SIEMENS

Ingenuity for life

Creation of cam disks at runtime for S7-1500T

Library LCamHdl – Advanced cam creation

https://support.industry.siemens.com/cs/ww/en/view/105644659

Siemens Industry Online Support

Legal information

Use of application examples

Application examples illustrate the solution of automation tasks through an interaction of several components in the form of text, graphics and/or software modules. The application examples are a free service by Siemens AG and/or a subsidiary of Siemens AG ("Siemens"). They are non-binding and make no claim to completeness or functionality regarding configuration and equipment. The application examples merely offer help with typical tasks; they do not constitute customer-specific solutions. You yourself are responsible for the proper and safe operation of the products in accordance with applicable regulations and must also check the function of the respective application example and customize it for your system.
Siemens grants you the non-exclusive, non-sublicensable and non-transferable right to have the application examples used by technically trained personnel. Any change to the application examples is your responsibility. Sharing the application examples with third parties or copying the application examples or excerpts thereof is permitted only in combination with your own products. The application examples are not required to undergo the customary tests and quality inspections of a chargeable product; they may have functional and performance defects as well as errors. It is your responsibility to use them in such a manner that any malfunctions that may occur do not result in property damage or injury to persons.

Disclaimer of liability

Siemens shall not assume any liability, for any legal reason whatsoever, including, without limitation, liability for the usability, availability, completeness and freedom from defects of the application examples as well as for related information, configuration and performance data and any damage caused thereby. This shall not apply in cases of mandatory liability, for example under the German Product Liability Act, or in cases of intent, gross negligence, or culpable loss of life, bodily injury or damage to health, non-compliance with a guarantee, fraudulent non-disclosure of a defect, or culpable breach of material contractual obligations. Claims for damages arising from a breach of material contractual obligations shall however be limited to the foreseeable damage typical of the type of agreement, unless liability arises from intent or gross negligence or is based on loss of life, bodily injury or damage to health. The foregoing provisions do not imply any change in the burden of proof to your detriment. You shall indemnify Siemens against existing or future claims of third parties in this connection except where Siemens is mandatorily liable.
By using the application examples you acknowledge that Siemens cannot be held liable for any damage beyond the liability provisions described.

Other information

Siemens reserves the right to make changes to the application examples at any time without notice. In case of discrepancies between the suggestions in the application examples and other Siemens publications such as catalogs, the content of the other documentation shall have precedence.
The Siemens terms of use (https://support.industry.siemens.com) shall also apply.

Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, systems, machines and networks.
In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial security concept. Siemens' products and solutions constitute one element of such a concept.
Customers are responsible for preventing unauthorized access to their plants, systems, machines and networks. Such systems, machines and components should only be connected to an enterprise network or the Internet if and to the extent such a connection is necessary and only when appropriate security measures (e.g. firewalls and/or network segmentation) are in place.
For additional information on industrial security measures that may be implemented, please visit https://www.siemens.com/industrialsecurity.
Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends that product updates are applied as soon as they are available and that the latest product versions are used. Use of product versions that are no longer supported, and failure to apply the latest updates may increase customer's exposure to cyber threats.
To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed at: https://www.siemens.com/industrialsecurity.

Table of contents

1 Library Overview

What you get

This document describes the block LCamHdl_CreateCamAdvanced of the LCamHdl block library. The block library provides you with the tested code with clearly defined interfaces. They can be used as a basis for your task to be implemented.

A key concern of the document is to describe

- all blocks of the block library
- the functionality implemented through these blocks.

Furthermore, this documentation shows possible fields of application and helps you integrate the library into your STEP 7 project using step-by-step instructions.

Scope of application

- STEP 7 Professional V17
- Motion Control V6.0
- S7-1500T CPU as of firmware V2.9

1.1 Different user scenarios

Possible application(s) for the LCamHdl library

The present application is to support the user with the configuration of cam disks by applying the laws of motion.

In general, cam disks are electronic gears at a non-constant transition, where, for example, a constant drive motion is converted into a non-constant drive motion by applying the laws of motion.

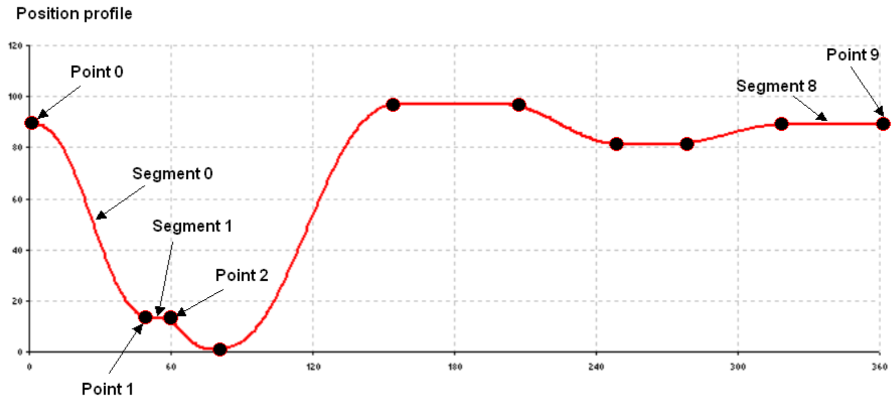
Within the scope of the SIMATIC S7-1500T, there are two ways to configure cam disks:

- at engineering in the TIA Portal with the help of the cam editor
- at runtime by definition of a cam profile

The present application is dedicated to the configuration of cam disks at runtime.

The SIMATIC S7-1500T makes it possible to generate complex cam disks with n profiles, with n+1 points and / or m single points at runtime. For this, you can freely choose the law of motion for each single element

Figure 1-1: Cam disk consisting of several elements



If the working ranges change during the operation of a machine by, for example, a changed length of the product, it is necessary to adapt the motion transitions as well.

The following section shows scenarios for a possible application of the LCamHdl library:

1.1.1 Scenario 1

A fully defined cam disk shall be created at runtime. Points in the cam disk and the according dynamics are known. Transitions can be made via straight lines and 5th degree polynomials, taken into account velocity and acceleration.

You should choose the **LCamHdl_CreateCamBasic** function block to create the cam disk. It eases the cam disk creation for cam disks with interpolation algorithms up to 5th degree polynomials.

NOTE Further information can be found in \2\.

Figure 1-2: Cam disk with 8 points created by LCamHdl_CreateCamBasic

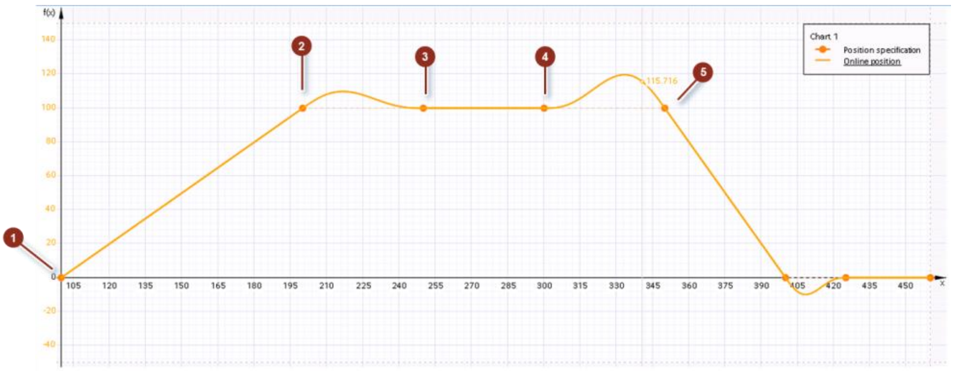


Figure 1-3: Configuration of cam with 8 points created by LCamHdl_CreateCamBasic

Cam4_Basic	Array[1..8] of "LCa...	
Cam4_Basic[1]	"LCamHdl_typeBas...	
leadingValue	LReal	100.0
followingValue	LReal	0.0
velocityRatio	LReal	1.0
accelerationRa...	LReal	0.0
Cam4_Basic[2]	"LCamHdl_typeBas...	
leadingValue	LReal	200.0
followingValue	LReal	100.0
velocityRatio	LReal	1.0
accelerationRa...	LReal	0.0
Cam4_Basic[3]	"LCamHdl_typeBas...	
leadingValue	LReal	250.0
followingValue	LReal	100.0
velocityRatio	LReal	0.0
accelerationRa...	LReal	0.0
Cam4_Basic[4]	"LCamHdl_typeBas...	
leadingValue	LReal	300.0
followingValue	LReal	100.0
velocityRatio	LReal	0.0
accelerationRa...	LReal	0.0
Cam4_Basic[5]	"LCamHdl_typeBas...	
leadingValue	LReal	350.0
followingValue	LReal	100.0
velocityRatio	LReal	-2.0
accelerationRa...	LReal	0.0
Cam4_Basic[6]	"LCamHdl_typeBas...	
Cam4_Basic[7]	"LCamHdl_typeBas...	
Cam4_Basic[8]	"LCamHdl_typeBas...	

