TM441

Motion control: Electronic gears and cam profiles

PERFECTION IN AUTOMATION
A MEMBER OF THE ABB GROUP

Prerequisites and requirements

Training modules	TM440 – Motion Control: Basic Functions
Software	Automation Studio 4.3.3 Automation Runtime 4.33 ACP10/ARNC0 Technology Package 5.0 mapp Technology Package 1.63 mapp Services Technology Package 5.0 Automation Help Upgrade 4.3.4
Hardware	X20 controller + ACOPOS / ACOPOSmulti / ACOPOSmicro or Simulation

Table of contents

1 Introduction

The B&R drive solution provides flexible, high-performance tools for coupling drives electronically. This makes it possible, for example, to create synchronous drives that are coupled for linear as well as for dynamic movements. There are many practical applications for these solutions, such as synchronous cutting procedures, dynamic transfer processes and flexible length allocation.

Easy-to-use mapp components – supported in the background by the ACP10_MC library – are available for comprehensive operation of these functions.

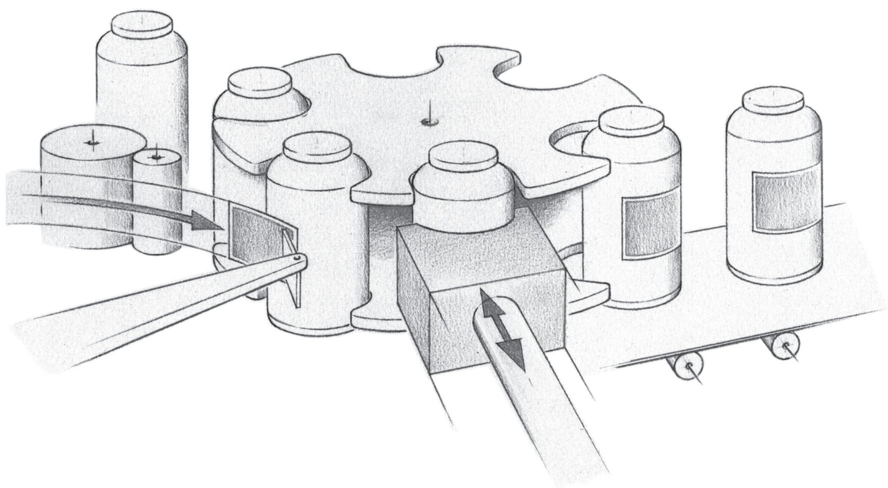


Figure 1: Labeling bottles

This training module explains how to use a range of functions to configure and control electronically coupled movement sequences.

We will begin with a brief overview to become familiar with the individual options. We will then cover some basic theory, ask ourselves some important questions and finally, learn how to use multi-axis functions in actual applications.

1.1 Learning objectives

This training module uses selected examples to demonstrate the use of PLCopen-compliant multi-axis functions.

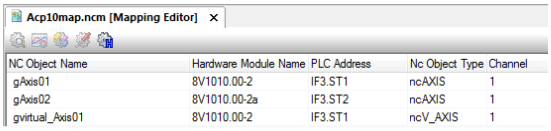
- You will learn about the possibilities of the mapp technology multi-axis components.
- You will learn how to connect drives using simple axis couplings and how to implement a phase shift.
- You will learn about the concept of cam profiles and virtual axes. You will be able to configure them in Automation Studio and use them for an axis coupling.
- You will be able to plot the course of movement of master and slave axes and will understand the relationships.
- You will learn the underlying principles behind the Cam Profile Automat.
- You will learn about the various compensation gears that can be used when switching cam profiles.
- You will learn how to prepare an axis coupling for operation and become acquainted with the process of error evaluation.
- You will be familiar with the Automation Studio help system and will be able to navigate the ACOPOS reference manual.

2 Preparing the coupling axes

Some requirements must be met for the exercises in this training module. Two single axes – a master axis and a slave axis – and a virtual axis are required. The axes are configured using the Drive Configuration wizard and added to the NC mapping table. Using the MpAxisBasic component, they are prepared to the extent that the driver controller can be enabled, the homing procedure started and one of the basic movements performed. It's also recommended to prepare the option for acknowledging driver errors.

The initialization of all axes should be carried out using the mapp configuration in the Configuration View. An axis period always has to be configured for the real axes.

Another virtual axis that is used as a setpoint source for one or more slave axes can also be configured.


Acp10map.ncm [Mapping Editor]

NC Object Name	Hardware Module Name	PLC Address	Nc Object Type	Channel
gAxis01	8V1010.00-2	IF3.ST1	ncAXIS	1
gAxis02	8V1010.00-2a	IF3.ST2	ncAXIS	1
gvirtual_Axis01	8V1010.00-2	IF3.ST1	ncV_AXIS	1

Figure 2: NC objects for master and slave axes as well as a virtual axis in the NC mapping table

> The configuration of the virtual axis is enabled as soon as the drive is added in the Drive Configuration wizard. Depending on the module, up to two real axes and two virtual axes are provided.

Configuration of the master and slave axes in the Configuration View

The master and slave axes are configured as periodic axes for the completion of exercises in this training module. The following configuration settings should be made:

Axis period for the master axis: 3600 units

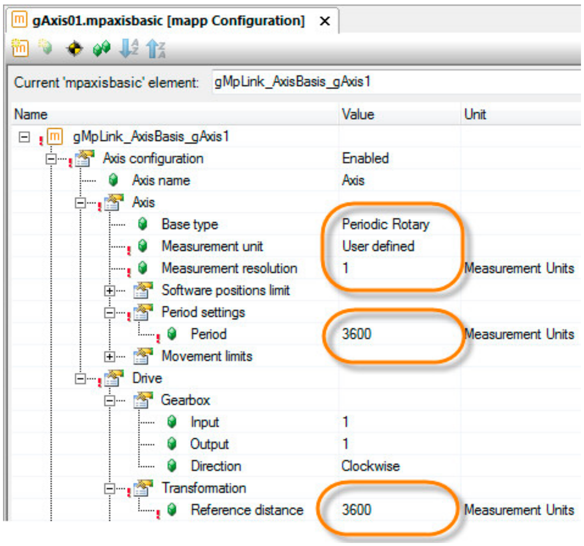


Figure 3: mapp axis configuration for the master axis

Axis period for the slave axis: 3600 units

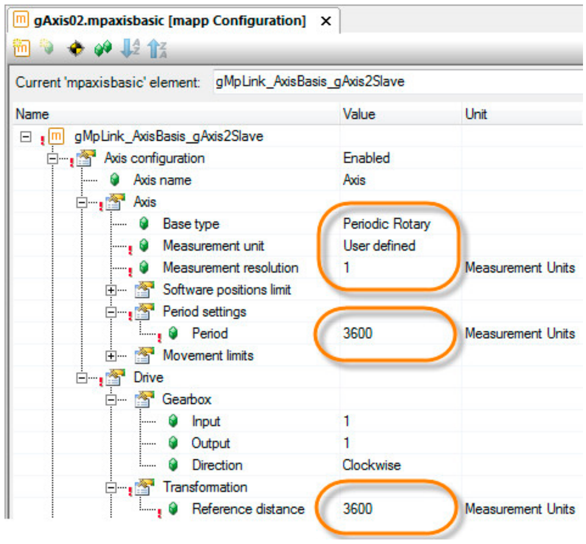


Figure 4: mapp axis configuration for the slave axis

The same parameters used for the master axis can be used for configuring the virtual axes.

In the standard configuration, 2 units are specified as the lag error limit. It may be necessary to increase this limit value.

Exercise: Preparation of two single axes

In preparation, it is necessary to add two real axes for a coupling between master and slave in the project and to control them using the MpAxisBasic component. With the master axis as well as the slave axis, the controller should be able to be switched on and carrying out a homing procedure in "direct" mode should be possible.

1) Add a master axis in the Physical View
2) Add a mapp configuration for the master axis in the Configuration View
3) Set the axis period of the master axis to 3600 units
4) Add a slave axis in the Physical View
5) Add the mapp configuration for the slave axis to the Configuration View
6) Set the axis period of the slave axis to 3600 units
7) Call MpAxisBasic for master axis and slave axis configuration
8) Switch on both controllers
9) Complete the homing procedure for both axes

Exercise: Configure a virtual axis and put it into operation

A virtual axis can also be configured in the Drive Configuration wizard while adding the two real axes. The same parameters used for the master axis are used.

With virtual axes, movements can be performed immediately after enabling MpAxisBasic. It is not necessary to switch on the controller or perform a homing procedure.

Motion control \ ACP10/ARNC0 \ Libraries \ ACP10_MC \ Concept \ Implementation \ Virtual axis

Motion control \ ACP10/ARNC0 Reference manual \ ACP10 \ NC objects \ NC object "ncV_AXIS"

3 General information about drive coupling

Coupling drives electronically results in a predefined synchronized movement of drive axes. The coupling is established using either a specified gear ratio or cams.

Drive A is coupled with drive B using the position setpoint. This means that while the drives are actively coupled, drive A must adjust its position according to the position of drive B. In this case, drive B is the master, which specifies a reference position, and drive A the slave, which has a position based on the master position.

Figure 5: Schematic illustration of a coupling

A coupling always requires a master signal, which provides the reference (position, time) and at least one slave drive, which must follow this reference value using a "rule". However the master signal does not have to come from a real drive as it does in this example. In principle, drives can also be coupled to a variety of suitable reference values. Virtual axes, among others, are available in the B&R drives for this purpose.

The master remains unaffected by the coupling procedure. It is simply used as the basis for the desired coupling signal. For example, if a drive's position value is used as the master signal, then this master axis can still be given a command even while coupling is active. In this situation, the slave drive is still completely dependent on the master signal.

The type of position coupling (i.e. the "rule" that tells the slave drive how it must follow the master signal) can be displayed in a diagram with a comparison of the master and slave position.

This is shown in the image for a linear relationship between the master and slave position:

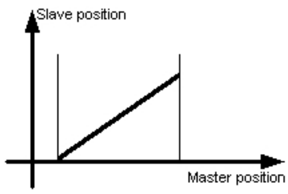


Figure 6: Linear coupling of the slave position to the master position

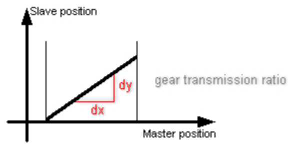


Figure 7: Gear ratio between master position and slave position

Coupling using electronic gears

The position of the coupled master is shown in the horizontal direction. The position of the coupled slave can be seen in the vertical direction.

According to this specification, when the master signal changes uniformly (e.g. the master axis moves at a constant velocity), the velocity of the slave axis is also constant.

In this case, we are referring to "electronic gears". We are talking about a type of coupling that, in practice, is often required. The gear ratio represents the slope of the linear curve.

Coupling using a cam profile

The position relationship of master and slave axis does not have to be necessarily be linear. In principle, custom-defined positioning paths, so-called electronic cam profiles, can be created and used.

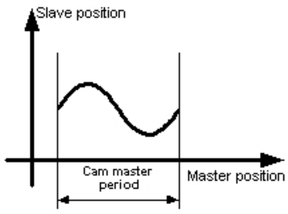


Figure 8: Nonlinear positioning path - Coupling the slave position to the master position using a cam profile

A simple coupling for an electronic gear as well as a coupling via cam profiles can be quickly implemented for the drive. Automation Studio provides the cam profile editor for creating user-specific cam profiles.

The corresponding function blocks used to configure and control the drive couplings can be found in the mechatronics mapp Technology category. The MpAxisCoupling component enables coupling by electronic gears or a cam. In the background, individual function blocks of the ACP10_MC PLCopen library are used.

The Cam Profile Automat offers extensive settings for connecting multiple cam profiles to each other. This done using the MpAxisCamSequenzer function block.

As specified, the mapp technology multi-axis functions should be operated like the MpAxisBasic component. Implementation of these functions in the automatic sequence of an application program is therefore also the same.

Services \ Legacy \ Components \ Mechatronics \ MpAxis - individual and multi-axis controllers \

- Function blocks
- Technical Information \ Internal PLCopen usage

Motion control \ ACP10/ARNC0 \ Libraries \ ACP10_MC \ Categorized function blocks \ Overview of function blocks

3.1 Compensation of delay times

Transferring data from a coupling master to a coupling slave via a network results in a delay. This can be compensated on the coupling master using the position controller parameter for total delay time ("t_total"). Details regarding the compensation of delay times and how they are calculated can be found in the **"Calculation of delay times" table**. The table is provided in Automation Help as a download.

It is necessary to configure a broadcast channel in order to couple an axis to setpoints that originate from outside of the network.

