ (HWLimitNeg).		
xStartFlyingSaw BOOL	TRUE	The "Flying Saw" technology module is activated. • Parameter xSyncFlyingSaw = TRUE: The saw carriage is synchronised to the material position. • Parameter xSyncFlyingSaw = FALSE: The saw carriage waits for a signal for synchronising to the material position.	●	●
	FALSE	The saw carriage is disengaged from the material position and brakes to a standstill. • If the saw carriage is in waiting position, it is only disengaged. • xParkPos parameter = TRUE: the saw carriage is disengaged first and then travels to its parking position.		
xSyncInInstant BOOL	The input is edge-controlled and evaluates the rising edge.		●	●
	FALSE↗TRUE	The saw carriage is synchronised to the material position. This can occur in the parking position or during a return.		
xExecuteNextCut BOOL	The input is edge-controlled and evaluates the rising edge.		●	●
	FALSE↗TRUE	The saw carriage moves to the next machining position.		
xExecuteGap BOOL	The input is edge-controlled and evaluates the rising edge.		●	●
	FALSE↗TRUE	The saw carriage moves by the distance set in the lrGap parameter. This distance is added to the relative travel way (lrCuttingLength input).		
lrCuttingLength LREAL	Cutting length (relative positioning) or cutting position (absolute positioning)		●	●

<table>
<tr><th rowspan="2">Designator
Data type</th><th rowspan="2" colspan="2">Description</th><th colspan="2">Available in version</th></tr>
<tr><th>Base</th><th>State</th></tr>
<tr><td rowspan="2">xSetMaterialCounter
BOOL</td><td colspan="2">The input is edge-controlled and evaluates the rising edge.</td><td rowspan="2">●</td><td rowspan="2">●</td></tr>
<tr><td>FALSE↗TRUE</td><td>Sets the material position (lrMaterialCounter output) to the value of the lrStartPosition parameter.</td></tr>
<tr><td rowspan="2">xMoveToParkPos
BOOL</td><td colspan="2">The input ...
• is only evaluated if the technology module is in "READY" state;
• is edge-controlled and evaluates the rising edge.</td><td rowspan="2"></td><td rowspan="2">●</td></tr>
<tr><td>FALSE↗TRUE</td><td>The saw carriage is driven to the parking position (lrParkPosition parameter).</td></tr>
<tr><td rowspan="2">xResumeTargetPos
BOOL</td><td colspan="2">The input is edge-controlled and evaluates the rising edge.</td><td rowspan="2"></td><td rowspan="2">●</td></tr>
<tr><td>FALSE↗TRUE</td><td>The saw carriage is moved to the position of the lrTargetPos output.</td></tr>
<tr><td rowspan="2">xResetMarkStack
BOOL</td><td colspan="2">The input is edge-controlled and evaluates the rising edge.</td><td rowspan="2"></td><td rowspan="2">●</td></tr>
<tr><td>FALSE↗TRUE</td><td>All marks on the mark stack are removed.
If this input is confirmed during the clutched-in state, the axis travels automatically to the waiting position (lrLowerLimit parameter).</td></tr>
<tr><td>xMarkReceive
BOOL</td><td>TRUE</td><td>A touch probe mark has been detected in the connected touch probe sensor.</td><td></td><td>●</td></tr>
<tr><td>lrActMarkPosIn
LREAL</td><td colspan="2">Current position of the touch probe mark with regard to the axis reference used.
• Unit: units</td><td></td><td>●</td></tr>
</table>

3.3.3 Outputs

Designator Data type	Description		Available in version Base	State
xInternalControlActive BOOL	TRUE	The internal control of the axis is activated via the visualisation. (xEnableInternalControl input = TRUE)	●	●
eTMState L_TT1P_States	Current state of the technology module ▸ State machine (📖 24)		●	●
scStatusABC scStatus_ABC	Structure of the status data of the **L_MC1P_AxisBasicControl** function block		●	●
xError BOOL	TRUE	There is an error in the technology module.	●	●
xWarning BOOL	TRUE	There is a warning in the technology module.	●	●
eErrorID L_IE1P_Error	ID of the error or warning message if xError = TRUE or xWarning = TRUE. Possible error or warning messages (IDs): 17125: The position of the slave axis exceeds the set "lrUpperLimit" parameter. 17126: The position of the slave axis falls below the set "lrLowerLimit" parameter. 17127: The target position to be approached exceeds the set "lrUpperLimit" parameter. 17128: The position detected by the mark sensor is out of the valid range. **"FAST technology modules" reference manual:** Here you can find information on further error or warning messages.		●	●
scErrorInfo L_TT1P_scErrorInfo	Error information structure for a more detailed analysis of the error cause		●	●
scSignalFlow L_TT1P_scSF_FlyingSaw[Base/State]	Structure of the signal flow The data type depends on the version used (Base/State). ▸ Signal flow diagrams (📖 25)		●	●
xAxisEnabled BOOL	TRUE	The axis is enabled.	●	●
xDone BOOL	TRUE	The request/action has been completed successfully.	●	●
xBusy BOOL	TRUE	The request/action is currently being executed.	●	●
xIsHomed BOOL	TRUE	The axis has been referenced (reference known).	●	●
xHwLimitSwitchNeg BOOL	TRUE	The negative hardware limit switch has been reached or approached. • The xHwLimitNeg input has to be connected to the digital input that is connected to the limit switch. • The xHWLimitNeg input is also set to TRUE. • The drive is brought to a standstill with the deceleration set in the lrStopDec parameter. • The state changes to "ERROR" with the error message '20501' (HWLimitNeg).	●	●

_ _

Designator / **Data type**	**Description**		**Available in version**	
			Base	**State**
xHwLimitSwitchPos BOOL	TRUE	The positive hardware limit switch has been reached or approached. • The xHwLimitPos input has to be connected to the digital input that is connected to the limit switch. • The xHWLimitPos input is also set to TRUE. • The drive is brought to a standstill with the deceleration set in the lrStopDec parameter. • The state changes to "ERROR" with the error message ’20500’ (HWLimitPos).	●	●
xSwLimitEnabled BOOL	TRUE	Activate the monitoring of the software limit positions.	●	●
xSwLimitSwitchActive BOOL	TRUE	A software limit position has been reached or exceeded. • The drive is brought to a standstill with the deceleration set in the lrStopDec parameter. • The state changes to "ERROR" with error message '20306' (SWLimitPos) or '20307' (SWLimitNeg).	●	●
xSynchronised BOOL	TRUE	The axis is synchronised to the cam.	●	●
xAccDecSync BOOL	TRUE	The synchronisation function is active. The axis is synchronised or desynchronised (clutch opens or closes).	●	●
xInPosition BOOL	TRUE	The axis has reached the parking position.	●	●
xPositioning BOOL	TRUE	The axis travels to the parking position	●	●
lrActPos LREAL	Current position • Unit: units		●	●
lrActVel LREAL	Current velocity • Unit: units/s		●	●
lrActFollowingError LREAL	Current following error • Unit: units/s^2		●	●
lrMaterialCounter LREAL	Continuous material position • Unit: units		●	●
lrTargetPos LREAL	Target position of the saw carriage • Unit: units		●	●
lrCutGap LREAL	Distance between two cutting positions • Unit: units		●	●
byRemainMarkStack BYTE	Display of the positions still available (max. 15) on the mark stack			●

3.3.4 Parameters

L_TT1P_scPar_FlyingSaw[Base/State]

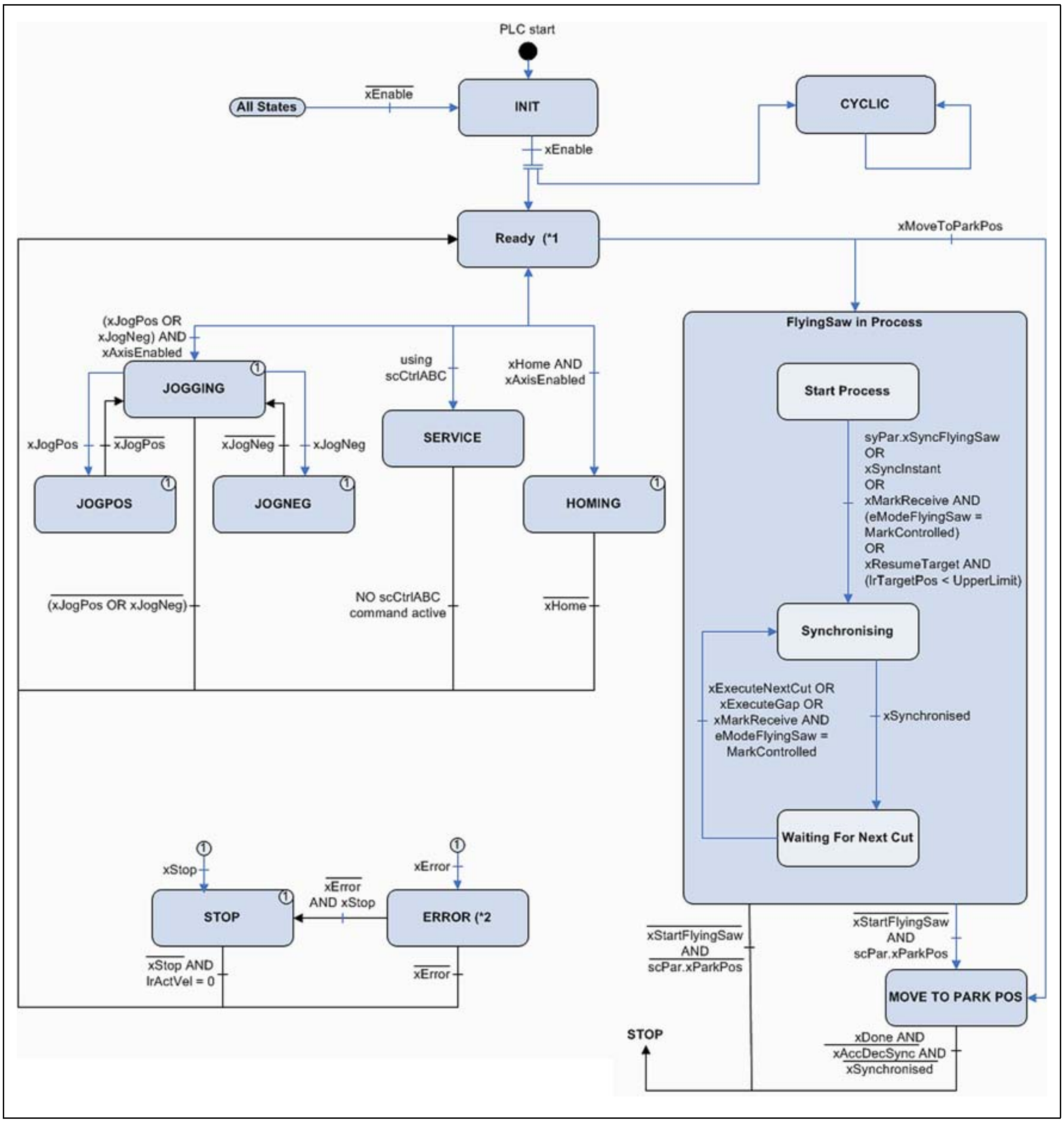
The **L_TT1P_scPar_FlyingSaw[Base/State]** structure contains the parameters of the technology module.