1.1.2 Scenario 2

The function block (FB) **LCamHdl_CreateCamAdvanced** can be used to merge working ranges and motion transitions into one cam disk at runtime. Unlike directly assigning the cam's data block, the FB can be used without having to calculate the polynomial coefficients before

The FB is based on the motion rules for cam mechanisms according to VDI 2143.

The cam profile configuration of the position as well as the geometric derivations is made in the real section (e.g. velocity, acceleration, jerk).

There are different mathematic functions available for the motion transitions (elements), subsequently called profile types. Besides polynomials like

- 3rd degree polynomial,
- 5th degree polynomial,
- 7th degree polynomial

further profiles exist

- straight line,
- quadratic parabola,
- basic sine,
- inclined sine,
- modified acceleration trapezoid,
- modified sine,
- sine-straight line-combination – velocity trapezoid,
- harmonic combination

In addition to that, it is also possible to transfer single points, which makes it possible to generate cam disks with combined ranges consisting of transition functions and single points.

As an extension to the profiles listed above, there are additional function blocks in the LCamHdl library. These function blocks expand the **LCamHdl_CreateCamAdvanced** by the following profiles:

- sine profile
- inverse sine profile (arc sine)
- polynomial profile with trigonometric portion
- double harmonic sine profile ($cosine^4$)

NOTE Support for the additional function blocks **LCamHdl_AddCamSine**, **LCamHdl_AddCamInvSine**, **LCamHdl_AddCamPolynomial** and **LCamHdl_AddCamDblHarmonic** can be found in the manual "LCamHdl Advanced cam creation – Additional profiles" \2\

The additional function blocks can be used combined with the LCamHdl_CreateCamAdvanced or as a standalone function.

NOTE

One cam element will result in up to six cam segments, depending on the profile type.

Keep in mind the maximum number of cam segments (50) and check with the resulting number of cam segments due to the user profile – see Table 2-7.

Also keep in mind that the maximum number of cam points is limited (TO_Cam: 1000, TO_Cam_10k: 10000).

In difference to the **LCamHdl_CreateCamBasic** block the function block **LCamHdl_CreateCamAdvanced** works segment based. This allows gaps (filled by runtime interpolation) between segments and also the usage of the points array in the cam technology object.

Figure 1-4: Cam disk with 4 segments created with LCamHdl_CreateCamAdvanced

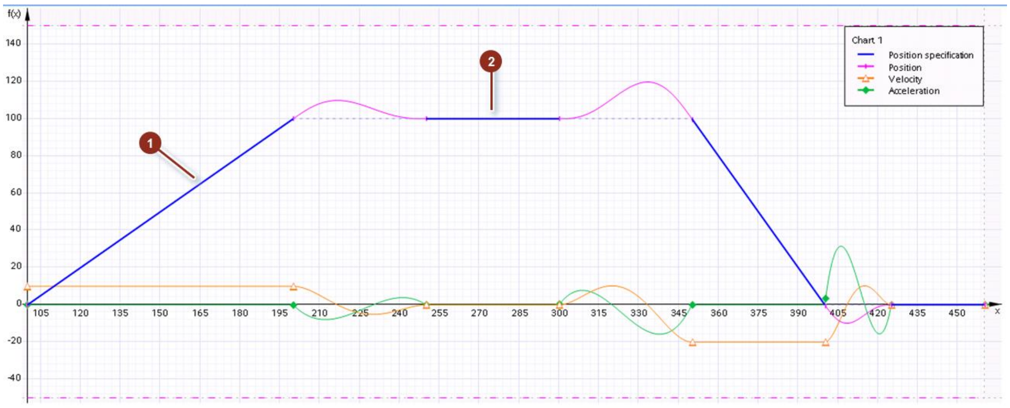


Figure 1-5: Configuration of cam with 4 segments created by CamHdl_CreateCamAdvanced

Cam3_Advanced	Array[1..4] of "LCa...	
(1) Cam3_Advanced[1]	"LCamHdl_typeAdv...	
leadingValueStart	LReal	100.0
leadingValueEnd	LReal	200.0
followingValueStart	LReal	0.0
followingValueEnd	LReal	100.0
geoVeloStart	LReal	0.0
geoVeloEnd	LReal	0.0
geoAccelStart	LReal	0.0
geoAccelEnd	LReal	0.0
geoJerkStart	LReal	0.0
geoJerkEnd	LReal	0.0
inflectionPointPara...	LReal	0.5
modVeloTrapezoid...	LReal	1.0
modSineMaxAccel...	LReal	0.0
camProfileType	DInt	"LCAMHDL_PROFILE_CONST_
(2) Cam3_Advanced[2]	"LCamHdl_typeAdv...	
leadingValueStart	LReal	250.0
leadingValueEnd	LReal	300.0
followingValueStart	LReal	100.0
followingValueEnd	LReal	100.0
geoVeloStart	LReal	0.0
geoVeloEnd	LReal	0.0
geoAccelStart	LReal	0.0
geoAccelEnd	LReal	0.0
geoJerkStart	LReal	0.0
geoJerkEnd	LReal	0.0
inflectionPointPara...	LReal	0.5
modVeloTrapezoid...	LReal	1.0
modSineMaxAccel...	LReal	0.0
camProfileType	DInt	"LCAMHDL_PROFILE_DWELL"

1.1.3 Scenario 3

A cam disk based on interpolation points is to be created at runtime. Only the X and Y coordinates of the interpolation points are known (X - master, Y - slave).

You should choose the **LCamHdl_CreateCamBasedOnXYPoints** function block. It eases the cam disk creation for cam disks consisting of just interpolation points.

The interpolation mode (linear / C splines / B splines) can be defined via the TO-Cam DB - *TO-Cam.InterpolationSettings.InterpolationMode*.

Figure 1-6: Cam disk example (interpolation mode C splines) created by LCamHdl_CreateCamBasedOnXYPoints

NOTE	Further information can be found in \2\.

1.1.4 Explanations of the different profile types

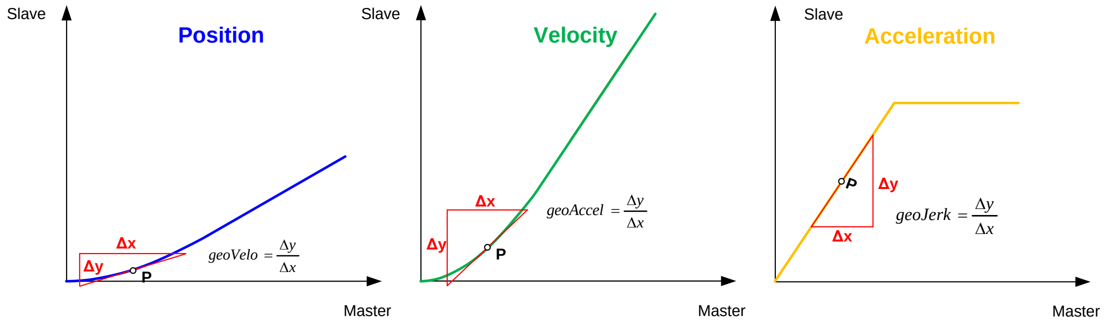
The following table shows all available profile types. The user has to consider the necessary boundary conditions.