Motion control \ ACP10/ARNC0 \ Libraries \ ACP10_MC \ Categorized function blocks \ Important points \ Axis coupling

- Compensation of delay times
- Coupling axes to different networks

Position controller total delay time ("t_total") in the controller block diagram

The ACOPOS block diagram is shown in the upper part of the figure. In it, the setpoint generator supplies the position controller with the position setpoint. In the lower part, the position controller is shown in detail. Here you can see, among other things, how the value of "t_total" is applied in the position controller.

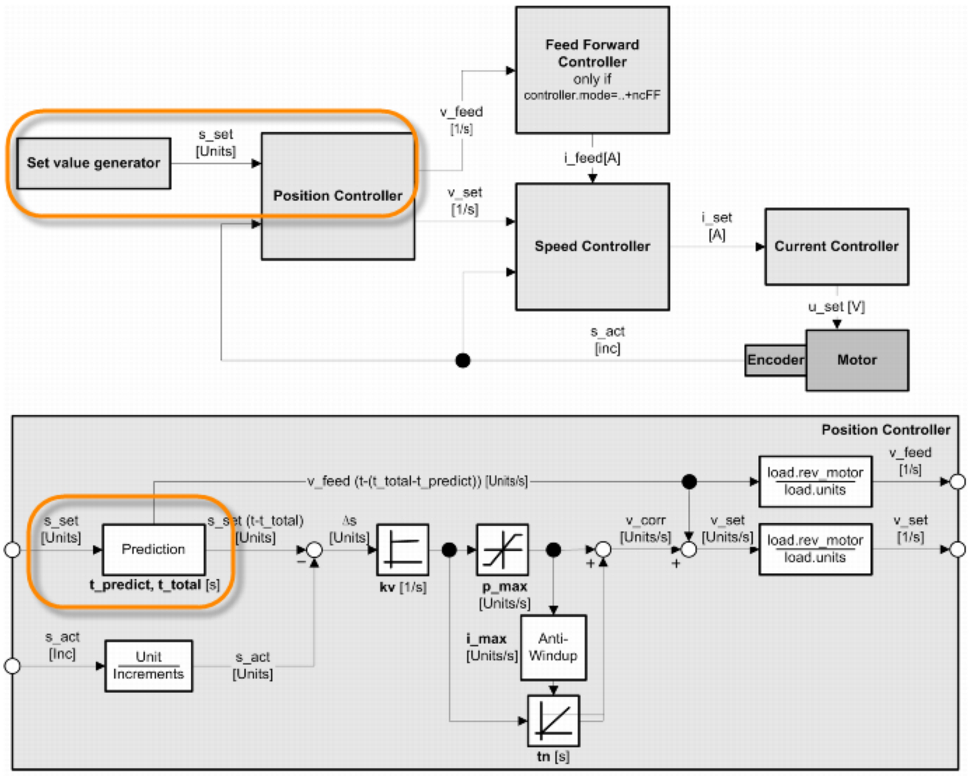


Figure 9: "ncPOSITION" block diagram controller mode

Motion control \ ACP10/ARNC0 \ Reference manual \ ACP10 \ NC objects \ NC object "ncAXIS" \ Controller \ Controller mode "ncPOSITION"

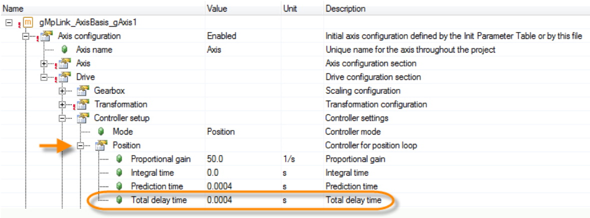


Figure 10: Configuration of "t_total" in the mapp configuration for MpAxisBasic

The "t_total" parameter is configured in the mapp configuration with the parameters for the position controller.

"t_total" monitoring

It's necessary to configure this properly so that the compensation of the delay times on the network works. CTRL position controller: The effect of the compensation can be determined by plotting the "PCTRL_V_SET" parameter: Parameter ID 114: CTRL position controller: Velocity setpoint" with the NC Trace on all coupling axes. In the NC Trace, the setpoint for the velocity setpoint must be changed simultaneously on all coupling axes when a set-point is changed on the coupling master.

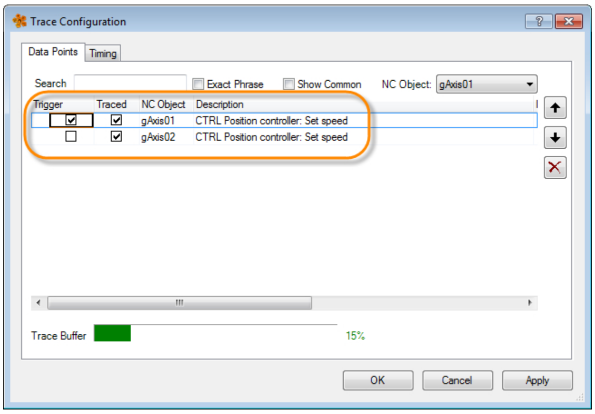


Figure 11: Trace configuration for plotting the velocity setpoint of the position controller for all coupling axes

The following figure shows an NC Trace recording in which the value for the velocity setpoint of the position controller for two coupling axes has been plotted. The recording interval for this NC Trace has been configured with 400 µs. Placing the measurement cursor at a random position results in the same setpoint for both curves. If a deviation is recognized, then the configuration of the "t_total" parameter is unsuitable.

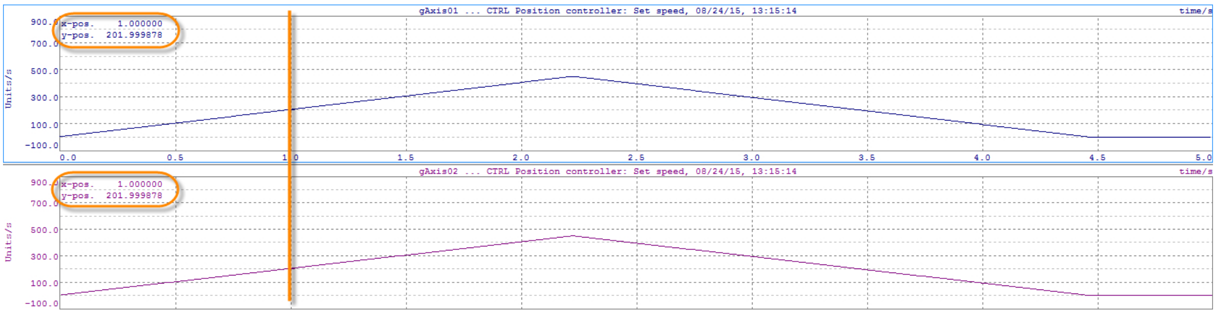


Figure 12: Recording PCTRL_V_SET: Parameter ID 114 on all coupling axes

?

Motion control \ ACP10/ARNC0 \ Reference manual \ ACOPOS drive functions \

- Driver controller \ Position controller
- Controller identification \ Controller parameters (autotuning)

Additional information

Motion control \ ACP10/ARNC0 \ Reference manual \ ACOPOS drive functions \ Network, position coupling and axis cross-link

- Network coupling
- Axis cross-link

Motion control \ ACP10/ARNC0 \ Reference manual \ ACP10SDC

- Feed forward and delay compensation
- Setpoint / position setpoint calculation - SDC internal

4 Linear couplings

4.1 Simple coupling with gear ratio

The functions in the MpAxisCoupling mapp component for controlling electronic gears are quite easy to use.

An MpLink of the master axis and of the slave axis is necessary for coupling two axes.

To prepare the axes for movements, they must be switched on and homed using the MpAxisBasic component.

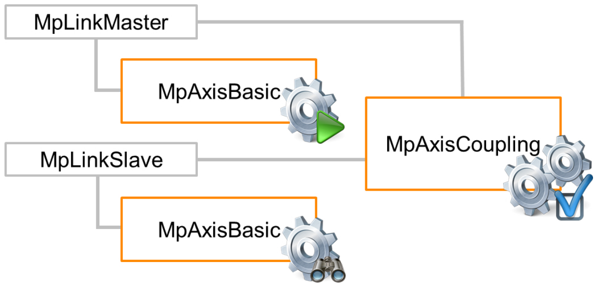


Figure 13: Sketch of the mapp components used

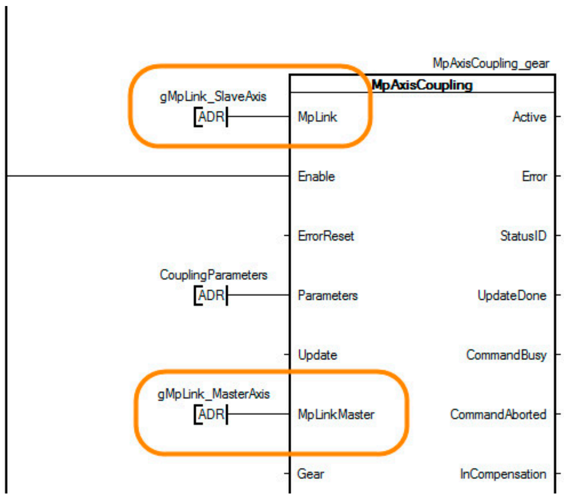


Figure 14: Configuration of MpAxisCoupling in Ladder Diagram: MpLink connection of slave axis and master axis

The figure shows that the MpLink of the master axis as well as of the slave axis have been connected to the MpAxisCoupling component. The component is enabled using the Enable input.

The component is configured via the parameter structure on the "Parameters" input.

Fundamentally, the slave position can also be connected to a master ParID. This is declared in the parameter structure via the "MasterParID" parameter. The specification of the MpLink for the master axis is no longer required in this case.

In parameter structure "CouplingParameters" of data type MpAxisCouplingParTyp, the gear ratio is specified using the numerator and denominator. Additionally, the acceleration and deceleration for the slave axis are determined from the properties in the "Gear" sub-element. In this example, the parameterization is shown in Structured Text. The gear ratio for the coupling is 1:1. The values for acceleration and deceleration for the coupling is 2000 units/s² each.

```
(*set parameters for gear ratio 1:1*)
CouplingParameters.RatioNumerator := 1.0;
CouplingParameters.RatioDenominator := 1.0;
CouplingParameters.Gear.Acceleration := 2000;
CouplingParameters.Gear.Deceleration := 2000;
```

```
(*set parameters for gear ratio 1:1*)
CouplingParameters.RatioNumerator := 1.0;
CouplingParameters.RatioDenominator := 1.0;
CouplingParameters.Gear.Acceleration := 2000;
CouplingParameters.Gear.Deceleration := 2000;
```

Starting the coupling

Input "Gear" of function block MpAxisCoupling is used to start the coupling. In order to do this, the master and the slave axes must be switched on and homed. The values from the info structure of the MpAxisCoupling component can be used as information about the readiness for coupling.

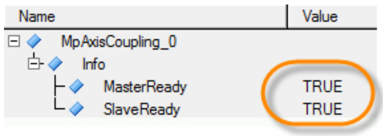


Figure 15: Master and slave axes were fully initialized

> The state of the master axis is not affected by the coupling.
>
> Coupling parameters cannot be changed if the master moves backwards in the active coupling.
>
> The coupling cannot be started if the master is moving backwards!

Decoupling

There are different ways to decouple depending on the application.

- If the slave axis must be stopped, then the "Gear" input of the MpAxisCoupling component can be set to the value FALSE.
- If the slave axis is stopped, then the "Stop" input of the MpAxisBasic component can be set to TRUE for the slave axis.
- If the slave axis must perform a basic movement, one of the **"MoveXYZ"** inputs of the MpAxis-Basic component for the slave axis can be called.

> If the slave should continue at the last velocity, then the **MC_GearOut** function block from the ACP10_MC library can be called.
>
> The state of the slave changes to "Synchronized Motion" when the coupling is started successfully. When the coupling is terminated using MC_GearOut the drive maintains its current velocity and the state changes to Continuous Motion.
>
> Therefore, the **MC_Stop** function block would also have to be used to stop movement of the slave axis, for example.

> Motion control \ ACP10/ARNC0 \ Project development \ Motion control \ Coupling
>
> Motion control \ ACP10/ARNC0 \ Libraries \ ACP10_MC \ Categorized function blocks \
>
> - Important points \ Axis coupling
> - Electronic gears
>
> Services \ Legacy \ Components \ Mechatronics \ MpAxis - individual and multi-axis controllers \ Function blocks \ MpAxisCoupling

Exercise: Configure a simple linear axis coupling

A slave axis should follow the master position with a ratio of 1:1 when the "cmdStartGear" command = TRUE.

1) Create the "CouplingParameters" configuration structure with data type "MpAxisCouplingParType".

2) Configure a 1:1 gear ratio.