Designator / Data type	Description	Available in version: Base	Available in version: State
lrStopDec LREAL	Deceleration for the stop function and when hardware/software limit switches and the following error monitoring function are triggered • Unit: units/s^2 • Initial value: 10000	●	●
lrStopJerk LREAL	Jerk for the stop function and for the triggering of the hardware limit switches, software limit positions, and the following error monitoring function • Unit: units/s^3 • Initial value: 100000	●	●
lrHaltDec LREAL	Deceleration for the holding function Specification of the maximum speed variation which is to be used for deceleration to standstill. • Unit: units/s^2 • Initial value: 3600 • Only positive values are permissible.	●	●
lrJerk LREAL	Jerk for compensating a clutch or holding function • Unit: units/s^3 • Initial value: 100000	●	●
lrJogJerk LREAL	Jerk for manual jog • Unit: units/s^3 • Initial value: 10000	●	●
lrJogVel LREAL	Maximum speed to be used for manual jog. • Unit: units/s • Initial value: 10	●	●
lrJogAcc LREAL	Acceleration for manual jog Specification of the maximum speed variation which is to be used for acceleration. • Unit: units/s^2 • Initial value: 100	●	●
lrJogDec LREAL	Deceleration for manual jog Specification of the maximum speed variation which is to be used for deceleration to standstill. • Unit: units/s^2 • Initial value: 100	●	●
lrHomePosition LREAL	Home position for a reference run (homing) • Unit: units • Initial value: 0	●	●
xUseHomeExtParameter BOOL	Selection of the homing parameters to be used • Initial value: FALSE FALSE: The homing parameters defined in the axis data are used. TRUE: The **scHomeExtParameter** homing parameters from the application are used.	●	●
scHomeExtParameter L_MC1P_HomeParameter	Homing parameters from the application • Only relevant if xUseHomeExtParameter = TRUE.	●	●

<table>
<tr><th rowspan="2">Designator
Data type</th><th rowspan="2" colspan="2">Description</th><th colspan="2">Available in version</th></tr>
<tr><th>Base</th><th>State</th></tr>
<tr><td>scHomeExtTP
MC_TRIGGER_REF</td><td colspan="2">Transfer of an external touch probe event
• Only relevant for "external encoder" touch probe configuration.
• For describing the MC_TRIGGER_REF structure, see the MC_TouchProbe function block.</td><td>●</td><td>●</td></tr>
<tr><td>lrSyncVel
LREAL</td><td colspan="2">Additive velocity of the saw carriage during synchronisation (travel distance, approach target position again)
The set value is added to the velocity of the master axis.
• Unit: units/s
• Initial value: 3600</td><td>●</td><td>●</td></tr>
<tr><td>lrSyncAcc
LREAL</td><td colspan="2">Maximum acceleration of the saw carriage when clutching in to the master value of the master axis
• Unit: units/s^2
• Initial value: 10000</td><td>●</td><td>●</td></tr>
<tr><td>lrSyncDec
LREAL</td><td colspan="2">Maximum deceleration of the saw carriage when declutching from the master value of the master axis
• Unit: units/s^2
• Initial value: 10000</td><td>●</td><td>●</td></tr>
<tr><td>lrPosVel
LREAL</td><td colspan="2">Maximum velocity of the saw carriage during positioning (travel to the next cutting position, positioning to parking position)
• Unit: units/s
• Initial value: 3600</td><td>●</td><td>●</td></tr>
<tr><td>lrPosAcc
LREAL</td><td colspan="2">Maximum acceleration of the saw carriage during positioning (travel to the next cutting position, positioning to parking position)
• Unit: unit/s^2
• Initial value: 10000</td><td>●</td><td>●</td></tr>
<tr><td>lrPosDec
LREAL</td><td colspan="2">Maximum deceleration of the saw carriage during positioning (travel to the next cutting position, positioning to parking position)
• Unit: units/s^2
• Initial value: 10000</td><td>●</td><td>●</td></tr>
<tr><td>lrSetMaterialPos
LREAL</td><td colspan="2">Default value for the material counter
The value of the material counter results from the value set here plus the actual position of the saw carriage.
The value is accepted when a rising edge (FALSE↗TRUE) is detected a the xSetMaterialPos input.
• Unit: units
• Initial value: 0</td><td>●</td><td>●</td></tr>
<tr><td>lrUpperLimit
LREAL</td><td colspan="2">Upper travel range limit of the saw carriage
• Unit: units
• Initial value: 1000</td><td>●</td><td>●</td></tr>
<tr><td>lrLowerLimit
LREAL</td><td colspan="2">Lower travel range limit of the saw carriage
• Unit: units
• Initial value: 0</td><td>●</td><td>●</td></tr>
<tr><td rowspan="3">xSyncFlyingSaw
BOOL</td><td colspan="2">Response of the "Flying Saw" technology module after switch-on (xStartFlyingSaw input = TRUE)
• Initial value: FALSE</td><td rowspan="3">●</td><td rowspan="3">●</td></tr>
<tr><td>FALSE</td><td>The saw carriage is only synchronised with the material position if the xSynxInInstant input detects a rising edge (FALSE↗TRUE).</td></tr>
<tr><td>TRUE</td><td>The saw carriage is immediately synchronised to the material position when the xStartFlyingSaw is set = TRUE. This can take place independent of the current position of the saw carriage.</td></tr>
</table>

<table>
<tr><th rowspan="2">Designator
Data type</th><th rowspan="2" colspan="2">Description</th><th colspan="2">Available in version</th></tr>
<tr><th>Base</th><th>State</th></tr>
<tr><td>lrGap
LREAL</td><td colspan="2">Distance to be travelled after a saw cut
With length-controlled operation, the lrGap parameter is added to the relative travel way (lrCuttingLength input).
The distance is travelled after a rising edge (FALSE↗TRUE) at the xExecuteGap input.
▸ Length-controlled operation (cutting of relative product lengths) (📖 34)
• Unit: units
• Initial value: 0</td><td>●</td><td>●</td></tr>
<tr><td>lrBladeWidth
LREAL</td><td colspan="2">Blade width of the saw blade
The value is added to the cutting length in the operating mode Length-controlled operation (cutting of relative product lengths) (📖 34).
• Unit: units
• Initial value: 0</td><td>●</td><td>●</td></tr>
<tr><td rowspan="3">xParkPos
BOOL</td><td colspan="2">When the technology module has been switched on (xStartFlyingSaw input = TRUE), the saw carriage is moved to the parking position (lrParkPosition parameter).
• Initial value: FALSE</td><td rowspan="3"></td><td rowspan="3">●</td></tr>
<tr><td>FALSE</td><td>The saw carriage is not moved to the parking position but is brought to a standstill immediately.</td></tr>
<tr><td>TRUE</td><td>The saw carriage is moved to the parking position</td></tr>
<tr><td>lrParkPosition
LREAL</td><td colspan="2">Parking position of the saw carriage if the xParkPos parameter is set to TRUE.
• Unit: units
• Initial value: 0</td><td></td><td>●</td></tr>
<tr><td>lrMarkSensorPos
LREAL</td><td colspan="2">Position of the touch probe sensor (relating to the measuring system of the saw carriage)
• Unit: units</td><td></td><td>●</td></tr>
<tr><td rowspan="3">eModeFlyingSaw
ENUM</td><td colspan="2">Selection of the operating mode
• Initial value: 0 (length-controlled operation)</td><td rowspan="3"></td><td rowspan="3">●</td></tr>
<tr><td>0</td><td>Length-controlled operation (cutting of relative product lengths) (📖 34)</td></tr>
<tr><td>1</td><td>Mark-controlled operation (cutting to position markers) (📖 37)</td></tr>
<tr><td rowspan="2">xPosInWindow
BOOL</td><td colspan="2">Activation of the tolerance window for following error monitoring
• Initial value: FALSE</td><td rowspan="2"></td><td rowspan="2">●</td></tr>
<tr><td>TRUE</td><td>The tolerance window (following error monitoring) is activated.</td></tr>
<tr><td>lrPosInWindow
LREAL</td><td colspan="2">Size of the tolerance window for following error monitoring
• Unit: units
• Initial value: 0</td><td></td><td>●</td></tr>
<tr><td>lrTimePosInWindow
LREAL</td><td colspan="2">Duration of stay of the following error inside the tolerance window
• Unit: ms
• Initial value: 50</td><td></td><td>●</td></tr>
</table>