Please pay special attention to the subject concerning reversal of velocity (problematical with some machine types) as well as to the meaning of „geometric velocity / acceleration / …“.

Derivations in the boundary points

The derivations have to be parameterized with real values, the standardization described in the laws of motion according to VDI 2143 does not have to be taken into account by the user.

Figure 1-7: Derivations in the boundary points



Inflection point

Symmetric laws of motion are characterized by an inflection point at $\lambda = 0{,}5$ within the leading range (x_0-x_1).

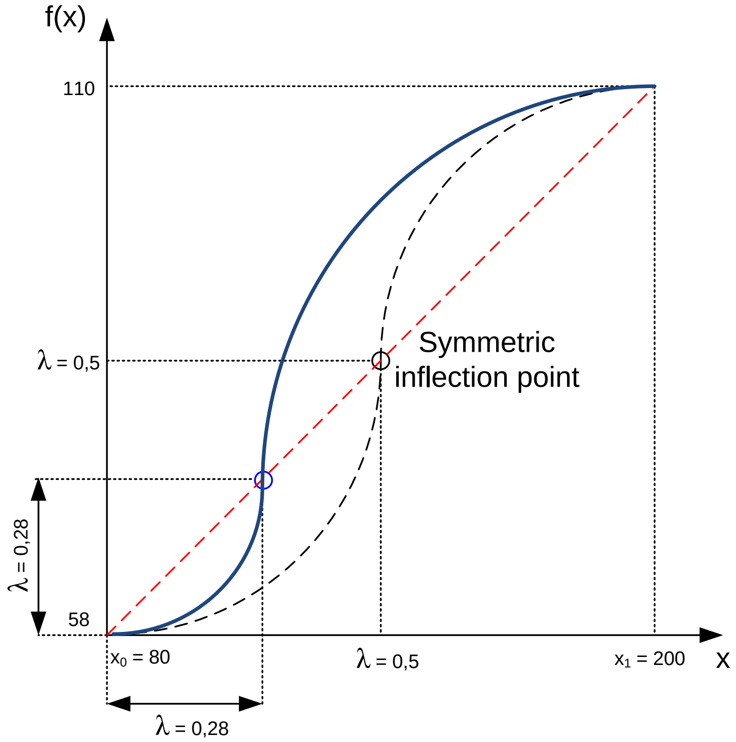
Asymmetric laws of motion are created by displacing the inflection point e.g. by shifting on the straight line with the function:

$$f(x) = \frac{\Delta y}{\Delta x} x$$

Resulting from this, there are different maximum acceleration values Ca in the first part of the profile and Ca* in the second. Furthermore the point of the maximum velocity displaces in the same way as the inflection point in this profile.

For the new ranges $0 \leq x \leq \lambda$ and $\lambda \leq x \leq 1$ there are enlarged and reduced parts of symmetric laws of motion applied

Figure 1-8: Asymmetric law of motion by displacing the inflection point



Explanation of characteristic values

Cv = Cmstat	maximum value of velocity (= static torque characteristic)
Ca	maximum value of acceleration (first half of profile, if asymmetrical)
Ca*	maximum value of acceleration (second half of profile, if asymmetrical)
Cj	maximum value of jerk
Cmdyn	dynamic torque characteristic

List of profile types

Please consider that for the different profile types only certain boundary conditions are supported. All other parameters are not taken into account.

Table 1-1: Explanations of the different profile types

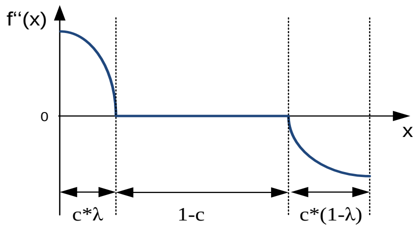
Profile type	General information	Supported boundary condition parameters
EMPTY (default): LCAMHDL_PROFILE_ EMPTY	**Application:** Empty element – not used in current cam **Number of required TO cam segments:** 0	
POINT: LCAMHDL_PROFILE_ POINT	**Application:** Definition of a point with undefined dynamics **Number of required TO cam points:** 1	*leadingValueStart* *followingValueStart*
DWELL: LCAMHDL_PROFILE_ DWELL	**Application:** Standstill **Number of required TO cam segments:** 1 **Requirement:** leadingValueStart < leadingValueEnd	*leadingValueStart* *followingValueStart* *leadingValueEnd* *followingValueEnd*
CONST_VELO: Straight line LCAMHDL_PROFILE_ CONST_VELO	**Application:** Constant velocity **Number of required TO cam segments:** 1 **Disadvantages:** Motion with impacts, vibrations, noise, wear **Requirement:** *leadingValueStart < leadingValueEnd*	*leadingValueStart* *followingValueStart* *leadingValueEnd* *followingValueEnd*

Profile type	General information	Supported boundary condition parameters
POLY_3: 3rd degree polynomial LCAMHDL_PROFILE_ POLY_3	**Application:** Transition at continuous velocity **Number of required TO cam segments:** λ = 0.5: 1 λ <> 0.5: 2 **Addition:** Displacement of the inflection point (λ) λ = 0.5: The position of the inflection point is calculated automatically. An inflection point at λ = 0.5 can be "enforced" by using POLY_4. λ <> 0.5: Two 3rd degree polynomial segments are created. This leads to a discontinuity in jerk within the profile. Recommendation: Use POLY_4 instead. **Advantages:** Can be used for many motion tasks – presetting of position and velocity **Disadvantages:** Overshoot possible **Requirement:** *leadingValueStart < leadingValueEnd*	*leadingValueStart* *followingValueStart* *leadingValueEnd* *followingValueEnd* *geoVeloStart* *geoVeloEnd* **Addition:** *0 < λ < 1* λ: *inflectionPointParameter*
POLY_4 4th degree polynomial LCAMHDL_PROFILE_ POLY_4	**Application:** Transition at continuous velocity. **Number of required TO cam segments:** 1 **Addition:** Displacement of the inflection point (λ) Recommendation: Use POLY_3 with λ = 0.5 for automatic calculation of the inflection point. **Advantages:** Can be used for many motion tasks – presetting of position and velocity **Disadvantages:** Overshoot possible **Requirement:** *leadingValueStart < leadingValueEnd*	*leadingValueStart* *followingValueStart* *leadingValueEnd* *followingValueEnd* *geoVeloStart* *geoVeloEnd* **Addition:** -∞ < λ < ∞ λ: *inflectionPointParameter*

Profile type	General information	Supported boundary condition parameters
POLY_5: 5th degree polynomial LCAMHDL_PROFILE_POLY_5	**Application:** Continuous transition of velocity and acceleration **Number of required TO cam segments:** λ = 0.5: 1 λ <> 0.5: 2 **Addition:** Displacement of the inflection point (λ) λ = 0.5: The position of the inflection point is calculated automatically. An inflection point at λ = 0.5 can be "enforced" by using POLY_6. λ <> 0.5: Two 4th degree polynomial segments are created. This leads to a discontinuity in jerk within the profile. Recommendation: Use POLY_6 instead. **Advantages:** Can be used for nearly all motion tasks – presetting of position, velocity and acceleration **Disadvantages:** Overshoot possible **Requirement:** *leadingValueStart < leadingValueEnd*	*leadingValueStart* *followingValueStart* *leadingValueEnd* *followingValueEnd* *geoVeloStart* *geoVeloEnd* *geoAccelStart* *geoAccelEnd* **Addition:** *0 < λ < 1* *λ: inflectionPointParameter*
POLY_6 6th degree polynomial LCAMHDL_PROFILE_POLY_6	**Application:** Continuous transition of velocity and acceleration **Number of required TO cam segments:** 1 **Addition:** Displacement of the inflection point (λ) Recommendation: Use POLY_5 with λ = 0.5 for automatic calculation of the inflection point. **Advantages:** Can be used for nearly all motion tasks – presetting of position, velocity and acceleration **Disadvantages:** Overshoot possible **Requirement:** *leadingValueStart < leadingValueEnd*	*leadingValueStart* *followingValueStart* *leadingValueEnd* *followingValueEnd* *geoVeloStart* *geoVeloEnd* *geoAccelStart* *geoAccelEnd* **Addition:** -∞ < λ < ∞ *λ: inflectionPointParameter*