3) Set the gear parameters for acceleration and deceleration to 2000 units/s².
4) Call MpAxisCoupling and wait until the master and slave are ready for coupling.
5) Evaluate the "Info.MasterReady" and "Info.SlaveReady" outputs.

   Coupling can be initiated via the "Gear" input when the master and slave axis are ready for it.
6) Start the coupling using the "Gear" input.
7) Optional: Determine the master and slave positions using MC_ReadPosition.

4.2 Dynamic phase shifting

The "PhaseShift" function of the MpAxisCoupling component generates a phase shift in the master position for the slave axis with an active coupling. The master position is shifted with respect to its actual physical position.

The phase shift can only be "seen" by the slave, the master is not affected. The phase shift remains in place until another phase shift command changes it.

The phase shift function can be used if a coupling has already been started via a gear ratio or a cam profile.

Functionality of phase shift

The slave position is determined by the position of the coupling master and the coupling ratio (linear or via cam profile).

The phase shift function generates a value for an additive element or additive master axis. This element is added to the actual position of the master. The resulting value is then applied to the master side of the coupling ratio.

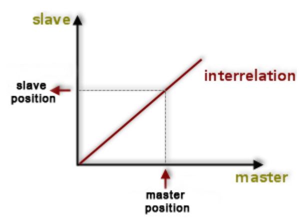


Figure 16: Master - Coupling ratio - Slave

When the phase shift function is activated, the specified target value for the phase shift is approached using a constant movement profile. The movement profile position is continually added to the actual master position, which prevents jumps in position on the slave axis. The master axis is not affected by this action at all.

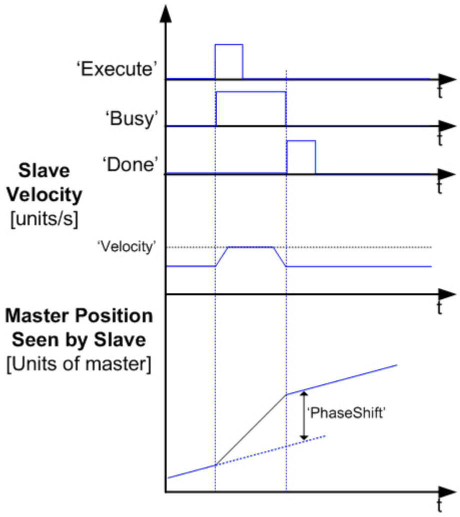


Figure 17: Targeted phase shift generated by the phase shift input

This changes the set position for the coupling slave. A targeted phase shift can therefore be implemented. This smooths out the additive master axis value that is generated.

The phase shift function can be implemented in order to separate products. After sheets of cardboard have been cut, they lie end-to-end on a conveyor belt. They are then transferred to a second conveyor belt. As each sheet reaches the second belt, a phase shift can be implemented. This creates a gap between the products, which is required for further processing.

The resulting slave position is directly dependent on the coupling ratio. The gear ratio for an electronic gear has the following effect on the result:

Gear ratio = 1:5 (Master:Slave)

Master-side shift = 2000 units (additive master axis)

Slave-side shift = 10,000 units

The phase shift is configured via the configuration structure on the "Parameters" input of the MpAxisCoupling component.

Similar functions:

- OffsetShift
- Recovery

Services \ Legacy \ Components \ Mechatronics \ MpAxis - individual and multi-axis controllers \ Function blocks \ MpAxisCoupling \ Description

Motion control \ ACP10/ARNC0 \ Libraries \ ACP10_MC \ Categorized function blocks \ Important points \ Phase and offset shift

Exercise: Configure dynamic phase shifting

Use the "PhaseShift" input of the MpAxisCoupling component in order to switch on a phase shift. The parameters are configured in the configuration structure in the "PhaseShift" substructure.

Use the following settings and plot the velocity setpoint in NC Trace:

- Gear ratio 1:1
- Phase shift: 100 units

The phase shift is additive to the current movement. The phase shift can also be tested while the master axis is idle.

5 Electronic cam profiles

To implement dynamic, nonlinear movements, the B&R drive solution offers the option of using electronic cam profiles for axis coupling. These cam profiles can be created by the user in Automation Studio.

Electronic cam profiles can be used in many different ways.

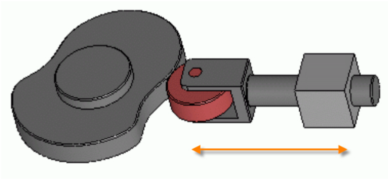


Figure 18: Mechanic cams (*Silberwolf / de.wikipedia.org*)

Cam profiles can be used quite effectively for spring winding machines. Separate axes are used to control the feed, curvature and slope respectively. This makes it possible to create any shape needed (slopes, cones, etc.)

5.1 Introduction

In the cam diagram, we see the master position value in the horizontal direction and the slave position in the vertical direction. The cam profile assigns a respective slave position value for each master position value within a defined range (master period). The slave drive must follow this profile while the drives are actively coupled.

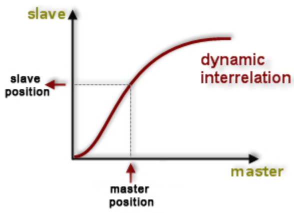


Figure 19: Cam profile as a position relationship between the master and the slave positions.

The master position is converted to a corresponding slave position via the cam. This allows the master to move in both directions. The slave drive is "tied" to the master via the cam.

This means that the velocity and acceleration values of the slave drive result from the velocity and acceleration of the master in connection with the characteristic curve of the cam.

For the entire course of the cam profile, it must be checked if the slave drive can handle the velocity and acceleration values that might occur.

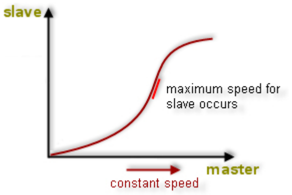


Figure 20: Maximum velocity that occurs on the slave drive

Assumed: The master signal constantly changes, which corresponds to a steady movement of the master axis.

Critical ranges (with maximum values for the slave velocity or acceleration) are represented in the cam profile by the maximum slope (-> velocity as first derivative of the position) and the maximum slope change (-> acceleration/deceleration as second derivative of the position) according to the position comparison.

5.2 Creating cam profiles

Automation Studio has a powerful cam profile editor for creating cam profiles. Cam profiles can be edited in the cam profile editor as soon as they have been inserted into the project.

Cam profiles are added as NC software objects in the Logical View in Automation Studio, then transferred to the controller and can be selected at runtime in the drive application.

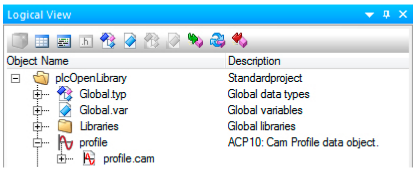


Figure 21: "Profile" object for a cam profile in the Logical View

It is generally advisable to create standardized cam profiles. These have end points with a ratio of 1:1 or 1:0. This allows the cam profile to be stretched in unit scaling as needed using corresponding function blocks or the Cam Profile Automat. As a result, they can be used on a wide range of axes with different scaling.

Motion control \ ACP10/ARNC0 \ Project development \ Motion control \ Coupling \ Cam coupling \ Cam editor

5.3 Editing a cam profile

The cam editor in Automation Studio is a full-featured tool that helps to create and adjust exact cams for the various different coupling tasks.

The following properties can be configured in the cam profile editor:

- General properties
- Color settings
- Extension
- Display options
- Labels and formulas
- Characteristic values for curves
- Notations in the diagram

The following functions are provided in the cam profile editor:

- Fixed points
- Synchronous sections
- Interpolation curves
- Importing mechanical cam profiles

It is now possible to define fixed points on the curve as well as synchronous sections (linear sections along the path of the curve) to create a cam profile. The connection to a continuous cam profile occurs via interpolation curves, which are automatically calculated by the cam editor.

A total of four fixed points and one synchronous section have been defined in the image. The cam profile editor automatically integrates these definitions into a complete cam profile. It does this by calculating and displaying interpolation curves. The user can even specify the form of the interpolation curves.

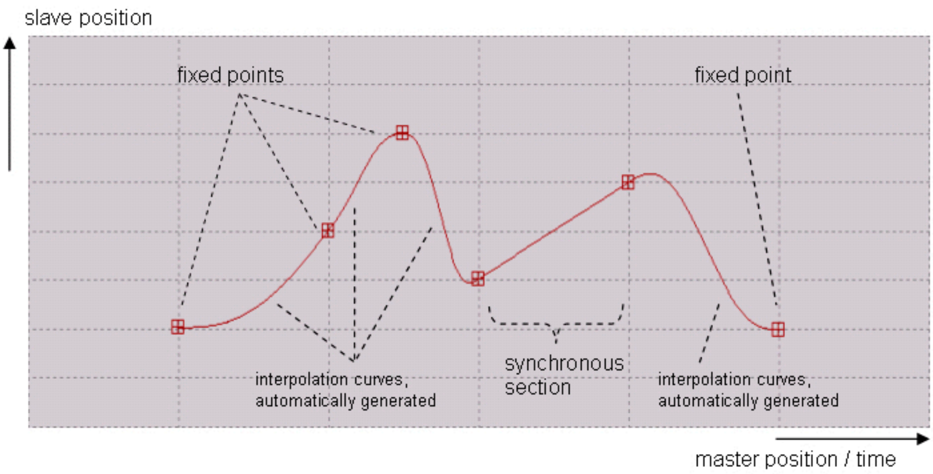


Figure 22: Structure of a cam profile

Motion control \ ACP10/ARNC0 \ Project development \ Motion control \ Coupling \ Cam coupling

Motion control \ ACP10/ARNC0 \ Project development \ Motion control \ Coupling \ Cam coupling \ Cam editor

Fixed points

A fixed point is a point in the cam profile for which the user defines the desired position of the slave axis in relation to a specific position of the master axis.

The notation indicates whether position or time units should be used in the diagrams on the horizontal axis (i.e. the master axis). The use of position is referred to as "mathematical notation". The use of time is referred to as "physical notation" (comparable to a constant master velocity).

Therefore, in physical notation, the first derivative in the fixed point is equal to the velocity and the second derivative in the fixed point is equal to the acceleration of the slave axis. The cam profile represents the path-time diagram of the slave axis.

Motion control \ ACP10/ARNC0 \ Project development \ Motion control \ Coupling \ Cam coupling \ Cam editor \ Fixed points

Synchronous sections

A synchronous section is a section in the cam profile where the user specifies a linear path for the master and slave positions.

A constant master axis velocity within a synchronous section also results in a constant slave movement. In other words, the cam is linear (comparable to an electronic gear).

When using physical notation (master axis = time) the master position is entered as a time value. The slope of the synchronous section corresponds to the velocity of the slave axis in this section. (→ time passes evenly)

Motion control \ ACP10/ARNC0 \ Project development \ Motion control \ Coupling \ Cam coupling \ Cam editor \ Synchronous sections

Interpolation curves

The section of a cam profile calculated by the cam profile editor to connect two defined elements (fixed points, synchronous sections) is called an interpolation curve.

After each new fixed point or synchronous section is entered, an interpolation curve integrating it in the rest of the profile is automatically calculated and displayed. The cam profile editor makes sure that there is exactly one interpolation curve between any two defined components.

Likewise, when a fixed point or a synchronous section is deleted, any extra interpolation curves are also deleted.

The calculation ensures that the cam profile function and its first derivative are constant at the transition points (e.g. the curves do not contain any jumps at the end points).

Various curve types can be selected for each interpolation curve to fine-tune the profile between the defined areas (fixed points and synchronous sections). These provide different predefined shapes according to the type. Specific curve profiles can be implemented using type-specific settings (turning points, joining points, etc.).

Motion control \ ACP10/ARNC0 \ Project development \ Motion control \ Coupling \ Cam coupling \ Cam editor \ Interpolation curves

Importing mechanical cam profiles

It is often necessary to replace mechanical cams with electronic ones or to reproduce content from a CAD system electronically. The cam editor makes it possible to import and then export interpolation points, which allows calculated curves to be reused in CAD, for example.

Motion control \ ACP10/ARNC0 \ Project development \ Motion control \ Coupling \ Cam coupling \ Cam editor \ Mechanical cams

Exercise: Create the "profile" cam profile

Insert a cam profile into your project and edit it in the cam profile editor. Do this using the option for defining fixed points and synchronous sections.

The graph shows a movement profile in which the slave axis reaches its maximum position in the right quarter of the profile, then turns and returns to the initial position.