3.4 State machine



[3-2] State machine of the technology module

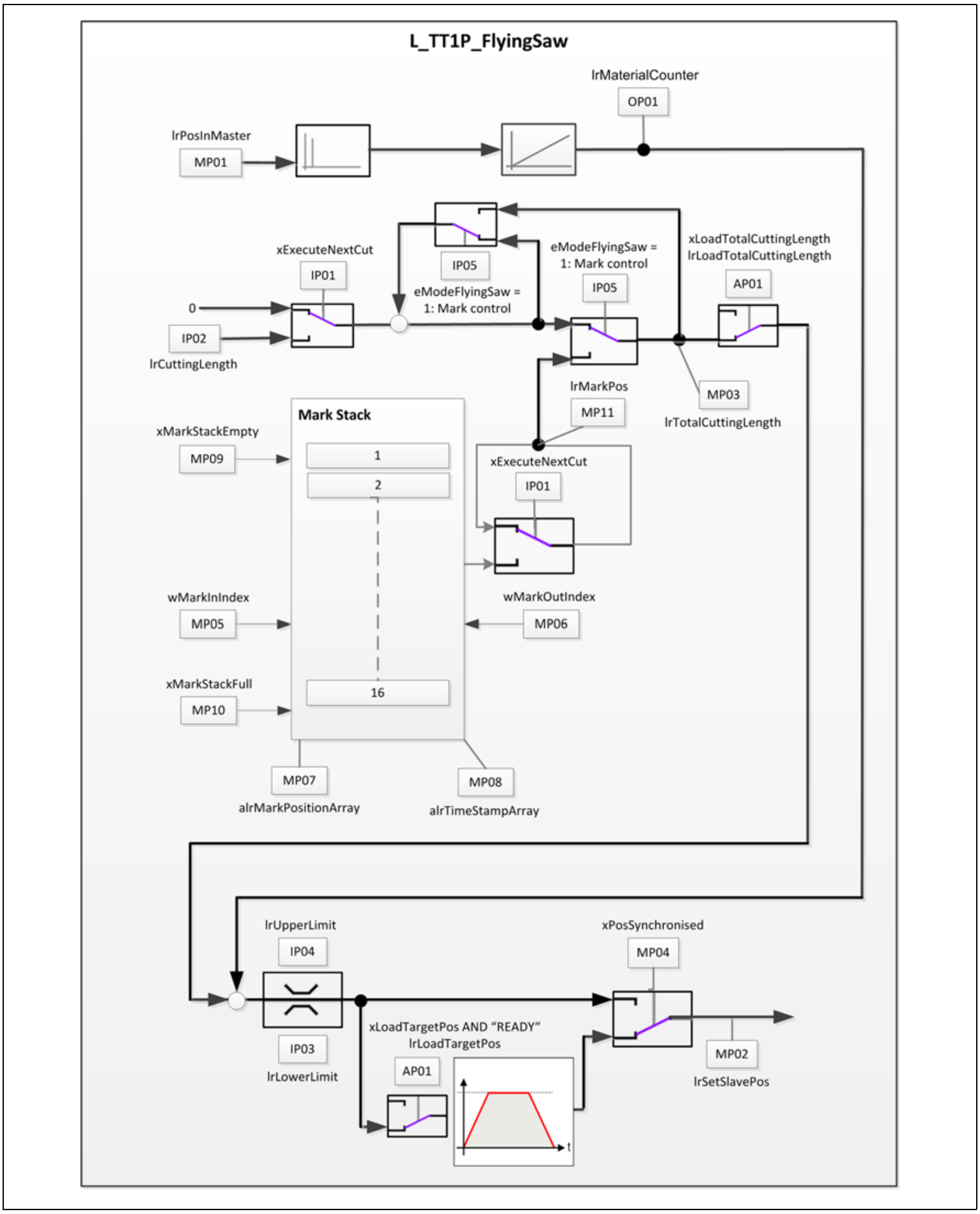
(*1 In the "Ready" state, xRegulatorOn has to be set to TRUE.

(*2 In the "ERROR" state, xResetError has to be set to TRUE in order to acknowledge and reset the errors.

3.5 Signal flow diagrams

The illustrations show the main signal flow of the functions implemented.

The signal flow of the additional functions such as "manual jog" is not displayed here.



[3-3] Signal flow: Flying Saw

3.5.1 Structure of the signal flow

L_TT1P_scSF_FlyingSaw[Base/State]

The contents of the **L_TT1P_scSF_FlyingSaw[Base/State]** structure is read-only and offer a practical diagnostics option within the signal flow (Signal flow diagrams (📖 25)).

Designator Data type	Description		Available in version Base	Available in version State
IP01_xExecuteNextCut BOOL	The input is edge-controlled and evaluates the rising edge.		●	●
	FALSE↗TRUE	The saw carriage moves to the next machining position.		
IP02_lrCuttinglength LREAL	Cutting length (relative positioning) or cutting position (absolute positioning)		●	●
IP03_lrLowerLimit LREAL	Lower travel range limit of the saw carriage • Unit: units • Initial value: 0		●	●
IP04_lrUpperLimit LREAL	Upper travel range limit of the saw carriage • Unit: units • Initial value: 1000		●	●
IP05_eModeFlyingSaw L_TT1P_eModesFlyingSaw	Selection of the operating mode • Initial value: 0 (length-controlled operation)			●
	0	Length-controlled operation (cutting of relative product lengths) (📖 34)		
	1	Mark-controlled operation (cutting to position markers) (📖 37)		
MP01_lrSetMasterPos LREAL	Set position of the master axis • Unit: units		●	●
MP02_lrSetSlavePos LREAL	Set position of the slave axis • Unit: units		●	●
MP03_lrTotalCuttingLength LREAL	Sum of the cut cutting lengths • Unit: units		●	●
MP04_xPosSynchronised BOOL	TRUE	The setpoints of the master axis and the slave axis are synchronous.	●	●
MP05_wMarkInIndex WORD	Index of the incoming mark (touch probe) on the mark stack			●
MP06_wMarkOutIndex WORD	Index of the mark (touch probe) to be called on the mark stack			●
MP07_alrMarkPositionArray ARRAY [0..15] OF LREAL	Memory of the detected position markers			●
MP08_alrTimeStampArray ARRAY [0..15] OF LREAL	Memory of the time stamps corresponding to the position markers in MP07_alrMarkPositionArray			●
MP09_xMarkStackEmpty BOOL	TRUE	There is no valid mark on the mark stack anymore.		●
MP10_xMarkStackFull BOOL	TRUE	The mark stack is completely filled. The next mark detected overwrites the oldest mark entered on the mark stack.		●
MP11_lrStackMarkPos LREAL	Position of the mark currently output by the mark stack. • Unit: units			●
OP01_lrMaterialCounter LREAL	Continuous material position • Unit: units		●	●
OP02_lrTargetPos LREAL	Target position of the saw carriage • Unit: units		●	●

3.5.2 Structure of the access points

L_TT1P_scAP_FlyingSaw[Base/State]

The access points (AP) can be used to influence signals. In the initial state, the access points do not have any effect.

Each access point acts as an alternative branch and is activated via an OR operation or a switch.

Designator	Data type	Description		Available in version: Base	Available in version: State
AP01_xLoadTotalCutting Length	BOOL	Selection of a value for the sum of all lengths cut (MP03_lrTotalCuttingLength variable) • Initial value: TRUE		●	●
		TRUE	The value in AP01_lrLoadTotalCuttingLength is preselected.		
AP01_lrLoadTotalCutting Length	LREAL	Value for the sum of all lengths cut (MP03_lrTotalCuttingLength variable) • Unit: units • Initial value: 0			
AP02_xLoadTargetPos	BOOL	Selection of a value for the target position of the saw carriage (lrTargetPos or OP02_lrTargetPos output) The precondition for this is that the technology module is in the "READY" state. • Initial value: TRUE		●	●
		TRUE	The value in lrLoadTargetPos is preselected.		
AP02_lrLoadTargetPos	LREAL	Value for the target position of the saw carriage (lrTargetPos or OP02_lrTargetPos output) • Unit: units • Initial value: 0			
AP03_xRangeOfTolerance	BOOL	Activation of the tolerance range set under AP03_lrRangeOfTolerance			●
		TRUE	Tolerance range AP03_lrRangeOfTolerance is activated.		
AP03_lrRangeOfTolerance	LREAL	Tolerance range which is put around the master value of the master axis to ensure the synchronism with the saw carriage. This is required in case of noisy master value signals.			

3.6 Manual jog (jogging)

Precondition

- The technology module is in the "Ready" state.
- The slave axis is enabled (*xRegulatorOn* = TRUE).

Execution

For manual jog of the axis, the manual jog speed *lrJogVel* is used.

If the *xJogPos* input is TRUE, the axis is traversed in positive direction and if the *xJogNeg* input is TRUE, the axis is traversed in negative direction. The axis is executed for as long as the input remains set to TRUE.

The current travel command cannot be replaced by another jog command. Only if both inputs have been reset, the State machine (📖 24) changes to the "Ready" state again.

Parameters to be set

The parameters for the manual jog are located in the L_TT1P_scPar_FlyingSaw[Base/State] (📖 21) parameter structure.

```
lrJogVel : LREAL := 10;       // Velocity [units/s]
lrJogAcc : LREAL := 100;      // Acceleration [units/s^2]
lrJogDec : LREAL := 100;      // Deceleration [units/s^2]
lrJogJerk : LREAL := 10000;  // Jerk [units/s^3]
```

The parameter values can be changed during operation. They are accepted when the *xJogPos* or *xJogNeg* input is set to TRUE again.

3.7 Homing

Precondition

- The technology module is in the "Ready" state.
- The slave axis is enabled (*xRegulatorOn* = TRUE).

Execution

Homing is started with a rising edge (FALSE↗TRUE) at the *xHomeExecute* input. The axis will be travelling until the home position is reached. After successful homing, the State machine (📖 24) changes back again to the "Ready" state.

The homing process is not interrupted if the *xHomeExecute* input is set to FALSE too early. The function is aborted via the *xStop* input.