Profile type	General information	Supported boundary condition parameters
POLY_7: 7th degree polynomial LCAMHDL_PROFILE_ POLY_7	**Application:** Continuous transition of velocity, acceleration and jerk **Number of required TO cam segments:** 2 **Addition:** Displacement of the inflection point (λ) **Advantages:** Can be used for nearly all motion tasks – presetting of position, velocity, acceleration and jerk **Disadvantages:** Overshoot possible The 7th degree polynomial is implemented internally by using two 6th degree polynomials. **Requirement:** *leadingValueStart* < *leadingValueEnd*	*leadingValueStart* *followingValueStart* *leadingValueEnd* *followingValueEnd* *geoVeloStart* *geoVeloEnd* *geoAccelStart* *geoAccelEnd* *geoJerkStart* *geoJerkEnd* **Addition:** *0 < λ < 1* λ: *inflectionPointParameter*
PARABOLA: LCAMHDL_PROFILE_ D_D_PARABOLA LCAMHDL_PROFILE_ C_C_PARABOLA	**Application:** dwell - dwell constant velocity - constant velocity **Number of required TO cam segments:** 2 **Addition:** Displacement of the inflection point (λ) **Advantages:** Lowest Ca value for this kind of motion transitions **Disadvantages:** Vibration, noise and wear due to acceleration jumps **Requirement:** *leadingValueStart* < *leadingValueEnd*	*leadingValueStart* *followingValueStart* *leadingValueEnd* *followingValueEnd* *geoVeloStart* *geoVeloEnd* **Addition:** *0 < λ < 1* λ: *inflectionPointParameter*

Profile type	General information	Supported boundary condition parameters
BASIC_SINE: Basic sine oscillation LCAMHDL_PROFILE_D_D_BASIC_SINE LCAMHDL_PROFILE_R_R_BASIC_SINE	**Application:** dwell - dwell reversal - reversal **Number of required TO cam segments:** 2 **Addition:** Displacement of the inflection point (λ) **Advantages:** Low Cv-value for this kind of transitions Low Ca-value for this kind of transitions **Disadvantages:** Acceleration jumps, vibrations, noise and wear **Requirement:** *leadingValueStart* < *leadingValueEnd* *followingValueStart* <> *followingValueEnd*	*leadingValueStart* *followingValueStart* *leadingValueEnd* *followingValueEnd* **Addition:** *0 < λ < 1* λ: *inflectionPointParameter*

Profile type	General information	Supported boundary condition parameters
MOD_VELO_ TRAPEZOID: Sine-straight line-combination LCAMHDL_PROFILE_ R_R_MOD_VELO_ TRAPEZOID	**Application:** reversal - reversal Content of sine in profile: *0 < c <= 1* *c: modVeloTrapezoidParameter*  **Example:** c = 0,3 30% sine (split up among start and ending) 70% straight line λ = 0,4 → displacement towards starting point **Number of required TO cam segments:** c = 1: 2 c <> 1: 3 **Addition:** 1. Displacement of the inflection point (λ) 2. If λ = -1, parameterization of starting and ending acceleration necessary for calculation of λ and c **Advantage:** Especially low Cv-value for this kind of transitions **Disadvantage:** Ca-value is higher compared to basic sine **Requirement:** *leadingValueStart < leadingValueEnd* *followingValueStart <> followingValueEnd*	*leadingValueStart* *followingValueStart* *leadingValueEnd* *followingValueEnd* **Addition:** *1. 0 < λ < 1* *λ: inflectionPointParameter* *2. λ = -1* *inflectionPointParameter = -1* *geoAccelStart <> 0* *geoAccelEnd <> 0*

Profile type	General information	Supported boundary condition parameters
INCLINED_SINE: Inclined sine LCAMHDL_PROFILE_D_D_INCLINED_SINE	**Application:** dwell - dwell **Number of required TO cam segments:** 2 **Addition:** Displacement of the inflection point (λ) **Advantage:** Extra low Cj-value, low vibration, well adapted for high speed **Disadvantage:** Cv, Ca, Cmdyn values greater than in POLY_5 **Requirement:** *leadingValueStart < leadingValueEnd*	*leadingValueStart* *followingValueStart* *leadingValueEnd* *followingValueEnd* *geoVeloStart = 0* *geoVeloEnd = 0* **Addition:** *0 < λ < 1* *λ: inflectionPointParameter*
MOD_ACCEL_TRAPEZOID: Modified acceleration trapezoid LCAMHDL_PROFILE_D_D_MOD_ACCEL_TRAPEZOID LCAMHDL_PROFILE_D_R_MOD_ACCEL_TRAPEZOID LCAMHDL_PROFILE_R_D_MOD_ACCEL_TRAPEZOID	**Application:** dwell - dwell dwell - reversal reversal - dwell **Number of required TO cam segments:** dwell - dwell: 6 dwell - reversal / reversal - dwell: 5 **Addition:** dwell - dwell Displacement of the inflection point (λ) dwell - reversal / reversal - dwell 1. Displacement of the inflection point (λ) 2. If λ = -1, parameterization of starting or ending acceleration necessary for calculation of λ **Advantage:** Extra low Ca-value for this kind of transitions **Disadvantage:** Cv, Cj and Cmdyn values are higher compared to harmonic combination **Requirement:** *leadingValueStart < leadingValueEnd*	*leadingValueStart* *followingValueStart* *leadingValueEnd* *followingValueEnd* *geoVeloStart = 0* *geoVeloEnd = 0* **Addition:** dwell - dwell *0 < λ < 1* *λ: inflectionPointParameter* dwell – reversal *1. 0 < λ < 1* *λ: inflectionPointParameter* *2. λ = -1* *inflectionPointParameter = -1* *geoAccelEnd <> 0* reversal – dwell *1. 0 < λ < 1* *λ: inflectionPointParameter* *2. λ = -1* *inflectionPointParameter = -1* *geoAccelStart <> 0*

Profile type	General information	Supported boundary condition parameters
MOD_SINE: Sinusoidal acceleration LCAMHDL_PROFILE_D_D_MOD_SINE LCAMHDL_PROFILE_C_C_MOD_SINE LCAMHDL_PROFILE_D_C_MOD_SINE LCAMHDL_PROFILE_C_D_MOD_SINE	**Application:** dwell - dwell constant velocity - constant velocity dwell - constant velocity constant velocity - dwell **Number of required TO cam segments:** 4 **Addition:** dwell - dwell Displacement of the inflection point (λ) All others 1. Displacement of the inflection point (λ) 2. If $\lambda = -1$, parameterization of Ca* necessary for calculation of λ **Advantage:** Well adapted for high speed, low Cv, Ca and Cmdyn values **Disadvantage:** Cj-value is higher compared to inclined sine **Requirement:** *leadingValueStart < leadingValueEnd* *followingValueStart <> followingValueEnd*	*leadingValueStart* *followingValueStart* *leadingValueEnd* *followingValueEnd* *geoVeloStart* *geoVeloEnd* dwell – dwell *geoVeloStart = 0* *geoVeloEnd = 0* **Addition:** dwell - dwell $0 < \lambda < 1$ λ: *inflectionPointParameter* All others 1. $0 < \lambda < 1$ λ: *inflectionPointParameter* 2. $\lambda = -1$ *inflectionPointParameter = -1* *modSineMaxAccelCaStar <> 0*