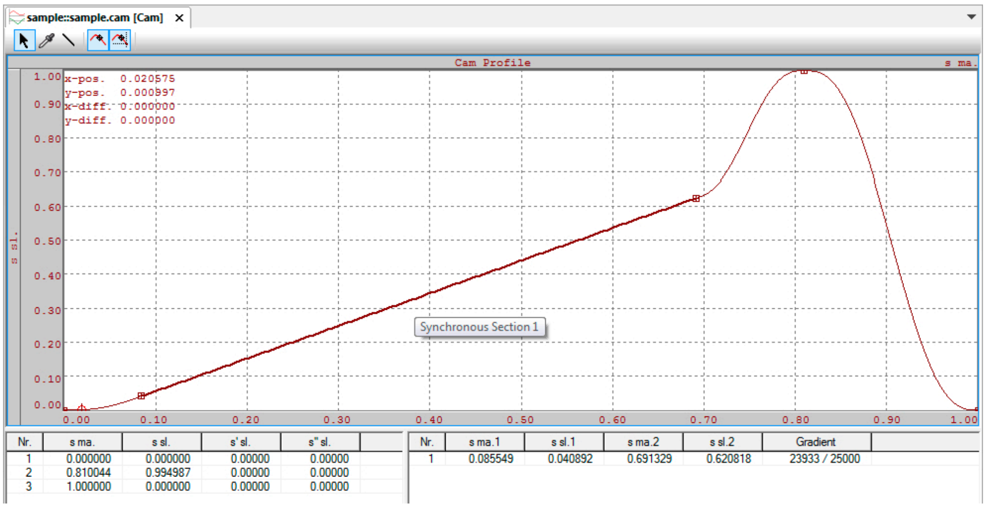


Figure 23: Example of a cam profile

When creating your cam profile, make sure that the profile has the same slope at the start and end points. This characteristic is important for subsequent coupling applications.

Transfer the project containing the new cam profile to the controller and activate online software comparison. The cam profile should now appear as a data object on the controller.

5.4 Configuring cam profile coupling

When enabling the MpAxisCoupling component, the cams specified in the parameter structure are transferred to the slave drive. The following section describes how the cam function is configured and how a coupling is started using cams.

5.4.1 Parameterization of MpAxisCoupling

The desired cams are specified in the parameter structure for transferring a cam object to the coupling slave. The desired cam can be selected via an index. When this function block is called, the corresponding cam is transferred to the slave drive.

The selected cam is processed cyclically until the coupling is removed again. This results in a continuous positioning loop for the slave.

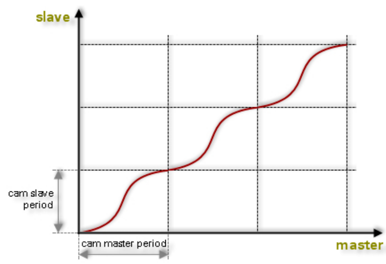


Figure 24: Cyclic sequence of the cam profile

Settings in the configuration structure

In the parameter structure, the scaling of the standardized cam profile can be set via the numerator and the denominator. The names of up to five cam profiles are specified using the "CamTable" STRING array. The cam profile that needs to be enabled is selected via the "CamTableID" index.

```
(*set parameters for cam coupling*)
(* note axis period is 3600 Units for master and slave axis
   slave runs profile 3 times each master period *)
CouplingParameters.RatioNumerator := 1200.0;
CouplingParameters.RatioDenominator := 1200.0;

CouplingParameters.Cam.CamTableID := 0; (*use cam index 0*)
CouplingParameters.Cam.CamTable[0] := 'profile';
(*set parameters for cam coupling*)
(* note axis period is 3600 Units for master and slave axis
   slave runs profile 3 times each master period *)
CouplingParameters.RatioNumerator := 1200.0;
CouplingParameters.RatioDenominator := 1200.0;

CouplingParameters.Cam.CamTableID := 0; (*use cam index 0*)
CouplingParameters.Cam.CamTable[0] := 'profile';
```

In the configuration example, the scaling of the cam in the x- and y-directions is specified via the numerator-denominator ratio of the parameter structure. It is assumed from this that a standardized cam (1:1) has been created. With a master period of 3600 units, this means that the slave begins the cam again from the beginning after 1200 units.

Smooth entry must be guaranteed when starting a cam profile coupling. When a sequence of cam profiles is created, the end of one cam profile leads directly into the beginning of the next. It is therefore important to ensure that the velocity and acceleration of the transition is consistent. Sharp angles are not allowed to occur anywhere in the positioning path.

Services \ Legacy \ Components \ Mechatronics \ MpAxis - individual and multi-axis controllers \ Function blocks \ MpAxisCoupling \ Description

5.4.2 Coupling cam profiles

The transfer of cam profiles to the slave drive occurs when the MpAxisCoupling component is enabled. Changes to the configuration can be activated via the "Update" command.

Ways to start the coupling

There are two ways to perform a lead-in or lead-out movement. The first is an additional cam profile that is used for coupling or decoupling. The second possibility is a defined distance within the actual cam profile that is used as compensation.

The start of the cam profile is specified using the "MasterStartPosition" parameter. The cam profile begins within an axis period at this absolute master position.

The coupling is started using the "Cam" command.

Lead-in and lead-out movements using a compensating movement

Using the "LeadIn" and "LeadOut" elements in the parameter structure, the "MasterOffset" parameter can be used to define from which point within the cam profile the master and slave must be synchronous. In addition, "MasterDistance" and "SlaveDistance" can be used to perform a compensating movement before the "MasterStartPosition" is reached.

Lead-in and lead-out movement using a cam profile

Using elements "LeadIn" and "LeadOut" in the parameter structure, the lead-in and lead-out movements can be expanded via an additional cam. This is specified via parameters "CamEnable" and "CamTableID". The can profile is scaled using parameters "MasterScaling" and "SlaveScaling".

The example shows the configuration of a compensating movement via cam "linear".

```
(*lead out with linear movement using cam table "linear"*)
CouplingParameters.Cam.CamTable[1] := 'linear';
CouplingParameters.Cam.LeadOut.CamTableID := 1;
CouplingParameters.Cam.LeadOut.CamEnable := TRUE;
CouplingParameters.Cam.LeadOut.MasterScaling := 400;
CouplingParameters.Cam.LeadOut.SlaveScaling := 400;

(*lead out with linear movement using cam table "linear"*)
CouplingParameters.Cam.CamTable[1] := 'linear';
CouplingParameters.Cam.LeadOut.CamTableID := 1;
CouplingParameters.Cam.LeadOut.CamEnable := TRUE;
CouplingParameters.Cam.LeadOut.MasterScaling := 400;
CouplingParameters.Cam.LeadOut.SlaveScaling := 400;
```

Since the slave requires a certain amount of time to reach its starting position, the master may pass its starting position several times before it is ready. Once the slave has reached its starting position, the coupling is started as soon as the next "MasterStartPosition" has been reached.

As a result, activation of the coupling may be shifted by a number of master periods depending on the situation. The master remains completely unaffected.

Motion control \ ACP10/ARNC0 \ Libraries \ ACP10_MC \ Function blocks \ MC_CamIn

Scaling the cam profile

A cam can be scaled on both the master and slave side in order to create the coupling. In the parameter structure of MpAxisCoupling, parameters "RatioNominator" and "RatioDenominator" are intended for this.

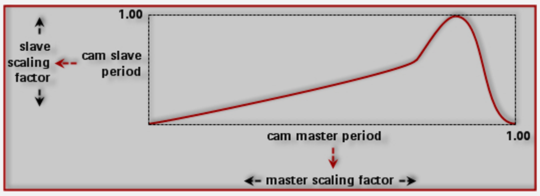
This extends the length of the master and slave cam by the corresponding factor.

```
CouplingParameters.RatioNumerator := 1200.0;
CouplingParameters.RatioDenominator := 1200.0;
```

```
CouplingParameters.RatioNumerator := 1200.0;
```

```
CouplingParameters.RatioDenominator := 1200.0;
```

Cams are often created with a master-side extension of one unit (cam master period = 1) and a slave-side extension of one unit (cam slave period = 1). This makes it rather easy to "stretch" a cam to match a particular process:



When using a linear cam, the "gear ratio" can be determined using the gauge factors.

Deactivating the coupling

An active coupling can be removed again by setting the "Cam" input to the value FALSE. Additionally, coupling can be deactivated by stopping the slave axis or executing a basic movement on the slave axis.

Exercise: Axis coupling using a cam profile

A master and slave axis should be coupled using a cam profile. To do so, a cam profile must first be created in the Logical View. The coupling is established using MpAxisCoupling. Coupling via the cam profile is carried out over a period of 1200 units.

1) Create a cam profile called "profile". The cam profile should look like the one in the graph.

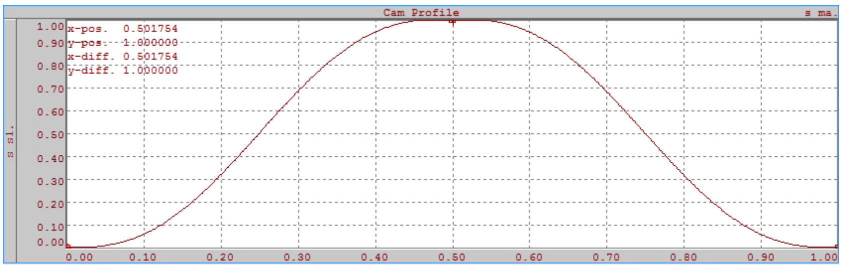


Figure 25: Example of the "profile" cam profile

2) Carry out the configuration using the "CouplingParameters" structure.

```
CouplingParameters.RatioNumerator := 1200.0;
CouplingParameters.RatioDenominator := 1200.0;
CouplingParameters.Cam.MasterStartPosition := 0;
```

3) Call MpAxisCoupling.

4) Wait until the master and slave axis are ready for coupling.

5) Enable the coupling via cam profile using the "Cam" input.

6) Check the master and slave positions.

Exercise: Trace the position setpoints with an active coupling via cam profile

The objective of this task is to plot the position behavior of master and slave axes.

In each case, the "Position" output of the MpAxisBasic component for the master and slave axes must be plotted in Automation Studio using the variable oscilloscope. In order to display both positions in a Trace cam, the position values can each be copied to a global variable, for example.

1) Start a variable oscilloscope
2) Add the position of the master axis
3) Add the position of the slave axis
4) Start a continuous movement on the master
5) Couple slave drive to the master drive
6) Transfer the trace configuration and plot the position behavior of the two axes
7) Upload the Trace data

The graph shows the comparison of master position and slave position, which results based on the sample configuration. In the bottom diagram, a full master period of 3600 units is shown. The top diagram shows the slave position. The "profile" cam profile used was scaled in the x and y directions by a factor of 1200. What emerges from this is that the slave performs the cam profile three times within a master period. The value 0 was specified for the "MasterStartPosition" parameter, whereby the coupling on the slave begins with the master period.

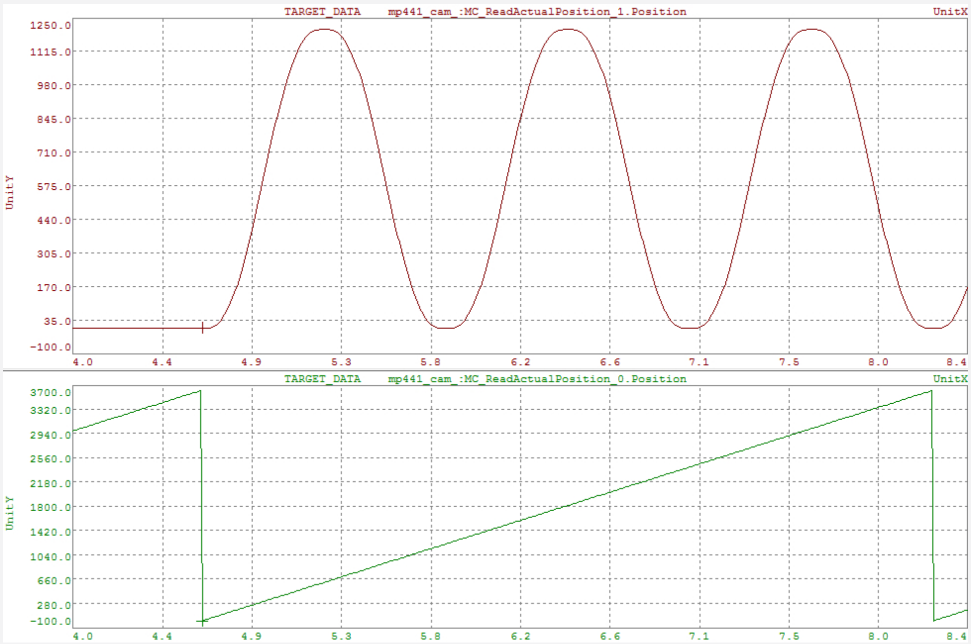


Figure 26: Comparison of the master position and slave position.