Parameters to be set

The parameters for homing are located in the L_TT1P_scPar_FlyingSaw[Base/State] (📖 21) parameter structure.

```
xUseHomeExtParameter : BOOL := FALSE;
lrHomePosition : LREAL := 0.0;
scHomeExtParameter : L_MC1P_HomeParameter;
scHomeExtTP : MC_TRIGGER_REF;
```

3.8 Setting motion profiles (linear/S-ramps)

Parameters to be set

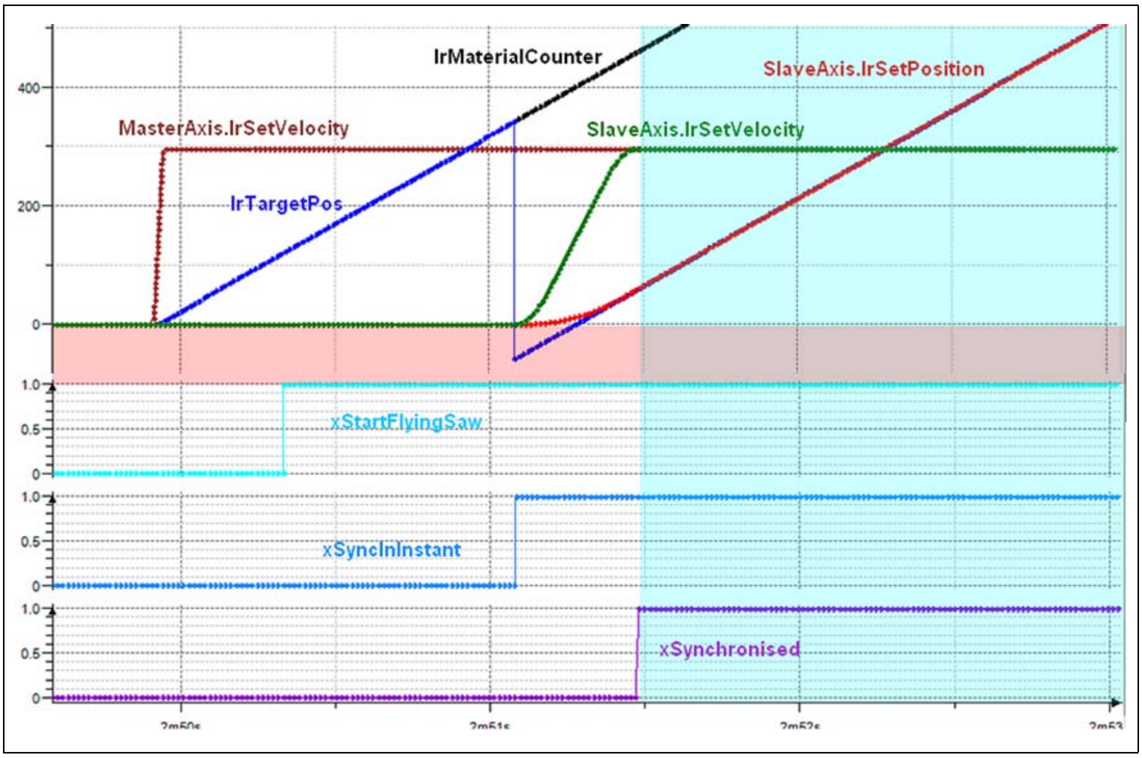
The parameters for the motion profile of the saw carriage are located in the L_TT1P_scPar_FlyingSaw[Base/State] (21) parameter structure.

```
lrJerk : LREAL := 100000; // units/s3;
lrSyncVel : LREAL := 3600; // units/s;
lrSyncAcc : LREAL := 10000; // units/s2;
lrSyncDec : LREAL := 10000; // units/s2;
lrPosVel : LREAL := 3600; // units/s;
lrPosAcc : LREAL := 10000; // units/s2;
lrPosDec : LREAL := 10000; // units/s2;
```

3.9 First synchronisation of the saw carriage to the material speed

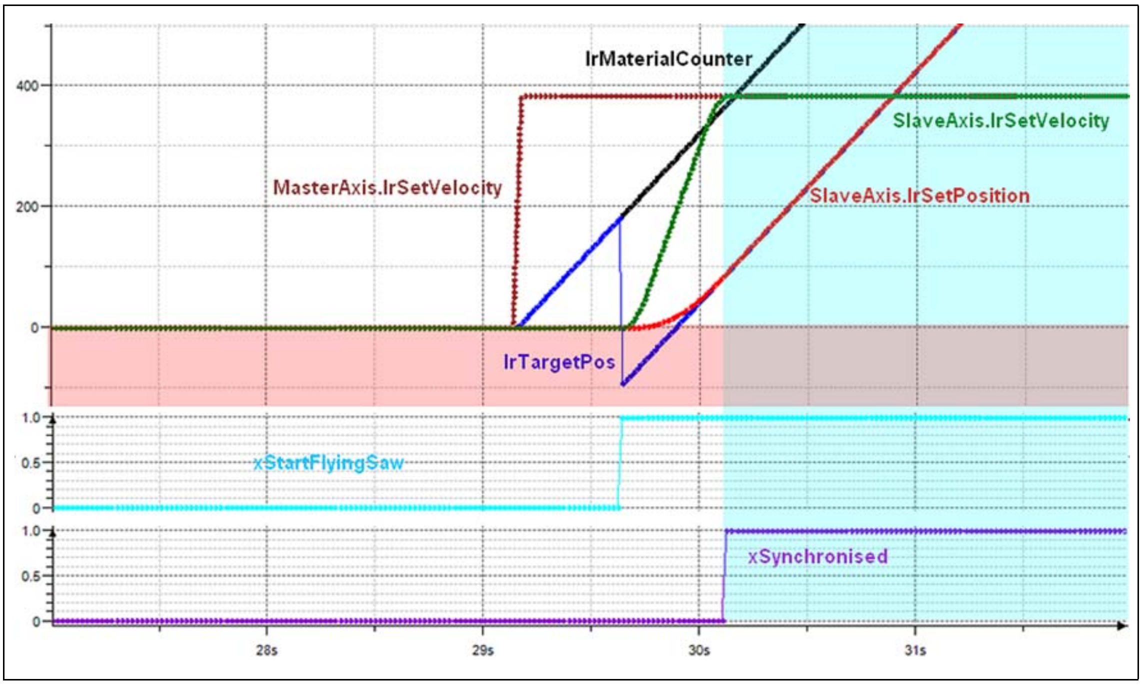
When the "flying saw" has been switched on by the *xStartFlyingSaw* input = FALSE↗TRUE, there are several possibilities of synchronising the saw carriage with the material speed:

A. *xSyncFlyingSaw* parameter = FALSE:
   The saw carriage is only synchronised with the material position if the *xSynxInInstant* input detects a rising edge (FALSE↗TRUE).
   This function can for example be used to carry out an initial top cut.
   The initial top cut at the start of length-controlled operation can either be carried out when the material web is at a standstill or if it is running.



[3-4] Example: Characteristic with xSyncFlyingSaw parameter = FALSE

B. *xSyncFlyingSaw* parameter = TRUE:
The saw carriage is synchronised with the material position immediately if the *xStartFlyingSaw* input is set to TRUE. This can take place irrespective of the current saw carriage position.



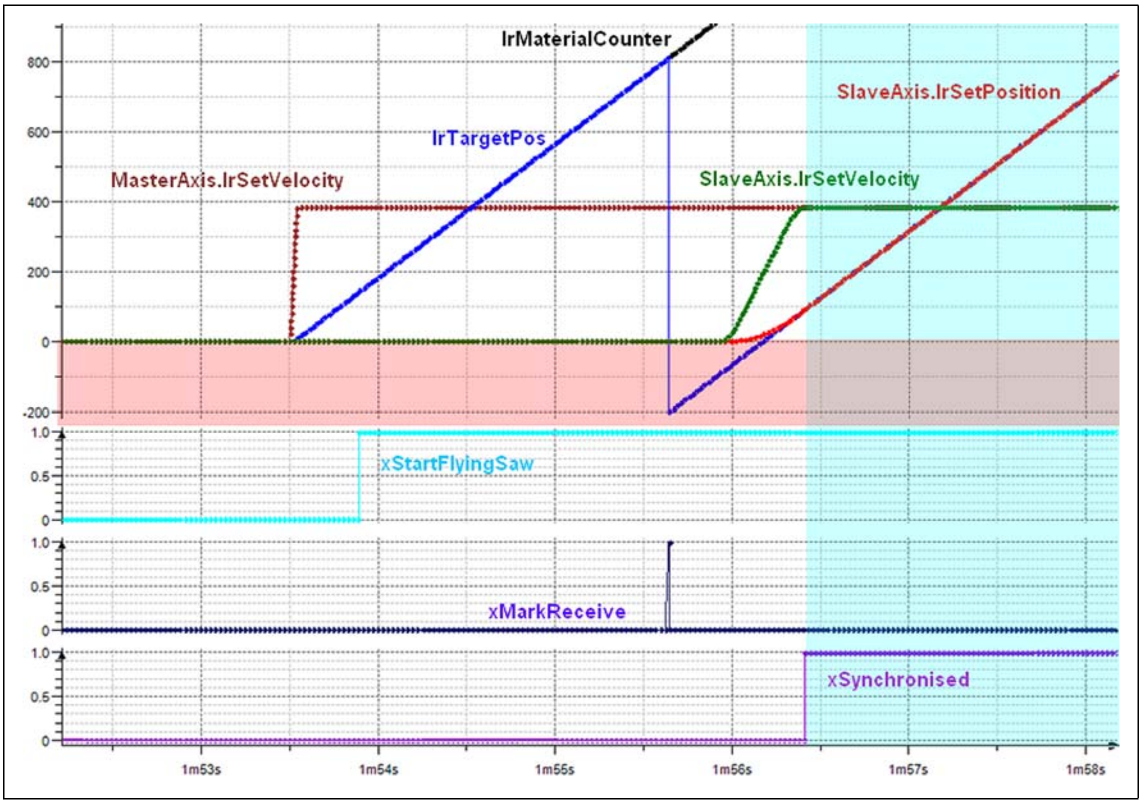
[3-5] Example: Characteristic with xSyncFlyingSaw parameter = TRUE

C. *eModeFlyingSaw* = 1 parameter (mark-controlled operation):

Note!