Profile type	General information	Supported boundary condition parameters
HARMONIC_ COMBINATION: Harmonic combination consists of harmonic parts only LCAMHDL_PROFILE_ C_R_HARMONIC_ COMBINATION LCAMHDL_PROFILE_ R_C_HARMONIC_ COMBINATION	**Application:** constant velocity - reversal reversal - constant velocity **Number of required TO cam segments:** 3 **Addition:** 1. Displacement of the inflection point (λ) 2. If λ = -1, parameterization of starting or ending acceleration necessary for calculation of λ **Advantage:** Extra low Cv and Cmdyn values for this kind of transitions **Disadvantage:** Ca-value is higher compared to modified acceleration trapezoid **Requirement:** *leadingValueStart < leadingValueEnd* *followingValueStart <> followingValueEnd*	*leadingValueStart* *followingValueStart* *leadingValueEnd* *followingValueEnd* constant velocity – reversal *geoVeloStart <> 0* *geoVeloEnd = 0* reversal - constant velocity *geoVeloStart = 0* *geoVeloEnd <> 0* **Addition:** *1. 0 < λ < 1* *λ: inflectionPointParameter* *2. λ = -1* *inflectionPointParameter = -1* constant velocity – reversal *geoAccelEnd <> 0* reversal – constant velocity *geoAccelStart <> 0*
LCAMHDL_PROFILE_ D_R_HARMONIC_ COMBINATION LCAMHDL_PROFILE_ R_D_HARMONIC_ COMBINATION	**Application:** dwell - reversal reversal - dwell **Number of required TO cam segments:** 3 **Addition:** 1. Displacement of the inflection point (λ) 2. If λ = -1, parameterization of starting or ending acceleration necessary for calculation of λ **Advantage:** Extra low Cv and Cmdyn values for this kind of transitions **Disadvantage:** Ca-value is higher compared to modified acceleration trapezoid **Requirement:** *leadingValueStart < leadingValueEnd* *followingValueStart <> followingValueEnd*	*leadingValueStart* *followingValueStart* *leadingValueEnd* *followingValueEnd* **Addition:** *1. 0 < λ < 1* *λ: inflectionPointParameter* *2. λ = -1* *inflectionPointParameter = -1* dwell - reversal *geoAccelEnd <> 0* reversal - dwell *geoAccelStart <> 0*

1.2 Hardware and software requirements

Requirements for this library

To be able to use the functionality of the library described in this document, the following hardware and software requirements must be met:

Hardware

Table 1-2

No.	Component	Article number	Alternative
1.	CPU 1515T-2 PN	6ES7 515-2TM01-0AB0	Other S7-1500T CPU with FW 2.9

Software

Table 1-3

No.	Component	Article number	Quantity
2.	STEP 7 Professional V17	6ES7822-1A.07-..	1

1.3 Library resources

What will you find in this section?

The following section gives you an overview of the size of the blocks of the LCamHdl library usable for advanced cam creation in the main and load memory.

Overall size

The overall size of the blocks of the LCamHdl library usable for advanced cam creation (TO_Cam or TO_Cam_10K) in the code work-memory is 80 Kbytes, in the data work-memory 5 Kbytes and 849 Kbytes in the load memory.

Size of the individual blocks

Table 1-4: Size of the blocks[1]

Block	Symbol	Size in code work-memory [Kbytes]	Size in data work-memory [Kbytes]	Size in load memory [Kbytes]
FB 31100	LCamHdl_CreateCamAdvanced	80		838
FB 31110	LCamHdl_CreateCam10kAdvanced	80		838
DB 31100	InstLCamHdl_CreateCamAdvanced		5	11
DB 31110	InstLCamHdl_CreateCam10kAdvanced		5	11

[1] Instance data blocks (prefix InstLCamHdl_) are not delivered with the library. They will be generated automatically with the call of a function block.

2 Blocks of the Library

What will you find in this section?

This chapter lists and explains all blocks of the LCamHdl library usable for advanced cam creation. Before that, however, you are informed of the blocks that are essentially involved in the implementation of the functionality.

2.1 List of the blocks

The following table lists all blocks of the LCamHdl library usable for advanced cam creation.

Table 2-1: List of blocks

Block	Symbol	Classification
FB 31100	LCamHdl_CreateCamAdvanced	In-house development
FB 31110	LCamHdl_CreateCam10kAdvanced	In-house development

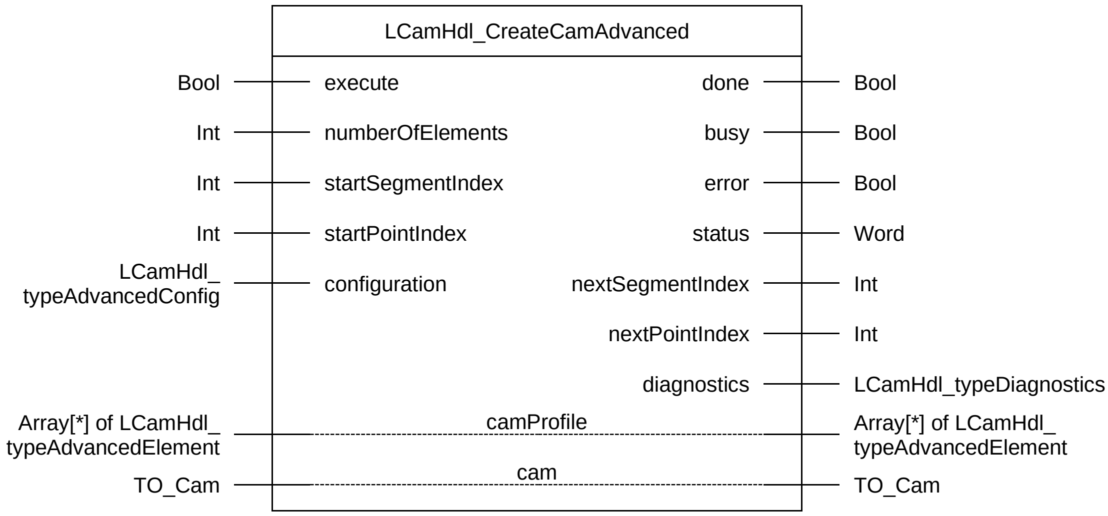
2.2 Explanation of the blocks

The following table explains all blocks of the LCamHdl library usable for advanced cam creation.

2.2.1 FB LCamHdl_CreateCamAdvanced (FB 31100)

Figure

Figure 2-1: Block diagram of *LCamHdl_CreateCamAdvanced*



Principle of operation

A cam disk can be created at runtime with a SIMATIC S7-1500T CPU. Segments with the profile types described in this document as well as points can be used to define a cam. Gaps between the segments and/or points are interpolated by the runtime system with the interpolation method chosen in the selected technology object cam.

A maximum number of 50 segments and 1000 points in a cam profile can be used to define a cam.

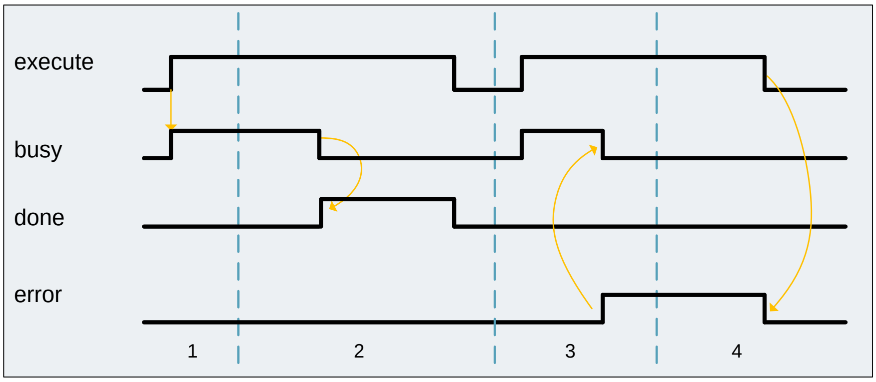
The FB **LCamHdl_CreateCamAdvanced** fills the necessary segments and points in the cam technology object starting at *startSegmentIndex* and *startPointIndex*. The function block can be configured to delete preceding or successive cam points and cam segments. In addition it is possible to interpolate the cam disk at the end with the *interpolateCam* configuration bit.

The default configuration has to be changed when using the **LCamHdl_CreateCamAdvanced** function block with additional function blocks.

For more information about the connection of the CamHdl function blocks see manual "LCamHdl Advanced cam creation – Additional profiles" \6\.