Exercise: Expand cam profile coupling by LeadOut with "linear" cam profile

When deactivating the coupling, the slave drive should perform another compensating movement using a cam. Cam "linear" is created for this. This consists of a one-to-one relationship between the master position and slave position and therefore contains only one synchronous section. Using scaling with parameters "MasterScaling" and "SlaveScaling" determines how the slave compensating movement is maintained after decoupling. The behavior should be recorded using a variable oscilloscope. The following parameters can be used:

```
(*lead out with linear movement using cam table "linear"*)
CouplingParameters.Cam.CamTable[1] := 'linear';
CouplingParameters.Cam.LeadOut.CamTableID := 1;
CouplingParameters.Cam.LeadOut.CamEnable := TRUE;
CouplingParameters.Cam.LeadOut.MasterScaling := 1700;
CouplingParameters.Cam.LeadOut.SlaveScaling := 1700;
```


```
(*lead out with linear movement using cam table "linear"*)
CouplingParameters.Cam.CamTable[1] := 'linear';
CouplingParameters.Cam.LeadOut.CamTableID := 1;
CouplingParameters.Cam.LeadOut.CamEnable := TRUE;
CouplingParameters.Cam.LeadOut.MasterScaling := 1700;
CouplingParameters.Cam.LeadOut.SlaveScaling := 1700;
```


1) Create the "linear" cam
2) Transfer the cam to the controller
3) Establish the LeadOut parameters for the "linear" cam
4) Start the coupling
5) Record the master position and slave position using a variable oscilloscope.
6) Deactivate the coupling and stop the trace
7) Evaluate the recorded data

The graph shows that, after uncoupling, the slave position completes processing of the current cam profile to the end. Immediately after that, the compensating movement is performed using the "linear" cam profile, just as it was configured via the scaling factors.

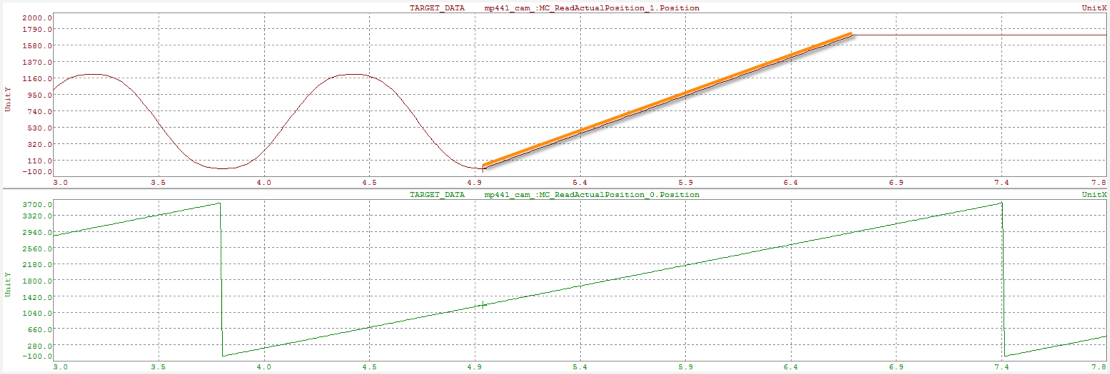


Figure 27: Comparison of slave position (above) and master position (below); LeadOut via "linear" cam profile is shown in orange

For a coupling via a 1:1 straight line, a cam already predefined in the drive can be used. In this case, it is not absolutely necessary to generate a separate cam object.

Further information:

• 6.6 "Predefined cam profiles and sequences" on page 44

5.4.3 Changing cam profiles

When a cam profile coupling is active, the cam profile can be switched by changing the "CamTableID" parameter and by subsequently calling the "Update" command.

When a new CamTableID is entered and a rising edge occurs on the Update input, the period of the active cam profile is ended and the new cam profile begins. The end point of the first cam profile is the starting point of the second cam profile.

Neither the master and slave-offset nor the start mode have any effect on the cam profile change.

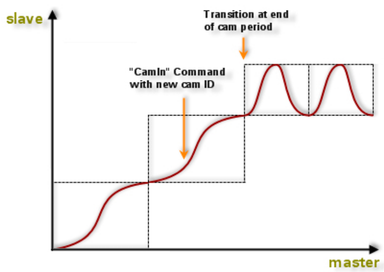


Figure 28: Cam profile transition

Even here it is important to ensure a smooth transition between cam profiles.

Once the cam profiles have been transferred to the respective drive, they can be arranged in any sequence needed. The routines for changing the cam profile must be performed in the application program at the corresponding times.

Motion control \ ACP10/ARNC0 \ Libraries \ ACP10_MC \ Categorized function blocks \ Cams \ MC_CamIn \ Changing the cam

Exercise: Change the cam profile

Prepare the configuration in such a way that cams "profile" and "linear" can be changed between using parameter "CamTableID". Record the position of the master axis and slave axis with the variable oscilloscope.

1) Activate coupling
2) Plot the position of the master and slave axes cyclically
3) Change the value of "CamTableID" parameter from 0 to 1
4) Execute "Update" command
5) Stop trace
6) Upload trace data and analyze cam transition

If a second cam profile is not available, it's enough to change the values for the cam profile scaling. In principle, this procedure is carried out the same way as coupling a new cam profile.

The graph shows the change of the "profile" cam profile to the "linear" one and vice versa. The change between the cam profiles always occurs on a positive edge on the "Update" input. Changing the cam profiles is highlighted in the top diagram.

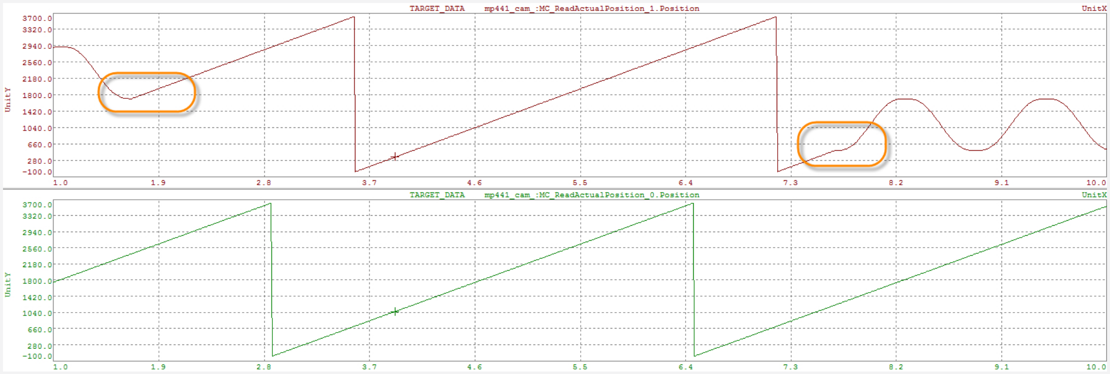


Figure 29: Comparison of the master position and slave position: The cam profile change is highlighted in orange in the top diagram.

5.5 Virtual axis as master

Until now, a real axis was used as a coupling master. A virtual axis can be used as a coupling master in order to solve more complex tasks. Real slave axes then follow the virtual master position. For a virtual axis, turning on the controller and homing are no longer required. Therefore, movements can be started on the virtual axis immediately after enabling MpAxisBasic.

In this example, the tip of a ski is sanded.
Axis A represents the feed rate from the left side of the ski to the right.
Axis B is used to implement the pivot movement up to the tip of the ski.
Both axes A and B are coupled by a separate cam profile to the position of a virtual master axis.
The three timing diagrams show the positions of the axes with reference to the time.
The path for setting off a contour in one direction is outlined.

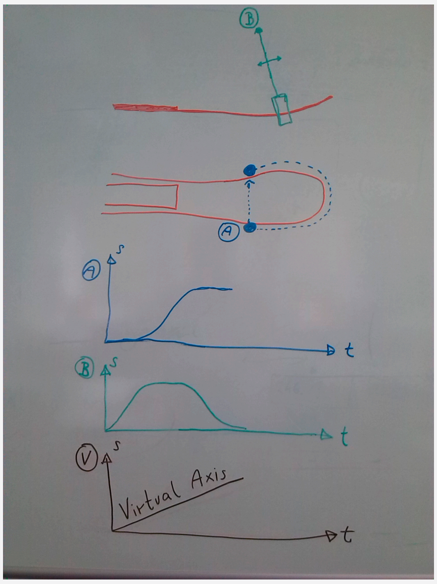


Figure 30: Sketch of the grinding machine as well as the path-timing diagrams for the virtual axis, axis A and axis B.

Motion control \ ACP10/ARNC0 \ Libraries \ ACP10_MC \ Concept \ Implementation \ Virtual axis

Motion control \ ACP10/ARNC0 Reference manual \ ACP10 \ NC objects \ NC object "ncV_AXIS"

6 Cam Profile Automat

The Cam Profile Automat allows event-controlled coupling of electronic cam profiles.

The following example moves step-by-step through a more in-depth explanation of Cam Profile Automat functions.

A Cam Profile Automat is already used in the background for gear couplings, cam profile coupling as well as for some predefined processes (6.6 "Predefined cam profiles and sequences"). The Automat configuration used is described in detail in the "Additional information" section of the function block description in the Automation Studio help system.

6.1 Introduction

First, let's look at a procedure using a packaging machine.

The product transporter acts as the master axis. The slave axis closes each plastic container with a cap. A high-speed digital input (trigger) detects if a product is present. If no product is present, then the slave remains in standstill. Otherwise the container is closed with a cap.

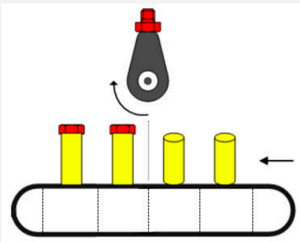


Figure 31: Starting point for the slave

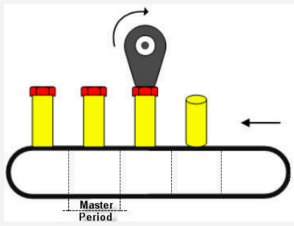


Figure 32: Cap applied

Necessary steps when using MpAxisCoupling ("Cam" function)

In the following section, we'll think about how we can solve this example with the MpAxisCoupling function block. First, we need two cam profiles:

Cam profile 1, which keeps the slave at a standstill when a container is not present.	Cam profile 2, which is required for the application of the cap.

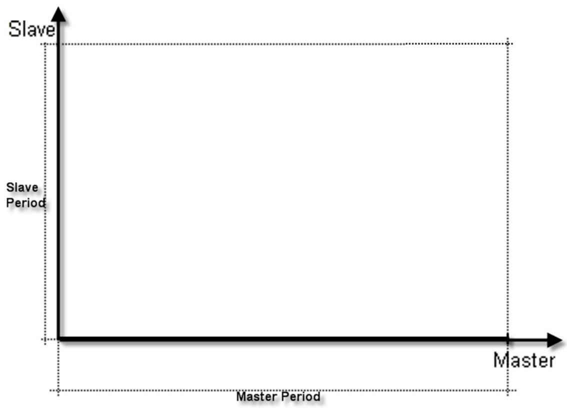


Figure 33: First cam profile for keeping the slave at a standstill

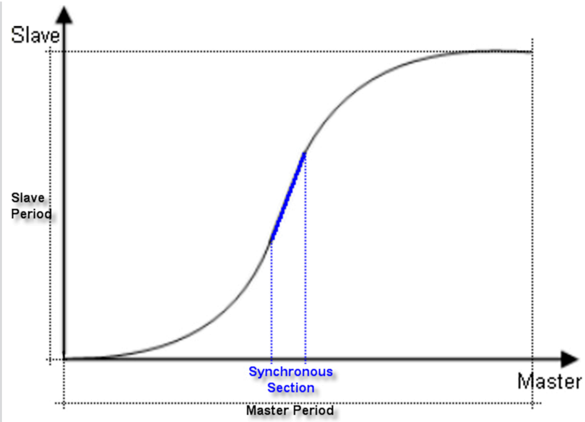


Figure 34: Second cam profile for applying the cap

First, the two cam profiles must be transferred to the drive. Then a control program must be used to check if a trigger signal has been received:

- If so, you would need to change to the cam profile 2 with the MpAxisCoupling function block.
- If there is no trigger signal, then cam profile 1 should be coupled via MpAxisCoupling.

Task of the Cam Profile Automat (MpAxisCamSequencer)

A more efficient method is to let the drive decide on its own which cam profile should be processed based on the current process situation. Thus the complexity in the control program is reduced and faster reaction times are achieved.