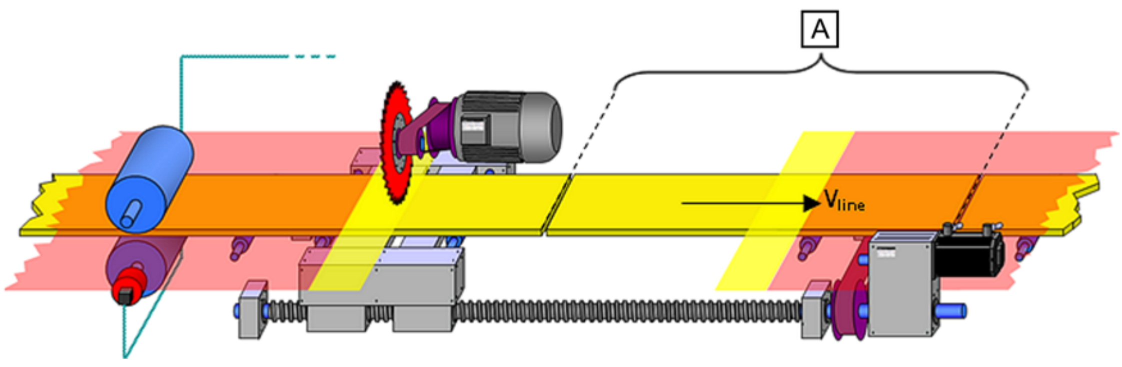
The mark-controlled operation is only possible in the **State version** of the technology module.

The saw carriage is only synchronised with the material position if a mark has been detected at the mark sensor connected (*xMarkReceive* input = TRUE).



[3-6] Example: Characteristic with eModeFlyingSaw parameter = 1 (mark-controlled operation)

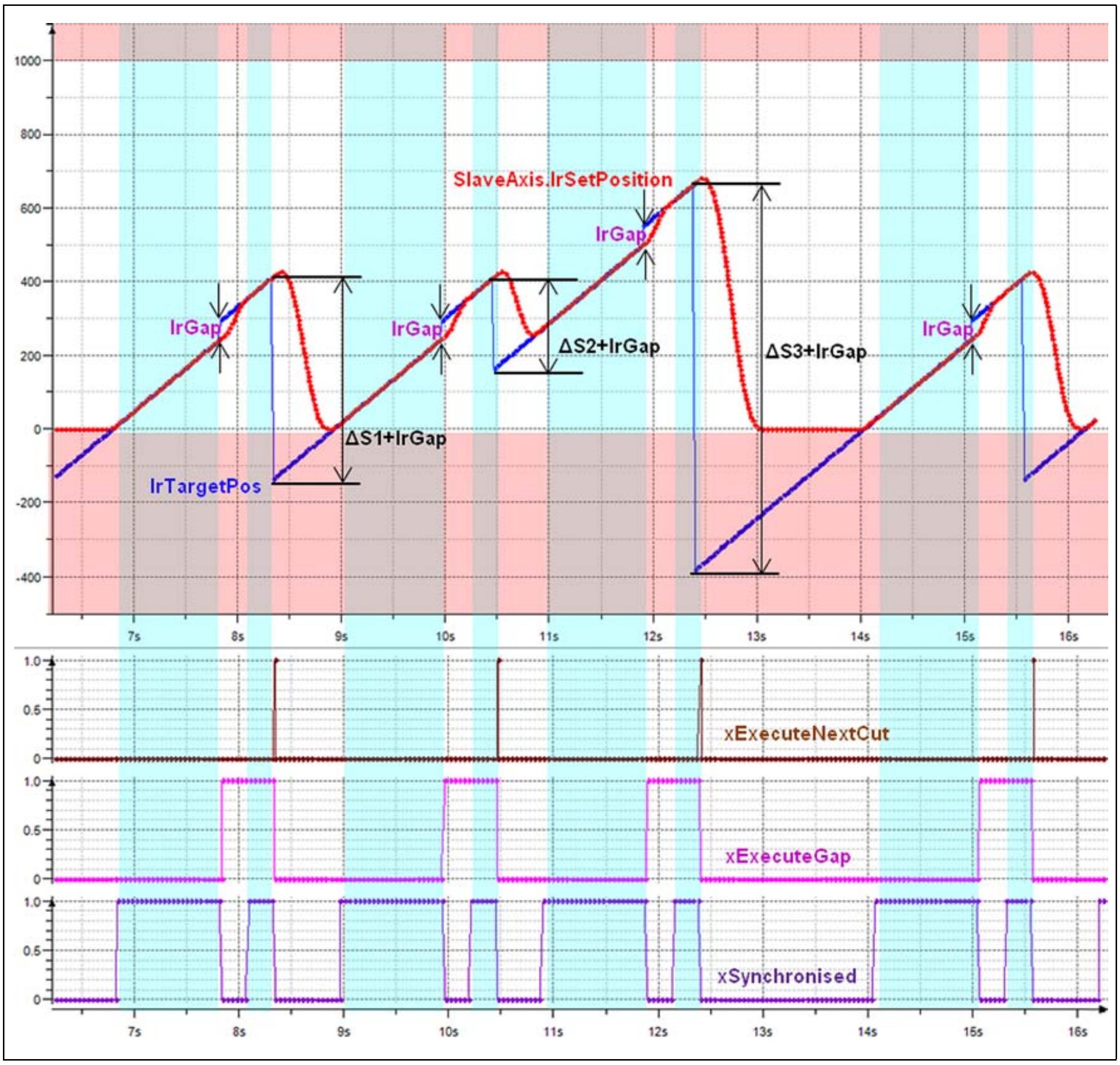
3.10 Length-controlled operation (cutting of relative product lengths)



A Cutting length

[3-7] Cutting length with length-controlled operation

In length-controlled operation, the first machining process (top cut) starts in any position. Then all subsequent saw cuts have the desired relative distance (cutting length) to the previous cut in each case.



[3-8] Example: Characteristic of length-controlled operation

In the time diagram in Fig. [3-8] three different saw cuts ΔS1 ... ΔS3 are carried out. After each cut a specific distance (*lrGap* parameter) is run.

The movement from one machining position to the next one is carried out via a superimposed relative positioning process. The positioning distance ΔS1 ... ΔS3 results from the cutting length at the *lrCuttingLength* input added by the length of the distance run in the *lrGap* parameter. The relative positioning process is initiated by a rising edge (FALSE↗TRUE) at the *xExecuteNextCut* input.

The *lrLowerLimit* and *lrUpperLimit* parameters are used for setting the machining area of the saw carriage. In the diagram above, the area outside these two limits is highlighted in red. When saw cut ΔS3 is requested, it can be clearly seen that the target position (*lrTargetPos* output) is below the *lrLowerLimit* limit. In this case, the saw carriage moves to the lower limit specified and waits until the target position enters the operating range so that synchronisation with the target position can then take place.

The areas highlighted in turquoise indicate when the saw carriage is synchronous to the material position. In the example, the saw carriage is only synchronised to the material position when a new target position is requested with the *xExecuteNextCut* input = TRUE or if a distance (*lrGap* parameter) is travelled after the saw cut with the *xExecuteGap* input = TRUE.

Parameters to be set

The parameters for the length-controlled operation are located in the L_TT1P_scPar_FlyingSaw[Base/State] (📖 21) parameter structure.

A new/changed cutting length with a rising edge (FALSE↗TRUE) is accepted at the *xExecuteNextCut* input.

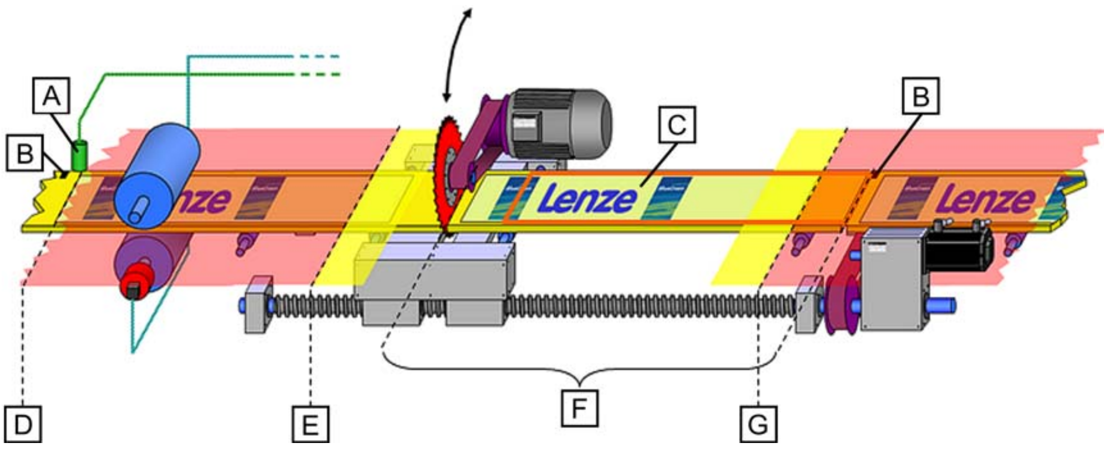
For the time diagram in figure [3-8] (📖 35), a sequence runs outside the technology module that successively defines the following product lengths at the *lrCuttingLength* input:

- 500 [units]
- 250 [units]
- 1000 [units]

Furthermore the following parameter settings have been carried out:

```
lrLowerLimit : LREAL := 0;
lrUpperLimit : LREAL := 1000;
lrGap : LREAL := 50;
xSyncFlyingSaw : BOOL := TRUE;
```