Function characteristics

Figure 2-2: Timing diagram of the *LCamHdl_CreateCamAdvanced* function block



1. The function block is activated by setting the *execute* input.
   The processing of the FB is indicated with *busy* = TRUE and *done* = FALSE.
2. If *busy* is FALSE and *done* is TRUE the function block finished processing. The cam disk is interpolated – a later addition of points / segments requires a new interpolation of the technology object cam.
3. If an error occurs during processing (e.g. error when executing the function block, internal errors of system functions, parameter supply error etc.), the *error* output is set and the corresponding error ID is written to the output *status*. Additional error information is provided at the output *diagnostics*.
4. All errors will be acknowledged and *error* output is reset if the *execute* input is reset. Signals stay active for at least one cycle. Reset of the diagnostic information is done with the next rising edge of *execute*.

Input parameters

Table 2-2: Input parameters of *LCamHdl_CreateCamAdvanced*

Parameter	Data type	Comment
execute	Bool	Rising edge starts action once (default: FALSE)
numberOfElements	Int	Number of used array elements of *camProfile* (default: -1 for whole array)
startSegmentIndex	Int	Start index of cam segment data (1-50) (default: 1)
startPointIndex	Int	Start index of cam point data (1-1000) (default: 1)
configuration	LCamHdl_typeAdvancedConfig	Configuration for interpolating the cam disk and deleting preceding or successive cam points/ segments

Output parameters

Table 2-3: Output parameters of *LCamHdl_CreateCamAdvanced*

Parameter	Data type	Comment
done	Bool	TRUE: Commanded action has been completed successfully (default: FALSE)
busy	Bool	TRUE: FB is not finished and new output values can be expected (default: FALSE)
error	Bool	TRUE: Rising edge informs that an error occurred during the execution of the FB (default: FALSE)
status	Word	16#0000 - 16#7FFF: Status of the FB, 16#8000 - 16#FFFF: Error identification (default: 16#7000, STATUS_NO_CALL)
nextSegmentIndex	Int	Next "empty" segment index (default: 0)
nextPointIndex	Int	Next "empty" point index (default: 0)
diagnostics	LCamHdl_typeDiagnostics	Diagnostics information of FB

InOut parameters

Table 2-4: InOut parameters of *LCamHdl_CreateCamAdvanced*

Parameter	Data type	Comment
camProfile	Array[*] of LCamHdl_type AdvancedElement	Definition of the cam disk to be created
cam	TO_Cam	Technology object cam disk

Status and error displays

Table 2-5: *LCamHdl_CreateCamAdvanced* status displays

Status	Meaning	Remedy / notes
16#0000	STATUS_EXECUTION_FINISHED	Execution finished without errors
16#7000	STATUS_NO_CALL	No call of FB
16#7001	STATUS_FIRST_CALL	First call of FB after enabling
16#7002	STATUS_SUBSEQUENT_CALL	Subsequent call of FB

Table 2-6: *LCamHdl_CreateCamAdvanced* error displays

Status	Meaning	Remedy / notes
16#8200	ERR_INVALID_PROFILE_TYPE	Invalid profile type in element no., see diagnostics.errorElementNo
16#8201	ERR_CAM_POINTS_OUT_OF_ BOUNDS	Maximum number of cam points (1000) of the technology object was exceeded
16#8202	ERR_CAM_SEGMENTS_OUT_OF _BOUNDS	Maximum number of cam segments (50) of the technology object was exceeded
16#8203	ERR_FOLLOWING_POS_IN_ PROFILE	Invalid following start or end position in element no., see diagnostics.errorElementNo
16#8204	ERR_MOD_VELO_TRAPEZOID_ PARAMETER	Invalid modified velocity trapezoid parameter C, permitted values: 0 < modVeloTrapezoidParameter <= 1 in element no., see diagnostics.errorElementNo
16#8205	ERR_MOD_SINE_MAX_ACCEL_ CA_STAR	Invalid value modSineMaxAccelCaStar in element no., see diagnostics.errorElementNo
16#8206	ERR_ACCELERATION_ZERO	Invalid acceleration geoAccelStart/End in element no., see diagnostics.errorElementNo
16#8207	ERR_INFLECTION_POINT	Invalid parameter for shifting the inflection point ($0< \lambda<1$) in element no., see diagnostics.errorElementNo
16#8208	ERR_CALCULATED_INFLECTION _POINT	Due to the parameterization, the calculated shifting of the inflection point (λ) is outside the range ($0< \lambda<1$) in element no., see diagnostics.errorElementNo
16#8209	ERR_LEADING_RANGE	Leading value is not valid (has to increase from start to end of profile) in element no., see diagnostics.errorElementNo
16#820A	ERR_FOLLOWING_RANGE_ZERO	The difference between start and end following value is zero in element no., see diagnostics.errorElementNo
16#820B	ERR_INVALID_NUMBER_OF_ ELEMENTS	Invalid number of cam elements
16#820C	ERR_HARM_COMB_LEADS_TO_ INVALID_CA_VALUE	Invalid parameterization of HARMONIC_COMBINATION (const. velocity – reversal / reversal – const. velocity), characteristic value Ca becomes invalid
16#820D	ERR_INVALID_PERFORMANCE_ MODE	Invalid performance mode parameter
16#820E	ERR_ACCELERATION_SIGN_ INVALID	Invalid acceleration sign (reversal point)

Status	Meaning	Remedy / notes
16#820F	ERR_CALCULATED_ PARAMETER_C	Due to the parameterization, the calculated value for c is outside the range (0< c <=1) in element no., see diagnostics.errorElementNo
16#8215	ERR_INVALID_PARAMETER_ COMBINATION	Parameter combination leads to invalid calculated coefficients in element no., see diagnostics.errorElementNo
16#8220	ERR_INVALID_DELETE_OPTION	Invalid delete option - check configuration
16#8222	ERR_INVALID_START_ SEGMENT_INDEX	Invalid startSegmentIndex, 1<=startSegmentIndex <=50
16#8223	ERR_INVALID_START_ POINT_INDEX	Invalid startPointIndex, 1<=startPointIndex <=1000
16#8400	ERR_CAM_IN_USE	Cam is in use and can't be interpolated
16#8600	ERR_INTERPOLATE_CAM	Error at interpolate cam – see return value of system function (diagnostics.subfunctionStatus)
16#8601	ERR_INVALID_STATE	Internal error, invalid state
16#8602	ERR_RESET_CAM	Error at reset cam – see return value of system function (diagnostics.subfunctionStatus)

2.2.2 FB LCamHdl_CreateCam10kAdvanced (FB 31110)

The function block LCamHdl_CreateCam10kAdvanced is a copy of the function block LCamHdl_CreateCamAdvanced. The "..10k.." version enables using a cam technology object of type TO_Cam_10k instead of TO_Cam, i.e. the maximum number of interpolation points increases from 1000 to 10000.