It is precisely situations such as this for which the Cam Profile Automat was created. This is initialized and parameterized on the corresponding slave drive, where it is then processed independently. This keeps the CPU load comparably low, even when a large number of axes are in use. The running process benefits from minimal response times. There are also many ways to intervene in the processing of active Cam Profile Automats.

In the following section, the structure and the functionality of a Cam Profile Automat are shown using the example.

6.2 Structure and functionality

The example can be structured in the cam automat as follows. In the process, automat states are defined, which are switched between using change events

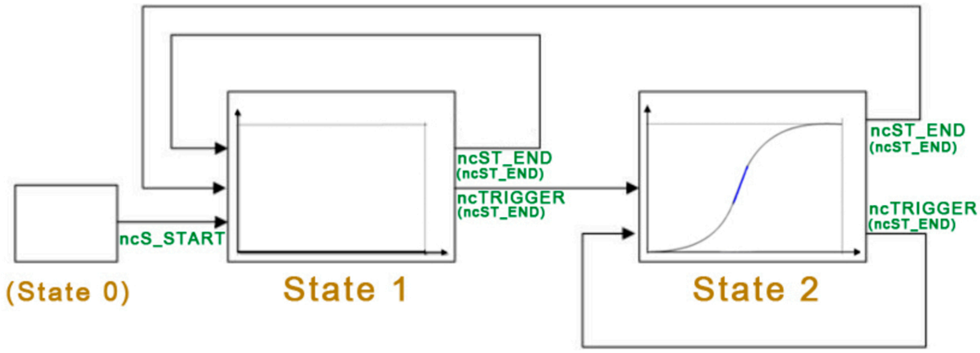


Figure 35: Cam Profile Automat structure for the bottle capping machine

Automat states

The two cam profiles used in the example of the bottle capping machine are each packed in a specific state. These are called Cam Profile Automat states. The following states result in the example:

- **State1**
  The slave doesn't execute any movement in this state because there is no product.
- **State2**
  The capping process is completed in this state.
- **State0**
  This state is optional and can be used as a starting state or a waiting state. This state does not have a cam profile assigned to it.

In terms of the master, the length of a state corresponds to the time it takes to transport a product, i.e. waiting for the next product. In terms of the slave, the length of a state corresponds to the distance required to complete the capping process.

Change events

A change event is a defined event that should cause a change of state (e.g trigger event ncTRIGGER, or reaching the end of the state ncST_END, etc).

The user must also define when the change should take effect. For example, it can take place at the end of the state (ncST_END) or immediately (ncAT_ONCE) when the event occurs. The new state that should be changed to must also be defined. The end result is an entire series of Cam Profile Automat states. Two change events have been defined for each of the two states in our example.

Cam Profile Automat sequence for the bottle capping machine

The bottle capping machine changes to State1 after the event-controlled start (the start occurs at a specific master position) in State0.

The slave does not perform any movements in State1. The first bottle must therefore be left out when starting the machine. If a trigger signal (ncTRIGGER) is detected during processing of State1, then the machine changes to State2 at the end of State1 (ncST_END), at which point the capping process is then executed.

During execution of State2, one bottle is capped. If another trigger signal is detected during this state it is repeated.

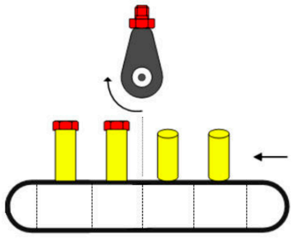


Figure 36: Bottle capping machine

If a product is not present or if State2 runs completely to the end (ncST_END) without a trigger signal having occurred, then the machine switches to State1. Then the slave drive doesn't perform any movement. The Cam Profile Automat remains in State1 until a trigger signal is received again. The automat is switched to State2 in which the capping process is continued.

It is necessary to ensure that the trigger signal arrives on time within a product interval. If multiple signals are sent, only the first trigger signal on the end of the state is evaluated.

According to this circuit diagram, different cam profiles can be sequenced as in a state machine.

First, the cam automat is parameterized. Then it can be started in an arbitrary state. The cam automat runs through the individual states in accordance with the set change events and subsequent states.

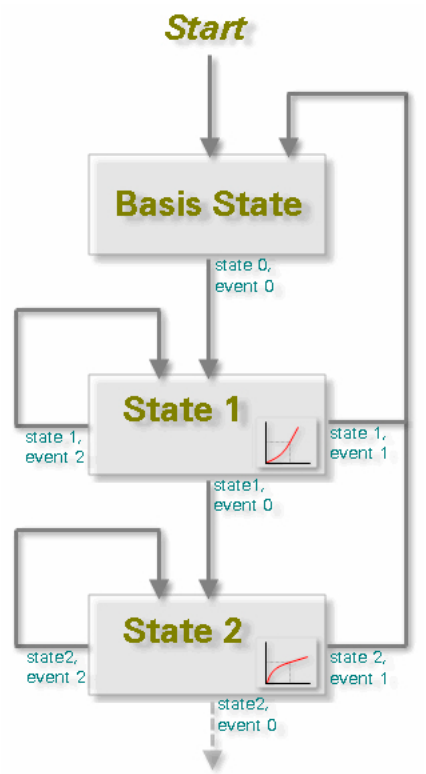


Figure 37: Sequence of Cam Profile Automat states

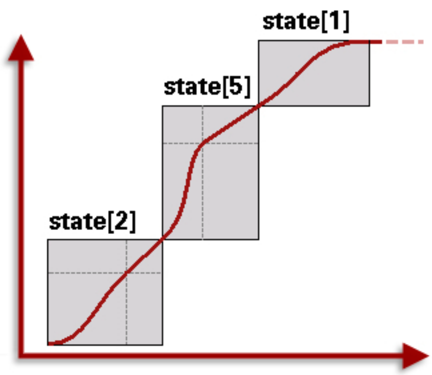


Figure 38: Sequence of Cam Profile Automat states

Motion control \ ACP10/ARNC0 \ Libraries \ ACP10_MC \ Categorized function blocks \ Cam automat

Motion control \ ACP10/ARNC0 \ Reference manual \ ACOPOS drive functions \ Cam automat \

Services \ Legacy \ Components \ Mechatronics \ MpAxis - individual and multi-axis controllers \ Function blocks \ MpAxisCamSequencer

6.3 Automat configuration

The MpAxisCamSequencer component is used for the parameterization and execution of the Cam Profile Automat. First the Cam Profile Automat states and change events are defined with the help of the parameter structure.

When enabling the MpAxisCamSequencer component, all used cam profiles are transferred to the drive. Then the Cam Profile Automat is started and controlled.

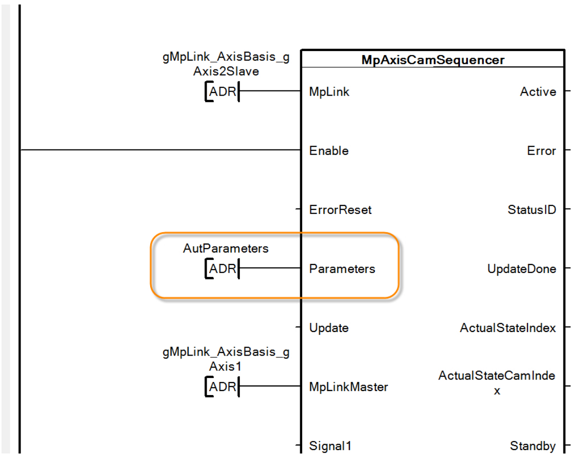


Figure 39: Calling the MpAxisCamSequencer component in Ladder Diagram; "AutParameters" parameter structure highlighted orange

6.3.1 Define the states

Up to 15 states can be defined. The base state (state 0), is a special case. Here it is not possible to assign a cam profile or a compensation gear. Only the desired change events have to be defined for the base state. This serves more or less as an initialization or waiting step.

For each automat state, except State0, the following can be defined:

- A cam profile, which is first transferred to the drive before it can be used. Then this can be used in any state.
- Optionally, a compensation gear that corresponds to an automatically calculated profile can be used. This offsets state transition position and velocity differences. It ensures a continuous connection of the cams. There are different variations of this (Configuring cam profile coupling).

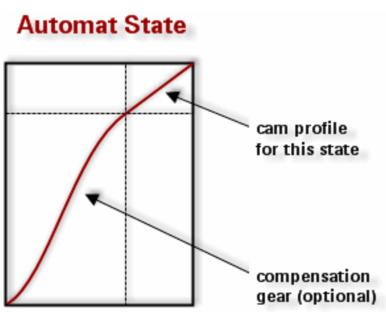


Figure 40: State with compensation and cam profile

It is possible to disable the compensation gear. If this is done, then the state only contains the cam profile itself.

If a compensation gear is used in an automat state, then it will always be processed before the corresponding cam profile in the state.

The parameter structure of MpAxisCamSequencer contains the "Configuration" element. This is used to parameterize the Cam Profile Automat states. The parameters for the 15 possible states can be described via the "State" array element.

```
AutParameters.Configuration.State[x]...
```

Parameter	Description
CamProfileIndex	The CamProfileIndex input is used to select the cam profile for the state.
MasterFactor SlaveFactor	MasterFactor and SlaveFactor define the master and slave-side scaling of the selected cam.

Table 1: Overview of basic parameters for state without compensation

Multiple change events can be determined for each state. Each state provides the "Event" array element for the definition of the change events.

```
AutParameters.Configuration.State[x].Event[y]...
```

Depending on which change events were used, the additional specification of optional parameters is required.

Parameter	Description
RepeatCounterInit	RepeatCounterInit is the initial value for the counter when using the ncCOUNT event type. The counter state is decreased by one each time the end of the state has been reached. The event is generated when the counter reaches zero.
RepeatCounterSet	RepeatCounterSet can be used to change the current counter state on a running automat.

Table 2: Optional parameters for state when using the ncCOUNT change event

Optional parameters for the use of compensation gears are described in the reference manual in the Automation Studio help system.

Motion control \ ACP10/ARNC0 \ Libraries \ ACP10_MC \ Categorized function blocks \ Cam automat \ General \ Cam automat structure

Motion control \ ACP10/ARNC0 \ Reference manual \ ACOPOS drive functions \ Cam automat \ Compensation gear and cam change

6.3.2 Definition of the change events

A change event must at least be defined for a state to induce a state change. Up to five change events are available for each state.

A change event has the following properties:

- Target state (NextState)
- Event type (Type)
- Event attribute (Attribute)

Motion control \ ACP10/ARNC0 \ Libraries \ ACP10_MC \ Categorized function blocks \ Cam automat \ General \ Cam automat structure

Target state (NextState)

The target state determines what state should be activated next. The current state can also be selected here for repetition.

Event type (Type)

The event type determines which event triggers a state change. This can be an "external" signal trigger or the end of the current cam profile. A complete list of event types and examples can be found in the Automation Studio help system.

Motion control \ ACP10/ARNC0 \ Libraries \ ACP10_MC \ Categorized function blocks \ Cam automat \ General \ Event types

Motion control \ ACP10/ARNC0 \ Reference manual \ ACOPOS drive functions \ Cam automat \ States \ Events

Event attribute (Attribute)

The event attribute defines the time at which the state change occurs, which is triggered by the corresponding event (action point). This means that the actual state change can be placed at the end of the cam profile when using a trigger as change event, which occurs according to circumstances in the cam profile.

Event attribute	Description
ncAT_ONCE	The transition to the next state is executed immediately or at the beginning of the next sampling cycle.
ncST_END	The change into the next state is not executed before the end of the current state after the compensation and the curve.

Table 3: Overview of the defined event attributes

Motion control \ ACP10/ARNC0 \ Libraries \ ACP10_MC \ Categorized function blocks \ Cam automat \ General \ Event attributes

Motion control \ ACP10/ARNC0 \ Reference manual \ ACOPOS drive functions \ Cam automat \ States \ Events \ Event attributes

The image illustrates how an event attribute works in relation to a change event. A linear curve progression is shown which runs from left to right.

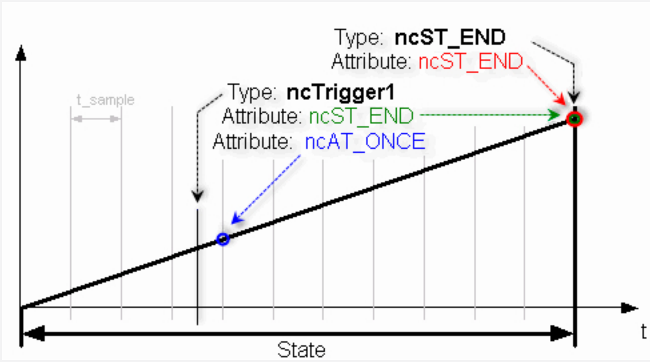


Figure 41: Action point for various events and attributes

Let's assume that a previously defined change event "Trigger1" occurs within this state. The event attribute ncAT_ONCE is used to immediately change to a defined new state, taking the sampling cycle in the drive into consideration.