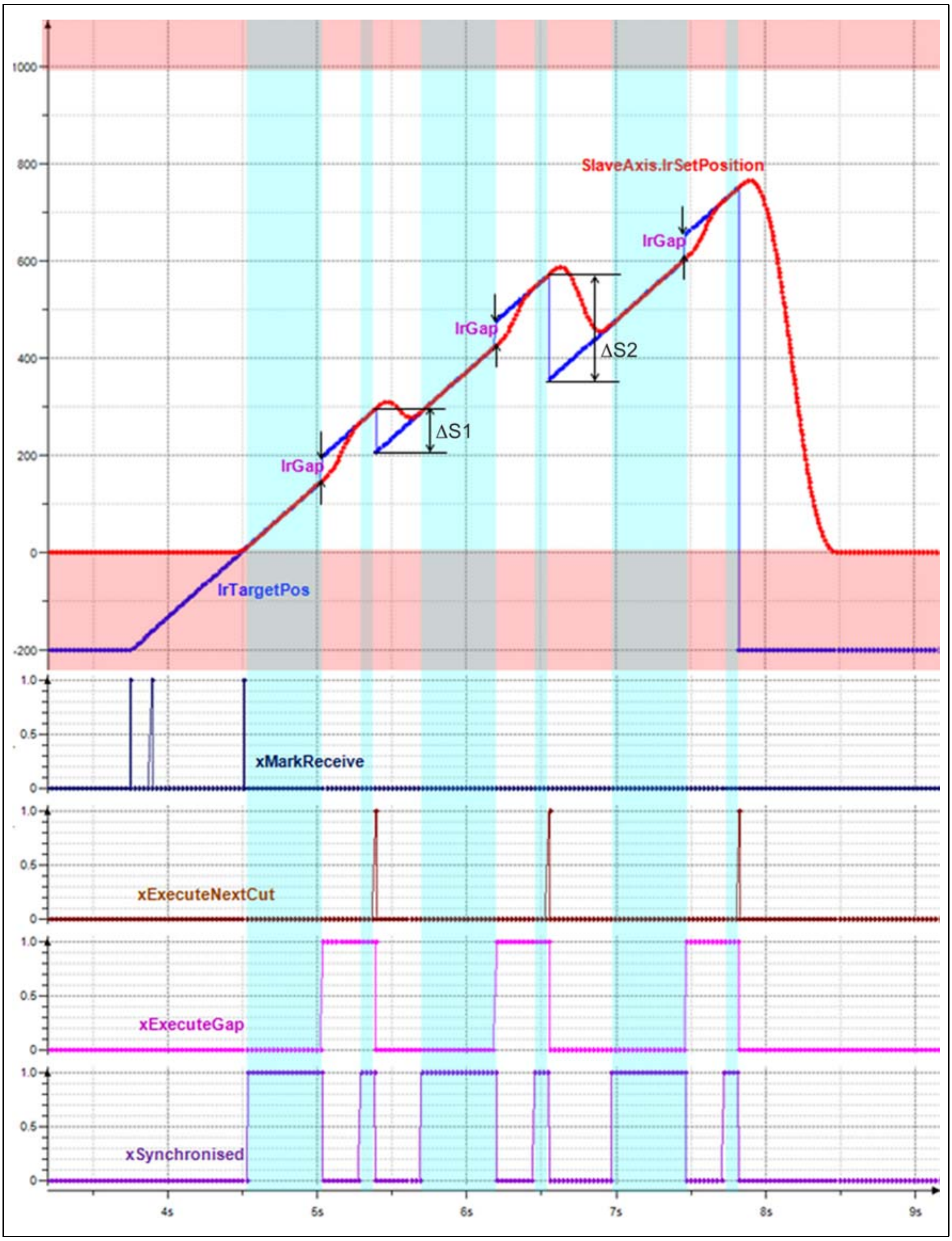
3.11 Mark-controlled operation (cutting to position markers)



A Mark sensor (touch probe sensor)
B Cutting mark
C Print image
D Position of the mark sensor (lrMarkSensorPos parameter)
E Lower travel range limit of the saw carriage (lrLowerLimit parameter)
F Cutting length (lrSetMaterialPos parameter)
G Upper travel range limit of the saw carriage (lrUpperLimit parameter)

[3-9] Cutting length for mark-controlled operation

Mark-controlled operation is used if the material already features partitions by position markers, e.g. in the case of a print image. In this case, the saw cuts must not be placed arbitrarily on the material, but must be executed in line with the position markers. An additional mark sensor that is placed in a sufficient distance in front of the operating range of the saw detects the cutting positions beforehand. These cutting positions are stored in a mark stack.



[3-10] Example: Characteristic of mark-controlled operation

In the time diagram in fig. [3-10] three marks are detected in a row. The first mark starts the process. During the machining phase (cutting and running the distance) of the first mark detected, two further marks are detected which are stored on the mark stack. The next valid mark position is approached with a rising edge (FALSE↗TRUE) at the *xExecuteNextCut* input = TRUE.

if there is no valid mark on the mark stack anymore, the saw carriage moves to the lower travel range limit (*lrLowerLimit* parameter) and waits for the next mark.

The cutting length is the distance between the detected marks. As soon as the technology module is switched on, the position at the *lrActMarkPosIn* input is converted to the setpoint position of the saw carriage for each detected mark at the *xReceiveMark* input and stored on the mark stack. With a rising edge (FALSE↗TRUE) at the *xExecuteNextCut* input, it is possible to change over to the next mark in the stack.

Parameters to be set

The parameters for the length-controlled operation are located in the L_TT1P_scPar_FlyingSaw[Base/State] (📖 21) parameter structure.

For the time diagram in figure [3-10] (📖 38), the following parameters have been set:

```
eModeFlyingSaw : ENUM : = 1;  // Mark controlled
lrSetMaterialPos : LREAL := 0;
lrLowerLimit : LREAL := 0;
lrUpperLimit : LREAL := 1000;
lrGap : LREAL := 50;
xSyncFlyingSaw : BOOL := FALSE;
```

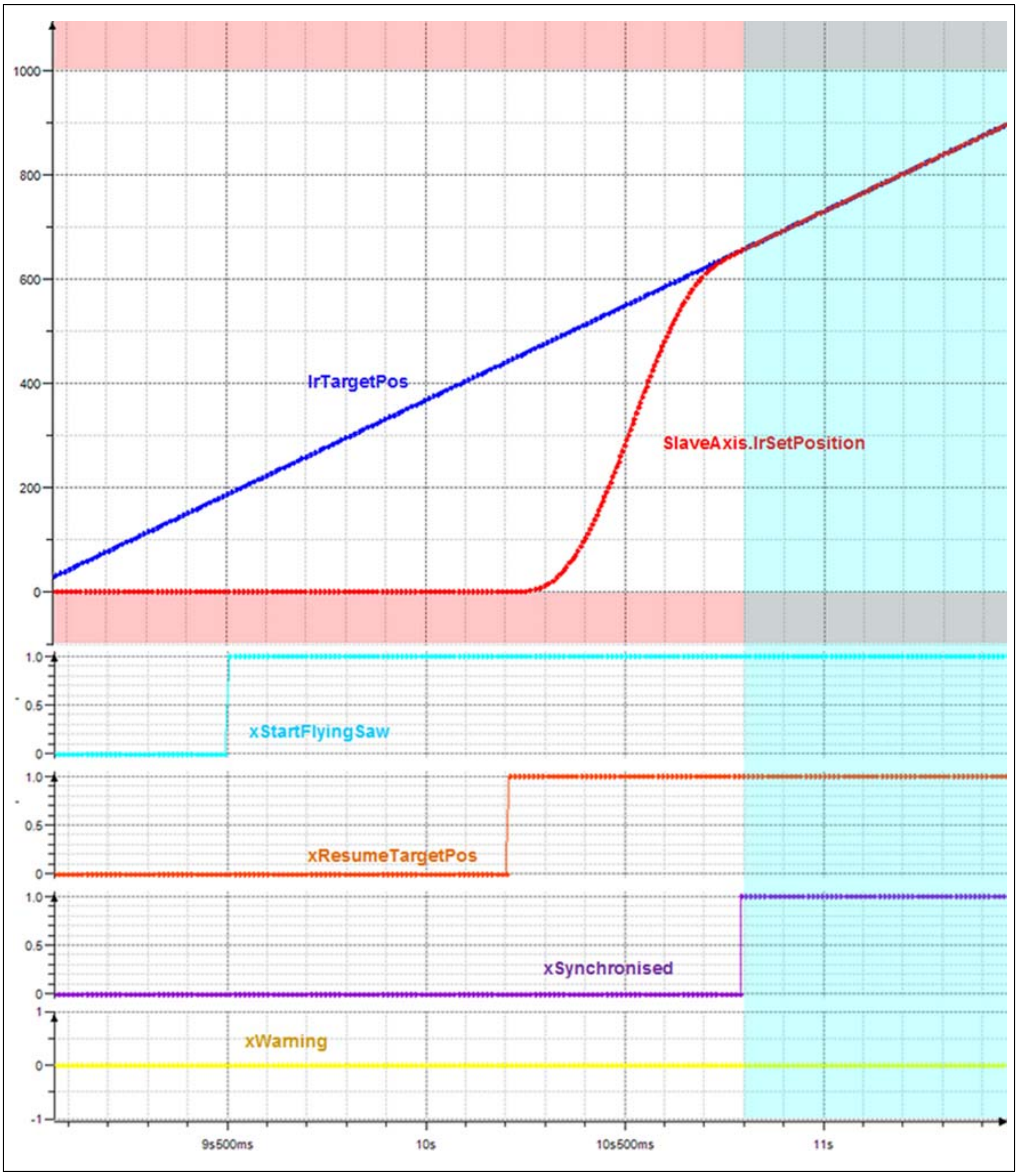
3.12 Restoring the target position

This function can be executed once when the sawing process has been started (*xStartFlyingSaw* = TRUE) in order to restore a target position.

In length-controlled operation and in mark-controlled operation, the process for restoring the target position is different. Furthermore it is checked in both operating modes whether the target position requested is below the upper travel range limit (*lrUpperLimit* parameter).

Length-controlled operation

After a rising edge (FALSE↗TRUE) at the *xResumeTargetPos* input, the target position (*lrTargetPos* output) is approached again.