2.2.3 PLC tags

LCamHdl – profile constants

Table 2-7: User constants in *LCamHdl_ProfileConstants*

Name	Data type	Value	Comment
LCAMHDL_PROFILE_EMPTY	DInt	0	Empty
LCAMHDL_PROFILE_POINT	DInt	1	Single point
LCAMHDL_PROFILE_DWELL	DInt	2	Dwell
LCAMHDL_PROFILE_CONST_VELO	DInt	3	Constant velocity
LCAMHDL_PROFILE_POLY_3	DInt	4	3rd degree polynomial
LCAMHDL_PROFILE_POLY_5	DInt	5	5th degree polynomial
LCAMHDL_PROFILE_POLY_7	DInt	6	7th degree polynomial - implemented by using two 6th degree polynomials
LCAMHDL_PROFILE_D_D_PARABOLA	DInt	7	Dwell → dwell - quadratic parabola
LCAMHDL_PROFILE_C_C_PARABOLA	DInt	8	Constant velocity → constant velocity - quadratic parabola
LCAMHDL_PROFILE_D_D_BASIC_SINE	DInt	9	Dwell → dwell - basic sine line
LCAMHDL_PROFILE_D_D_INCLINED_SINE	DInt	10	Dwell → dwell - inclined sine line
LCAMHDL_PROFILE_D_D_MOD_ACCEL_TRAPEZOID	DInt	11	Dwell → dwell - modified acceleration trapezoid
LCAMHDL_PROFILE_D_D_MOD_SINE	DInt	12	Dwell → dwell - modified sine line
LCAMHDL_PROFILE_C_C_MOD_SINE	DInt	13	Constant velocity → constant velocity - modified sine line
LCAMHDL_PROFILE_R_R_MOD_VELO_TRAPEZOID	DInt	14	Reversal → reversal - sine-straight line - combination
LCAMHDL_PROFILE_R_R_BASIC_SINE	DInt	15	Reversal → reversal – basic sine line
LCAMHDL_PROFILE_D_C_MOD_SINE	DInt	16	Dwell → constant velocity - modified sine line
LCAMHDL_PROFILE_C_D_MOD_SINE	DInt	17	Constant velocity → dwell - modified sine line
LCAMHDL_PROFILE_D_R_MOD_ACCEL_TRAPEZOID	DInt	18	Dwell → reversal - modified acceleration trapezoid
LCAMHDL_PROFILE_R_D_MOD_ACCEL_TRAPEZOID	DInt	19	Reversal → dwell - modified acceleration trapezoid
LCAMHDL_PROFILE_D_R_HARMONIC_COMBINATION	DInt	20	Dwell → reversal - harmonic combination
LCAMHDL_PROFILE_R_D_HARMONIC_COMBINATION	DInt	21	Reversal → dwell - harmonic combination
LCAMHDL_PROFILE_C_R_HARMONIC_COMBINATION	DInt	22	Constant velocity → reversal - harmonic combination
LCAMHDL_PROFILE_R_C_HARMONIC_COMBINATION	DInt	23	Reversal → constant velocity - harmonic combination
LCAMHDL_PROFILE_POLY_4	DInt	30	4th degree polynomial
LCAMHDL_PROFILE_POLY_6	DInt	35	6th degree polynomial

LCamHdl – Config constants

Table 2-8: User constants in *LCamHdl_ConfigConstants*

Name	Data type	Value	Comment
LCAMHDL_DELETE_NO_DATA	DInt	0	Cam data delete option: No deletion
LCAMHDL_DELETE_DATA_UNTIL_ FIRST_INVALID_DATA_FOUND	DInt	1	Cam data delete option: Delete respective data until first invalid data is found
LCAMHDL_DELETE_ALL_DATA	DInt	2	Cam data delete option: Delete all respective data

2.2.4 PLC data types

LCamHdl_typeAdvancedConfig

Table 2-9: Parameter of *LCamHdl_typeAdvancedConfig*

Name	Data type	Value	Comment
deletePrecedingCam SegmentData	DInt	0	Delete preceding cam segment data 0: Delete no data 1: Delete until first invalid data found 2: Delete all data
deletePrecedingCam PointData	DInt	0	Delete preceding cam point data 0: Delete no data 1: Delete until first invalid data found 2: Delete all data
deleteSuccessiveCam SegmentData	DInt	1	Delete successive cam segment data 0: Delete no data 1: Delete until first invalid data found 2: Delete all data
deleteSuccessiveCam PointData	DInt	1	Delete successive cam point data 0: Delete no data 1: Delete until first invalid data found 2: Delete all data
interpolateCam	Bool	TRUE	TRUE: Interpolate cam FALSE: Do not interpolate cam

LCamHdl_typeAdvancedElement

Table 2-10: Parameter of *LCamHdl_typeAdvancedElement*

Name	Data type	Value	Comment
leadingValueStart	LReal	0.0	Leading value at the beginning of the element
leadingValueEnd	LReal	0.0	Leading value at the end of the element
followingValueStart	LReal	0.0	Following value at the beginning of the element
followingValueEnd	LReal	0.0	Following value at the end of the element
geoVeloStart	LReal	0.0	Velocity at the beginning of the element (real – not standardized)
geoVeloEnd	LReal	0.0	Velocity at the end of the element (real – not standardized)
geoAccelStart	LReal	0.0	Acceleration at the beginning of the element (real – not standardized)
geoAccelEnd	LReal	0.0	Acceleration at the end of the element (real – not standardized)
geoJerkStart	LReal	0.0	Jerk at the beginning of the element (real – not standardized)
geoJerkEnd	LReal	0.0	Jerk at the end of the element (real – not standardized)
inflectionPointParameter	LReal	0.5	Inflection point parameter (λ) – default: 0.5 standardized (0< λ <1)

Name	Data type	Value	Comment
modVeloTrapezoid Parameter	LReal	1.0	Element parameter (c) – sine-straight line- combination, see profile MOD_VELO_TRAPEZOID for details (chapter 1.1.4)
modSineMaxAccelCaSta r	LReal	0.0	Special case of modified sine – presetting by Ca* standardized, see profile type MOD_SINE for details (chapter 1.1.4)
camProfileType	DInt	0	Profile type of the cam disk element , 0: LCAMHDL_PROFILE_EMPTY (default)

LCamHdl_typeDiagnostics

Table 2-11: Parameter of *LCamHdl_typeDiagnostics*

Name	Data type	Value	Comment
status	Word	16#0000	Status of FB
subfunctionStatus	Word	16#0000	Status or return value of called FBs, FCs and system blocks
state	DInt	0	State of the state machine
errorElementNo	DInt	-1	Index of the camProfile with the first error (-1: no parameter with error)

3 Working with the Library

What will you find in this section?

This chapter consists of instructions for integrating the LCamHdl library into your STEP 7 project and instructions for using the library blocks.

3.1 Integrating the library into STEP 7

The table below lists the steps for integrating the LCamHdl library into your STEP 7 project. Subsequently, you can use the blocks of the LCamHdl library.

Note The following section assumes that a STEP 7 project exists.

Table 3-1: Integrating the library into STEP 7

No.	Action
1.	Extract the library LCamHdl_V1_x_x.zip to a local folder.
2.	In TIA Portal select “Options” -> “Global libraries” -> “Open library…”.
3.	Browse to the file LCamHdl.al17. It can be found in the subfolder LCamHdl of the extracted zip file.
4.	Open the global library in read-only mode.
5.	The LCamHdl library is now available in the task card “Global libraries”.

3.2 Integrating the library blocks into STEP 7

The table below lists the steps for integrating the blocks of the LCamHdl library into your STEP 7 program.