As a result, the system places the subsequent curve profile precisely at the position of the actual trigger event. Therefore, inaccuracies do not occur in the positioning sequence due to the sampling cycle. When the ncST_END (state end) event attribute is used, this change is made at the end of the current cam profile.

The following example illustrates the layout of the data structure regarding the parameterization of the states and change events. Only a change event is defined in state 0.

In state 1, the state change at the end of the cam occurs when Signal1 ("Signal1" input of the MpAxisCamSequencer component) has been set.

```
(*State 0*)
AutParameters.Configuration.State[0].Event[0].Type      := ncST_END;
AutParameters.Configuration.State[0].Event[0].Attribute := ncAT_ONCE;
AutParameters.Configuration.State[0].Event[0].NextState := 1;

(*Automat STATE 1 Standstill*)
AutParameters.Configuration.State[1].CamProfileIndex    := 16#FFFE;
AutParameters.Configuration.State[1].MasterFactor       := 200;
AutParameters.Configuration.State[1].SlaveFactor        := 0;

AutParameters.Configuration.State[1].Event[0].Type      := ncSIGNAL1;
AutParameters.Configuration.State[1].Event[0].Attribute := ncST_END;
AutParameters.Configuration.State[1].Event[0].NextState := 2;

AutParameters.Configuration.State[2]...
...
(*State 0*)
AutParameters.Configuration.State[0].Event[0].Type      := ncST_END;
AutParameters.Configuration.State[0].Event[0].Attribute := ncAT_ONCE;
AutParameters.Configuration.State[0].Event[0].NextState := 1;

(*Automat STATE 1 Standstill*)
AutParameters.Configuration.State[1].CamProfileIndex    := 16#FFFE;
AutParameters.Configuration.State[1].MasterFactor       := 200;
AutParameters.Configuration.State[1].SlaveFactor        := 0;

AutParameters.Configuration.State[1].Event[0].Type      := ncSIGNAL1;
AutParameters.Configuration.State[1].Event[0].Attribute := ncST_END;
AutParameters.Configuration.State[1].Event[0].NextState := 2;

AutParameters.Configuration.State[2]...
...
```

Exercise: Generate automat configuration for a flying saw

A flying saw should be implemented. The description for the Cam Profile Automat configuration can be found in the Automation Studio help system. The source file (FlyingSaw.st) for the Automat configuration can be imported into the Automation Studio project with the LibACP10MC_Automat_ST library sample. In addition, examples of automat configurations are documented in the Automation Studio help system. Information about the settings for the master and slave axis periods can be found in the respective description.

1) Apply the automat configuration for a flying saw

2) Analyze the states and events for the selected automat configuration

See: Automat configuration in attachment

"Solution: Controlling a flying saw with MpAxisCamSequencer" on page 49

Motion control \ ACP10/ARNC0 \ Examples \ Motion control \ Cam automat \ Automat configuration \

- Automat configuration - Labeling machine
- Automat configuration - Flying saws
- Automation configuration - Cross cutter

Motion control \ ACP10/ARNC0 \ Reference manual \ ACOPOS drive functions \ Cam automat \ Examples

6.3.3 Define global parameters

The global parameters are settings that apply to all Cam Profile Automat states. Global settings for the Cam Profile Automats are made via the parameter structure.

Parameter	Description
StartPosition	The StartPosition allows changing from base state 0 to another state at the moment a specific master position is reached. To do this, a corresponding change event with the event type ncSTART must be defined for base state 0. Specification of the next parameter, StartInterval, is also important.
StartInterval	If the master position has already passed the StartPosition, then the change event ncSTART is generated at the next multiple of the StartInterval.

Table 4: Overview of base parameters

Parameter	Description
MaxMasterVelocity	The slave uses the maximum master velocity to calculate its compensation gearing and to check if its limits have been exceeded. (ACOPOS warning.) This parameter is only required when compensation gears are being used.
StartState	StartState enables the automat to be started in any state. The automat starts in the base state 0 if this parameter is not specified.
StartMaRelPos	StartMaRelPos can be used to start in the initial state within the cam profile. StartMaRelPos specifies the master distance relative to the beginning of the cam profile. If an existing compensation gear exists in the initial state, it is ignored.

Table 5: Overview of optional parameters

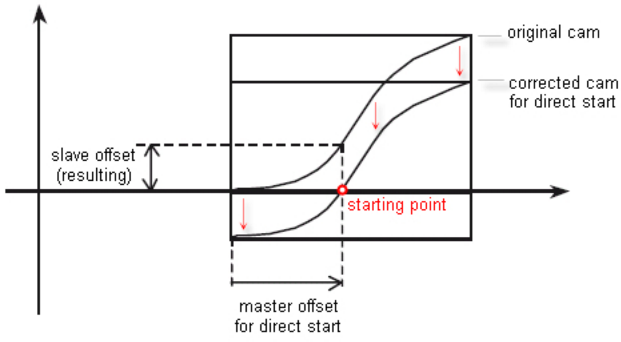
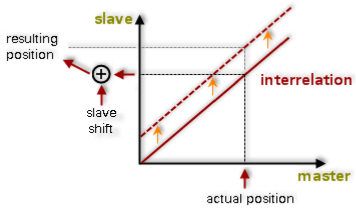
Parameter	Description
	 Figure 42: Representation of a direct start
MasterParID	A ParID can be used as master signal instead of the master axis position set-point.
AddMasterParID	The value of this ParID is added to the master position.
AddSlaveParID	The value of this ParID is added to the slave position calculated by the automat.  Figure 43: Master – coupling relationship – additive element - slave
SlaveFactorParID	The slave axis scaling is extended by the value of this ParID. This factor applies to all states in the automat.
EventParID	ParID specification, which serves as event source in states where the event type ncPAR_ID is used. An event is detected if the value of this ParID changes from 0 to a value != 0.
SlaveLatchParID	The slave compensation distance begins at the latched value of this ParID in the compensation mode ncSL_LATCHPOS. The value of the ParID specified here (INT type) is latched when a trigger occurs (TRIGGER1, TRIGGER2).

Table 5: Overview of optional parameters

Motion control \ ACP10/ARNC0 \ Libraries \ ACP10_MC \ Categorized function blocks \ Cam automat \ General \ Cam automat structure

Motion control \ ACP10/ARNC0 \ Reference manual \ ACOPOS drive functions \ Cam automat \ Direct start

6.4 Application of the Cam Profile Automat

The parameter structure of the MpAxisCamSequencer component contains all the parameters, states and change events of the Cam Profile Automats in structured form.

Procedure	Implementation
Define the global parameters for the automat Master and slave axis, start state, states and change events	Transfer of data via parameter structure of the MpAxisCamSequencer component
Download all of the cam profiles to the drive that are used in the Cam Profile Automat.	Enabling the MpAxisCamSequencer component
After the above steps have been performed, the automat can then be started and operated.	"StartSequence" input of the MpAxisCamSequencer component
Change of cam profiles via external signals, which are analyzed in the change events of the automat configuration.	"Signal1..4" inputs of the MpAxisCamSequencer component

Table 6: Procedure for implementing the Cam Profile Automats in the application

Based of the MC_AUTDATA_TYP data structure, which is a component of the parameter structure of MpAxisCamSequencer, an example is provided for different Cam Profile Automat configurations with Automation Studio.

Motion control \ ACP10/ARNC0 \ Examples \ Motion control \ Cam automat \ Automat configurations

- Automat configuration - Labeling machine
- Automat configuration - Flying saws
- Automation configuration - Cross cutter

Online parameter change: The cam automat parameters can be changed during runtime. Exceptions to this rule are compensation mode (CompMode), event type (Type), event attribute (Attribute) and MasterParID.

Other methods of stopping the automat: Automat operation can be ended at any time by stopping the slave axis (MpAxisBasic.Stop). Changing the state index to 255 can be used to exit the automat.

Motion control \ ACP10/ARNC0 \ Reference manual \ ACOPOS drive functions \ Cam automat \ Cams \ Predefined curves

Motion control \ ACP10/ARNC0 \ Reference manual \ ACOPOS drive functions \ Cam automat \ Online change

Exercise: Control a flying saw with MpAxisCamSequencer

In the previous example, the automat configuration was already imported into the project for a flying saw. Information about the settings for the master and slave axis periods can be found in the respective description.

The cam profile sequence is implemented using the MpAxisCamSequencer function block.

1) Set the axis periods on the master and slave according to the help documentation regarding automat configuration
2) Call "FlyingSaw" automat configuration
3) Call MpAxisCamSequencer
4) Start and home the master and slave axes (already carried out in the preparation of the two individual programs with MpAxis)
5) Start a positive movement on the master
6) Start the Cam Profile Automat on the slave
7) Wait for the "Info.MasterReady" and "Info.SlaveReady" status information
8) Start the sequence using the "StartSequence" command
9) Evaluate the position setpoint for the master and slave axes and record it with NC Trace

See: Calling MpAxisCamSequenzer

- 8.3 "Solution: Controlling a flying saw with MpAxisCamSequencer" on page 49

6.5 Compensation gear

For each state in the Cam Profile Automat, a compensation gear can be used.

The compensation gear is an automatically calculated curve which compensates for position differences during a state transition and ensures smooth transitions between cam profiles. The necessary parameters are provided in the **<MC_AUTDATA_TYP>.State[x]** data structure. (see 6.3.1 "Define the states" on page 35)

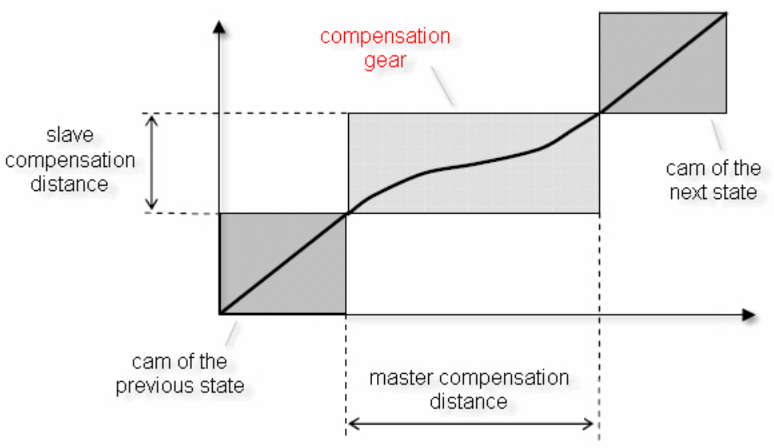


Figure 44: Displays the functionality of a compensation gear

The image shows compensation between two consecutive states (cam profiles). If compensation is used in a state, then the compensation movement is always performed before the cam profile of the state.

These are the base parameters that define the compensation:

- Compensation mode (CompMode)
- Master compensation distance (MasterCompDistance)
- Slave compensation distance (SlaveCompDistance)

The different compensation gear modes provide possibilities for compensating path as well as velocity differences. A precise overview and description can be found in the Automation Studio help system.

Motion control \ ACP10/ARNC0 \ Libraries \ ACP10_MC \ Categorized function blocks \ Cam automat \ General \ Event Compensation gears

Motion control \ ACP10/ ARNC0 \ Reference manual \ ACOPOS drive functions \ Cam automat \ Compensation gears \ Compensation mode \ Methods for compensation of position differences

Motion control \ ACP10/ ARNC0 \ Reference manual \ ACOPOS drive functions \ Cam automat \ Compensation gears \ Compensation mode \ Methods for compensation of speed differences

6.6 Predefined cam profiles and sequences

Predefined cam profiles

Some predefined cam profiles are already available on the drive and do not have to be transferred there.

CamProfileIndex	Description
0xFFFF	This predefined linear cam profile can be used with a master and slave length of one unit as the CamProfileIndex when configuring the automat. This can be used with gauge factors to produce any m:n straight line.
0xFFFE	This predefined point cam profile can be used as the CamProfileIndex when configuring the automat. This point cam profile can only be used when compensation mode is enabled. It cannot be used in states with CompMode = ncOFF. The master and slave interval length of this predefined cam profile is zero. However, the slope of the curve is not zero, but can be set using the gauge factors. This allows you to create applications that only require one compensation procedure without a cam profile.

Table 7: Overview of predefined cam profiles

Motion control \ ACP10/ARNC0 \ Libraries \ ACP10_MC \ Categorized function blocks \ Cams \ General \ Cams

Predefined processes

Applications can be created more easily using predefined processes and the predefined cam profiles on the drive.