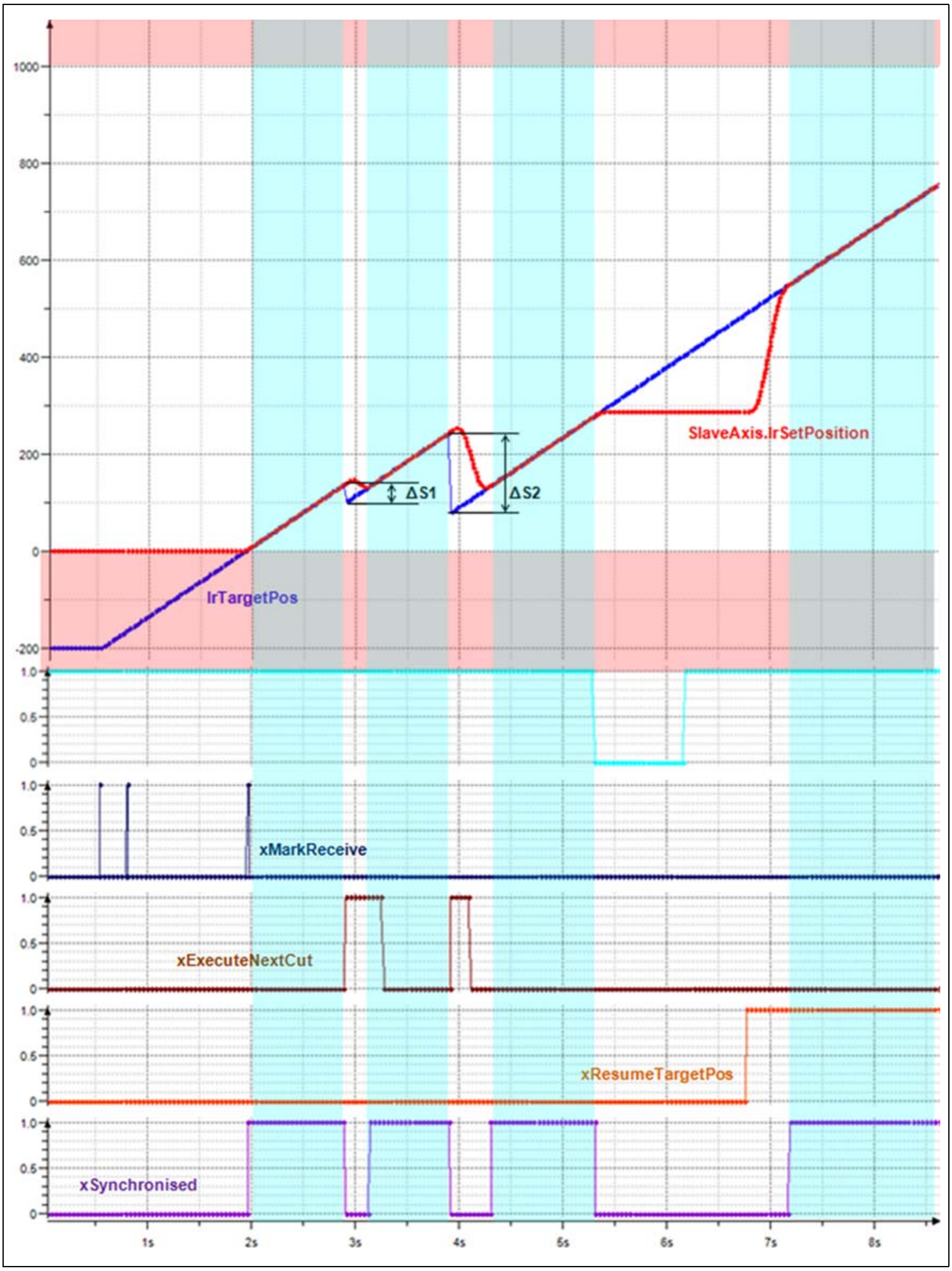
[3-11] Example: Characteristic for "Restoring the target position" in length-controlled operation

Mark-controlled operation

In case of mark-controlled operation, only one valid mark can be restored. Here, the target position (*lrTargetPos* output) has to be in front of the travel range limit (*lrUpperLimit* parameter).

If the target position has already exceeded the upper travel range limit, a rising edge (FALSE↗TRUE) at the *xExecuteNextCut* input serves to change over to the next mark in the mark stack. If no valid mark is available in the stack, no target position is approached.

After a rising edge (FALSE↗TRUE) at the *xResumeTargetPos* input, the target position is approached again.



[3-12] Example: Characteristic for "Restoring the target position" in mark-controlled operation

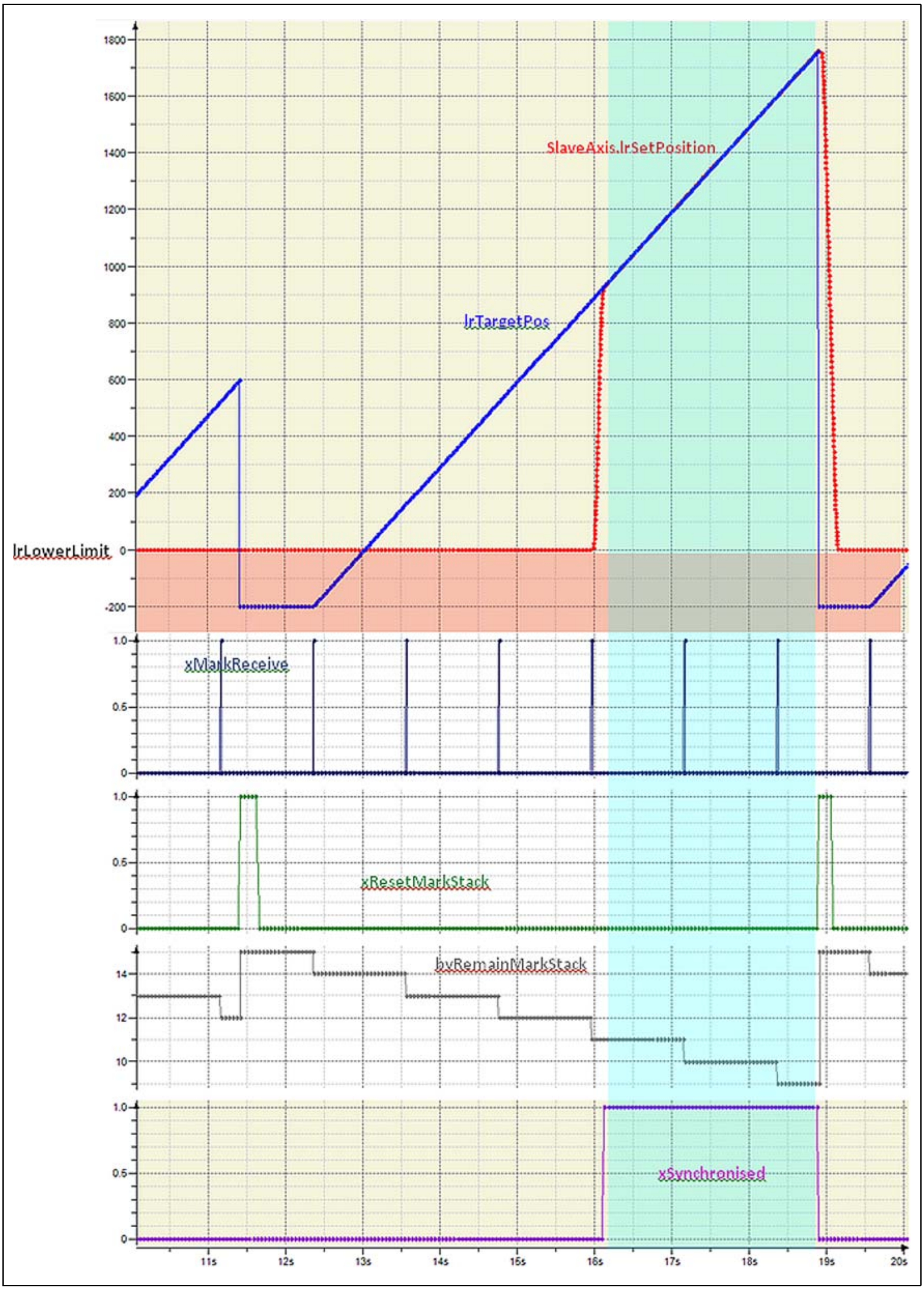
3.13 Deleting the mark stack

With a rising edge (FALSE↗TRUE) at the *xResetMarkStack* input, all marks stored on the mark stack are deleted.

The *byRemainMarkStack* output shows how many marks the mark stack can take.

If the function is executed while the "flying saw" is in the synchronous state, (*xStartFlyingSaw* input = TRUE), the saw carriage travels to the waiting position (*lrLowerLimit* parameter) and will be synchronised with the next mark.

Both options are shown in the time diagram in figure [3-13] (📖 44).



[3-13] Example: "Delete mark stack" history

3.14 Moving to the parking position

There are two options to approach the parking position:

A. "Flying Saw" is not active:
   - The technology module is in the "Ready" state.
   - A rising edge (FALSE↗TRUE) at the *xMoveToParkPos* input causes a change from "Ready" to "Move To Park Pos" and the saw carriage travels to the parking position set in the *lrParkPosition* parameter.

B. "Flying Saw" is deactivated:
   - The parameters *lrParkPosition* and *xParkPos* from the L_TT1P_scPar_FlyingSaw[Base/State] (📖 21) parameter structure are set to TRUE.
   - If then the *xStartFlyingSaw* input is set from TRUE to FALSE, the saw carriage travels to the parking position set in the *lrParkPosition* parameter.

During positioning, the *xPositioning*output is = TRUE.

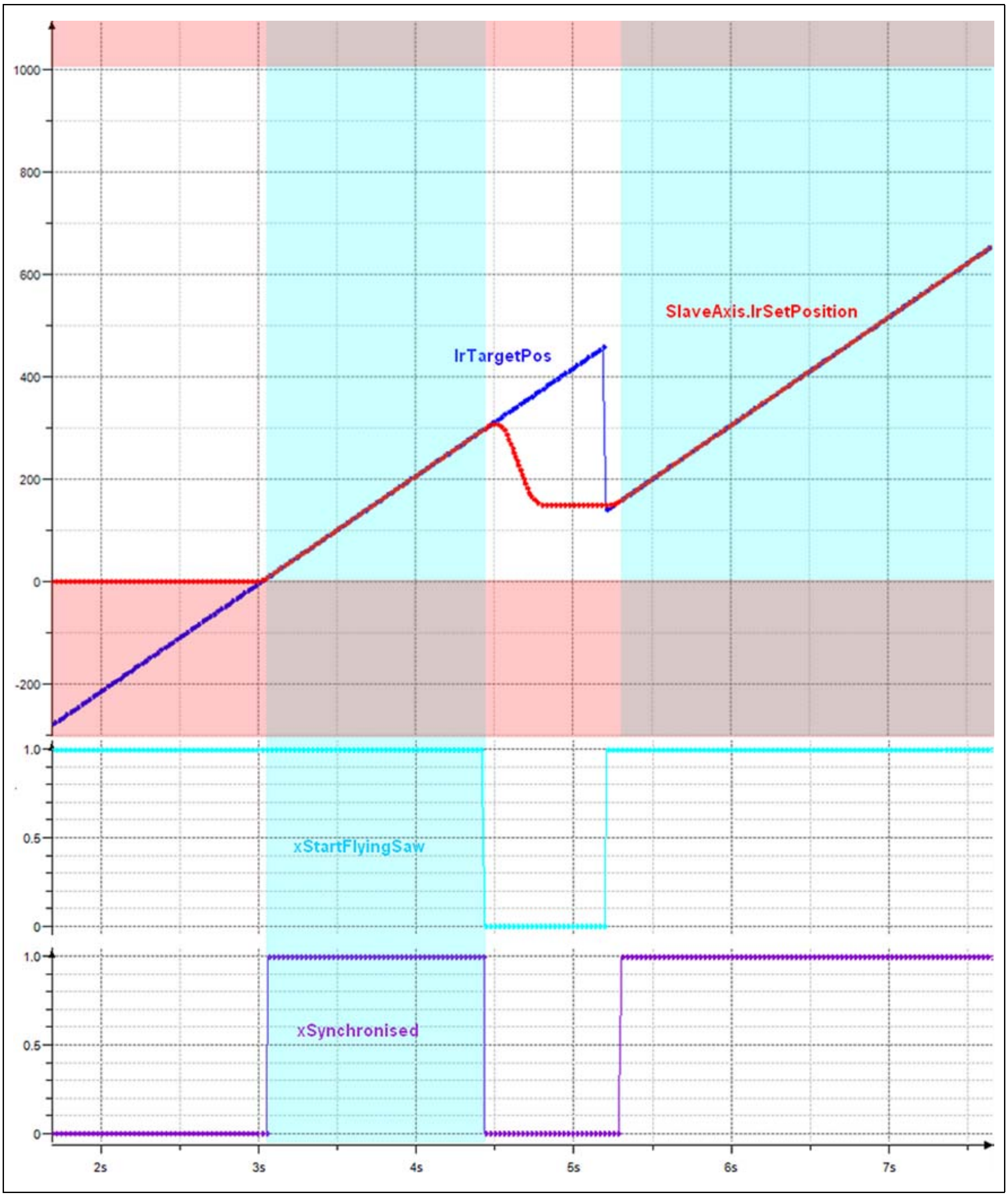
When the parking position has been reached, the output *xInPosition* is set = TRUE and the output *xPositioning* is set = FALSE.

Parameters to be set

The parameters for travelling to the parking position are located in the L_TT1P_scPar_FlyingSaw[Base/State] (📖 21) parameter structure.

For the time diagram in figure [3-14] (📖 46), the following parameters have been set:

```
eModeFlyingSaw : ENUM : = 0;  // Length controlled
xSyncFlyingSaw : BOOL := TRUE;
xParkPos : BOOL := TRUE;
lrParkPosition : LREAL := 150;  // units
```

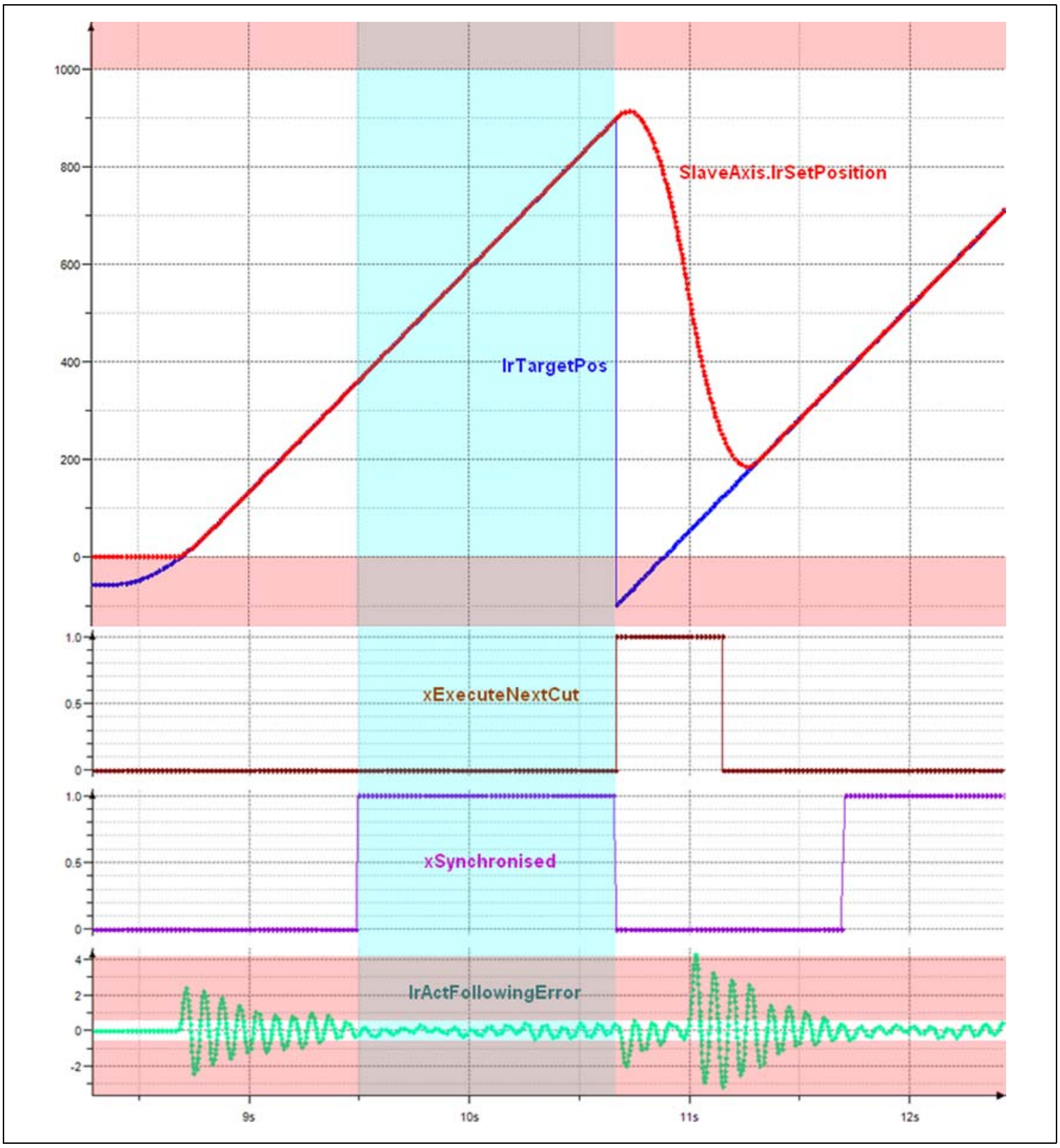


[3-14] Example: Characteristic for "Moving to parking position"

3.15 Window for following error monitoring

This function checks whether the following error is to be found within a defined window within a certain period. It is activated with the *xPosInWindow* parameter set to TRUE.

The window and the time are set via the parameters *lrPosInWindow* and *lrTimePosInWindow* from the L_TT1P_scPar_FlyingSaw[Base/State] (📖 21) parameter structure. As soon as the two setpoints of the master and slave axis are synchronous and the following error is inside the set window, the *xSynchronised* output is set = TRUE.



[3-15] Example: Characteristic of the following error monitoring

Parameters to be set

The parameters for the following error monitoring are located in the L_TT1P_scPar_FlyingSaw[Base/State] (21) parameter structure.

For the time diagram in figure [3-15] (47), the following parameters have been set:

```
eModeFlyingSaw : ENUM : = 0;  // Length controlled
xSyncFlyingSaw : BOOL := TRUE;
xPosInWindow : BOOL := TRUE;
lrPosInWindow : LREAL := 0.5;  // units
lrTimePosInWindow : LREAL := 50;  // ms
```

3.16 CPU utilisation (example Controller 3231 C)

The following table shows the CPU utilisation in microseconds using the example of the 3231 C controller (ATOM™ processor, 1.6 GHz).

Versions	Interconnection of the technology module	CPU utilisation	
		Average	Maximum peak

Base	xEnable := TRUE;
	xRegulatorOn := TRUE;
	xStartFlyingSaw := TRUE;
	xExecuteNextCut := TRUE;
		40 µs	90 µs

State	xEnable := TRUE;
	xRegulatorOn := TRUE;
	xStartFlyingSaw := TRUE;
	xExecuteNextCut := TRUE;
		80 µs	120 µs

Index

FEEDBACK

Your opinion is important to us

These instructions were created to the best of our knowledge and belief to give you the best possible support for handling our product.

Perhaps we have not succeeded in achieving this objective in every respect. If you have suggestions for improvement, please e-mail us to:

feedback-docu@lenze.com

Thank you very much for your support.

Your Lenze documentation team

Lenze Automation GmbH
Postfach 10 13 52, 31763 Hameln
Hans-Lenze-Straße 1, 31855 Aerzen
GERMANY
HR Hannover B 205381
+49 5154 82-0
+49 5154 82-2800
lenze@lenze.com
www.lenze.com

Service
Lenze Service GmbH
Breslauer Straße 3, 32699 Extertal
GERMANY
008000 24 46877 (24 h helpline)
+49 5154 82-1112
service@lenze.com