Function block	Description
MC_BR_CamDwell	Cam profile coupling between master and slave axes, dwell times can be defined, configurable entry and exit movements, definable compensation distances
MC_BR_AutoCamDwell	Like MC_BR_CamDwell, a cam profile does not have to be specified, the slave drive calculates a jolt-minimized movement profile

Table 8: Overview of predefined processes

Function block	Description
MC_BR_CamTransition	Cyclically repeating cam profile compensation process with optional entry and exit transition
MC_BR_CrossCutterControl	Control and correct cutting axes, combination of phasing and offset possible

Table 8: Overview of predefined processes

Motion control \ ACP10/ARNC0 \ Libraries \ ACP10_MC \ Categorized function blocks \ Cams \ Predefined processes

7 Summary

mapp technology components of the MpAxis library contain user friendly components for drive coupling.

The individual components are designed in accordance with the PLCopen Motion Control standard and feature a uniform design with regard to functional usage.

The electronic gear makes it possible to implement linear position couplings, even with a defined starting point for the axes, if necessary.

Corresponding function blocks are also provided for nonlinear position couplings using electronic cam profiles. The application program controls interactions between multiple cam profiles.

Cam profiles can be created using the cam profile editor in Automation Studio. A wide variety of settings makes it easy to tailor a cam profile to the demands of a particular process. Predefined cam profiles on the drive expand the range of functions.

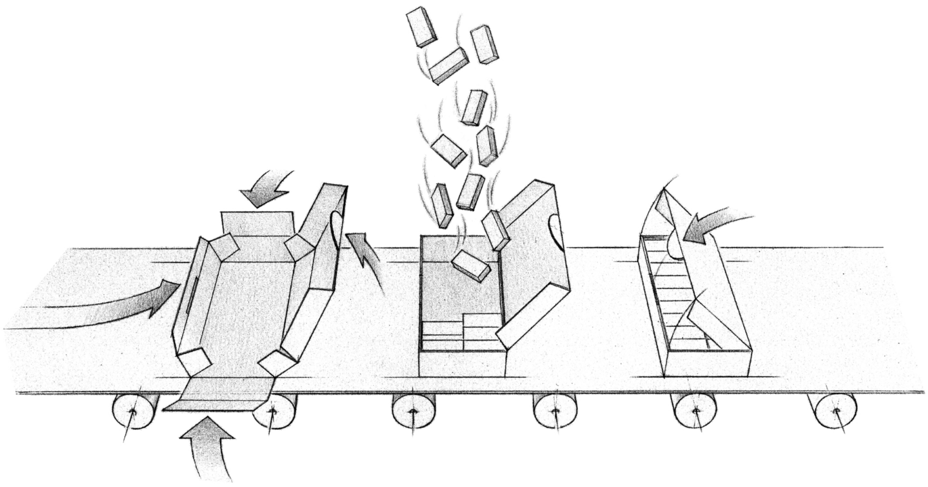


Figure 45: Schematic illustration of a cartoning application

The Cam Profile Automat is an extremely powerful tool for effectively linking cam profiles. The necessary sequences are completely predefined. Initialization of the automat structure and control of the automat mode can be handled using clear and organized functions. Once the Cam Profile Automat is started, the defined sequences are independently processed on the drive. This reduces the load on the application program and results in a very fast, event-controlled positioning sequence.

The MpAxis multi-axis functions are subject to the effects of the states in the PLCopen Motion Control state diagram. The user is provided with necessary information for planning the sequence here.

8 Solutions

8.1 Solution: Configure a simple linear axis coupling

```
PROGRAM _CYCLIC
    CASE sStep OF
        enCONFIGURATION:
            (*configure MpAxisCoupling*)
            MpAxisCoupling_0.MpLink := ADR(gMpLink_AxisBasis_gAxis2Slave);
            MpAxisCoupling_0.Enable := TRUE;
            MpAxisCoupling_0.Parameters := ADR(CouplingParameters);
            MpAxisCoupling_0.MpLinkMaster := ADR(gMpLink_AxisBasis_gAxis1);

            (*set parameters for gear ratio 1:1*)
            CouplingParameters.RatioNumerator := 1.0;
            CouplingParameters.RatioDenominator := 1.0;
            CouplingParameters.Gear.Acceleration := 2000;
            CouplingParameters.Gear.Deceleration := 2000;

            IF (MpAxisCoupling_0.Active = TRUE) THEN
                (*check if master and slave are ready for gear*)
                IF (MpAxisCoupling_0.Info.MasterReady = TRUE)
                    AND (MpAxisCoupling_0.Info.SlaveReady = TRUE) THEN
                    sStep := enOPERATION;
                END_IF
            END_IF

        enOPERATION:
            (*command interface*)
            MpAxisCoupling_0.Gear := cmdStartGear;
            MpAxisCoupling_0.Update := cmdUpdate;

            IF MpAxisCoupling_0.Error = TRUE THEN
                cmdStartGear := FALSE;
                cmdUpdate := FALSE;
                sStep := enERROR;
            END_IF

        enERROR:
            (*implement error handling here*)
            MpAxisCoupling_0.ErrorReset := cmdErrorReset;
            IF MpAxisCoupling_0.Error = FALSE THEN
                sStep := enCONFIGURATION;
            END_IF
    END_CASE

    (*call all mapp technology components*)
    MpAxisCoupling_0();
END_PROGRAM
```

8.2 Solution: Axis coupling using a cam profile

```
PROGRAM _CYCLIC
    CASE sStep OF

        enCONFIGURATION:
            (*configure MpAxisCoupling*)
            MpAxisCoupling_0.MpLink := ADR(gMpLink_AxisBasis_gAxis2Slave);
            MpAxisCoupling_0.Enable := 1;
            MpAxisCoupling_0.Parameters := ADR(CouplingParameters);
            MpAxisCoupling_0.MpLinkMaster := ADR(gMpLink_AxisBasis_gAxis1);

            (*set parameters for cam*)
            (* note axis period is 3600 Units for master and slave axis
               slave runs profile 3 times each master period *)
            CouplingParameters.RatioNumerator := 1200.0;
            CouplingParameters.RatioDenominator := 1200.0;
            CouplingParameters.Cam.MasterStartPosition := 0;

            CouplingParameters.Cam.CamTableID := 0; (*use cam index 0*)
            CouplingParameters.Cam.CamTable[0] := 'profile';

            (*check if master and slave are ready for gear*)
            IF (MpAxisCoupling_0.Info.MasterReady = TRUE)
                AND (MpAxisCoupling_0.Info.SlaveReady = TRUE) THEN
                sStep := enOPERATION;
            END_IF

        enOPERATION:
            (*command interface*)
            MpAxisCoupling_0.Cam := cmdStartCam;
            MpAxisCoupling_0.Update := cmdUpdate;

            IF MpAxisCoupling_0.Error = TRUE THEN
                cmdStartCam := FALSE;
                cmdUpdate := FALSE;
                sStep := enERROR;
            END_IF

        enERROR:
            (*implement error handling here*)
            MpAxisCoupling_0.ErrorReset := cmdErrorReset;
            IF MpAxisCoupling_0.Error = FALSE THEN
                sStep := enCONFIGURATION;
            END_IF

    END_CASE

    (*call all mapp components*)
    MpAxisCoupling_0();
END_PROGRAM
```

8.3 Solution: Controlling a flying saw with MpAxisCamSequencer

8.3.1 Cyclic program with MpAxisCamSequencer

```
PROGRAM _CYCLIC
    CASE sStep OF
        enCONFIGURATION:
            (*set parameters for flying saw cam automat*)
            FlyingSaw(AutConfiguration.Configuration);

            (*configure MpAxisCoupling*)
            MpAxisCamSequencer_0.MpLink := ADR(gMpLink_AxisBasis_gAxis2Slave);
            MpAxisCamSequencer_0.Enable := 1;
            MpAxisCamSequencer_0.Parameters := ADR(AutConfiguration);
            MpAxisCamSequencer_0.MpLinkMaster := ADR(gMpLink_AxisBasis_gAxis1);

            (*check if master and slave are ready for cam sequence*)
            IF (MpAxisCamSequencer_0.Info.MasterReady = TRUE)
                AND (MpAxisCamSequencer_0.Info.SlaveReady = TRUE) THEN
                sStep := enOPERATION;
            END_IF

        enOPERATION:
            (*command interface*)
            MpAxisCamSequencer_0.StartSequence := cmdStartSequence;
            MpAxisCamSequencer_0.Update := cmdUpdate;


        enERROR:

            (*implement error handling here*)
            MpAxisCamSequencer_0.ErrorReset := cmdErrorReset;
            IF MpAxisCamSequencer_0.Error = FALSE THEN
                sStep := enCONFIGURATION;
            END_IF
    END_CASE

    (*call all mapp components*)
    MpAxisCamSequencer_0();

END_PROGRAM
```

8.3.2 Automat configuration for a flying saw

```
FUNCTION FlyingSaw
    (*general automat parameters*)
    AutData.StartPosition       := 0;
    AutData.StartInterval       := 1000;
    AutData.MaxMasterVelocity   := 1000;

    (*Automat STATE 0 Basis State*)
    AutData.State[0].Event[0].Type      := ncST_END;
    AutData.State[0].Event[0].Attribute := ncAT_ONCE;
    AutData.State[0].Event[0].NextState := 1;

    (*Automat STATE 1 Standstill*)
    AutData.State[1].CamProfileIndex    := 16#FFFF;
    AutData.State[1].MasterFactor       := 200;
    AutData.State[1].SlaveFactor        := 0;
    AutData.State[1].CompMode           := ncONLYCOMP;
    AutData.State[1].MasterCompDistance := 800;
    AutData.State[1].SlaveCompDistance  := 0;

    AutData.State[1].Event[0].Type      := ncS_START;
    AutData.State[1].Event[0].Attribute := ncAT_ONCE;
    AutData.State[1].Event[0].NextState := 2;

    (*Autoamt STATE 2 Synchronous movement*)
    AutData.State[2].CamProfileIndex    := 16#FFFF;
    AutData.State[2].MasterFactor       := 500;
    AutData.State[2].SlaveFactor        := 500;
    AutData.State[2].CompMode           := ncONLYCOMP;
    AutData.State[2].MasterCompDistance := 100;
    AutData.State[2].SlaveCompDistance  := 100;

    AutData.State[2].Event[0].Type      := ncST_END;
    AutData.State[2].Event[0].Attribute := ncST_END;
    AutData.State[2].Event[0].NextState := 3;

    (*Automat STATE 3 Return to starting position*)
    AutData.State[3].CamProfileIndex    := 16#FFFF;
    AutData.State[3].MasterFactor       := 100;
    AutData.State[3].SlaveFactor        := 0;
    AutData.State[3].CompMode           := ncONLYCOMP;
    AutData.State[3].MasterCompDistance := 800;
    AutData.State[3].SlaveCompDistance  := -600;

    AutData.State[3].Event[0].Type      := ncST_END;
    AutData.State[3].Event[0].Attribute := ncST_END;
    AutData.State[3].Event[0].NextState := 1;

    FlyingSaw := TRUE;
END_FUNCTION
```

Offered by the Automation Academy

The Automation Academy provides targeted training courses for our customers as well as our own employees.
At the Automation Academy, you'll develop the skills you need in no time!
Our seminars make it possible for you to improve your knowledge in the field of automation engineering. Once completed, you will be in a position to implement efficient automation solutions using B&R technology. This will make it possible for you to secure a decisive competitive edge by allowing you and your company to react faster to constantly changing market demands.

Seminars

Quality and relevance are essential components of our seminars. The pace of a specific seminar is based strictly on the experience that course participants bring with them and tailored to the requirements they face. A combination of group work and self-study provides the high level of flexibility needed to maximize the learning experience.
Each seminar is taught by one of our highly skilled and experienced trainers.

Training modules

Our training modules provide the basis for learning both at seminars as well as for self-study. These compact modules rely on a consistent didactic concept. Their bottom-up structure allows complex, interrelated topics to be learned efficiently and effectively. They serve as the best possible companion to our extensive help system. The training modules are available as downloads and can be ordered as printed versions.

Topic categories:

- ⇨ Control technology
- ⇨ Motion control
- ⇨ Safety technology
- ⇨ HMI
- ⇨ Process control
- ⇨ Diagnostics and service
- ⇨ POWERLINK and openSAFETY

ETA system

The ETA system provides realistic constructions for training, education and laboratory use. Two different basic mechanical constructions can be selected. The ETA light system offers a high degree of mobility, saves space and is well-suited for lab work. The ETA standard system has a sturdy mechanical structure and includes pre-wired sensors and actuators.

Find out more!

Do you require additional training? Are you interested in finding out what the B&R Automation Academy has on offer? You've come to the right place.
Detailed information can be found under the following links:
www.br-automation.com/academy

TM441TRE.433-ENG