Figure 3-1: Integrating the library blocks into STEP 7

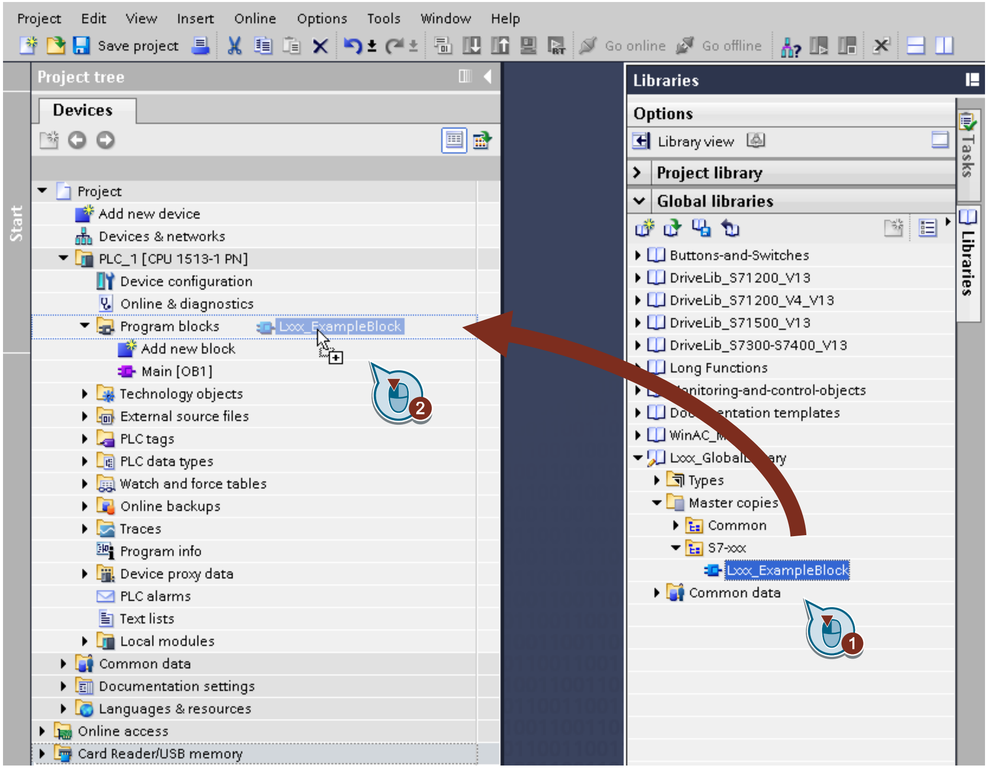


Table 3-2: Integrating the library blocks into STEP 7

No.	Action	Note
1.	Optional: Copy the folder *LCamHdl_Tags* with Drag & Drop into the “PLC tags” in the PLC.	Copy PLC tags
2.	Copy the folder *LCamHdl_Types* with Drag & Drop into the “PLC data types” in the PLC.	Copy PLC data types
3.	Copy the *LCamHdl_CreateCamXXX* FB with Drag & Drop into the “Program blocks” in the PLC.	Copy program blocks
4.	Now the blocks can be configured and called in the user program.	

4 Notes and Support

What will you find in this section?

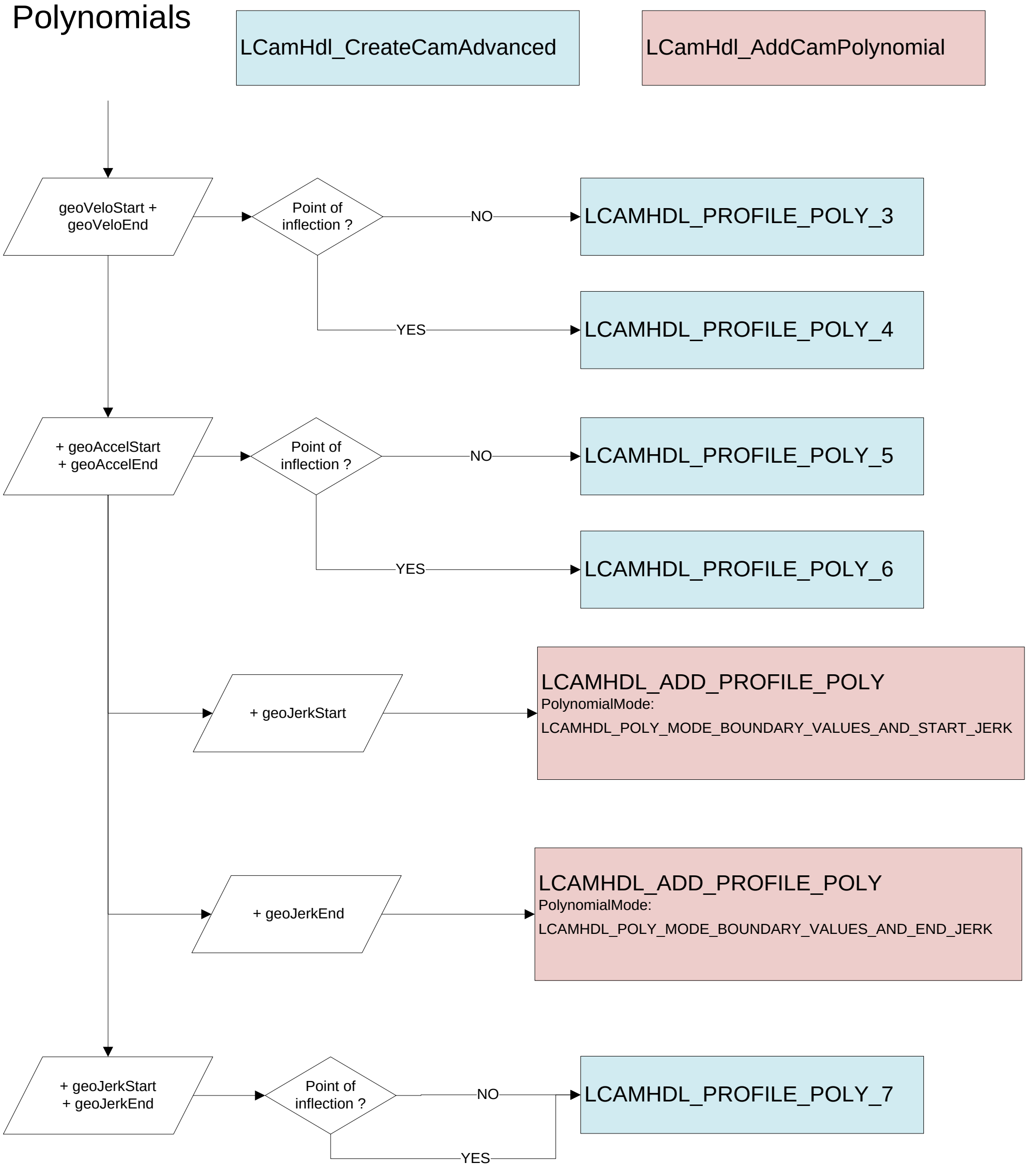
This chapter provides further support in handling the described LCamHdl library.

Note

Parameter comments in the programming editor are only available in language 'English (United States)'

4.1 Decision support for polynomials

Figure 4-1: Decision support for polynomials



4.2 Application examples

The examples require a technology object cam named “Cam_1” in the runtime system of the SIMATIC S7-1500T.

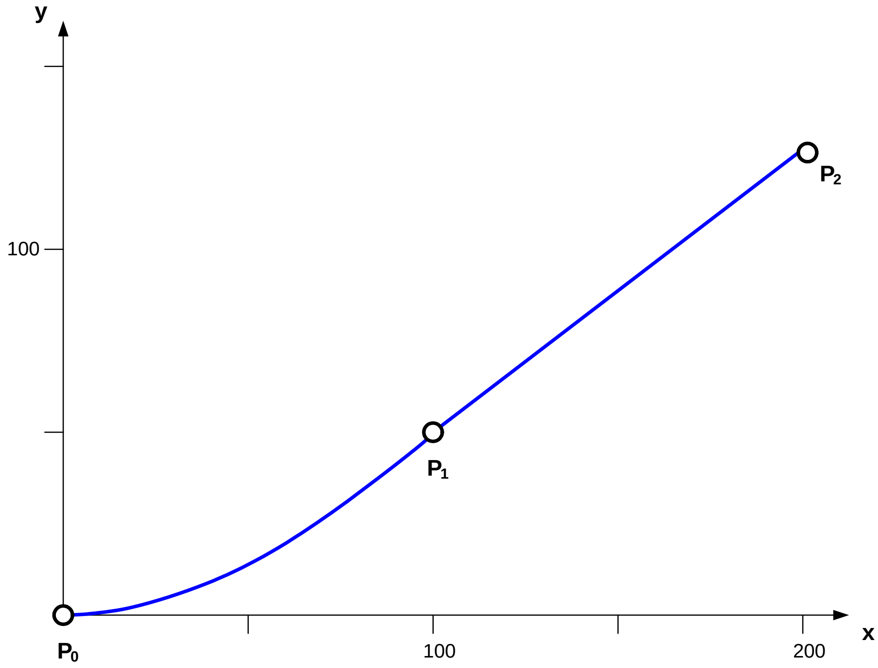
4.2.1 Example 1

Element 1: dwell – constant velocity → 3rd degree polynomial

Element 2: constant velocity → straight line

P_0:	$X_0 = 0$	$Y_0 = 0$	$m_0 = 0$
P_1:	$X_1 = 100$	$Y_1 = 50$	$m_1 = 0.75$
P_2:	$X_2 = 200$	$Y_2 = ?$	$m_2 = m_1$

Figure 4-2: Example 1



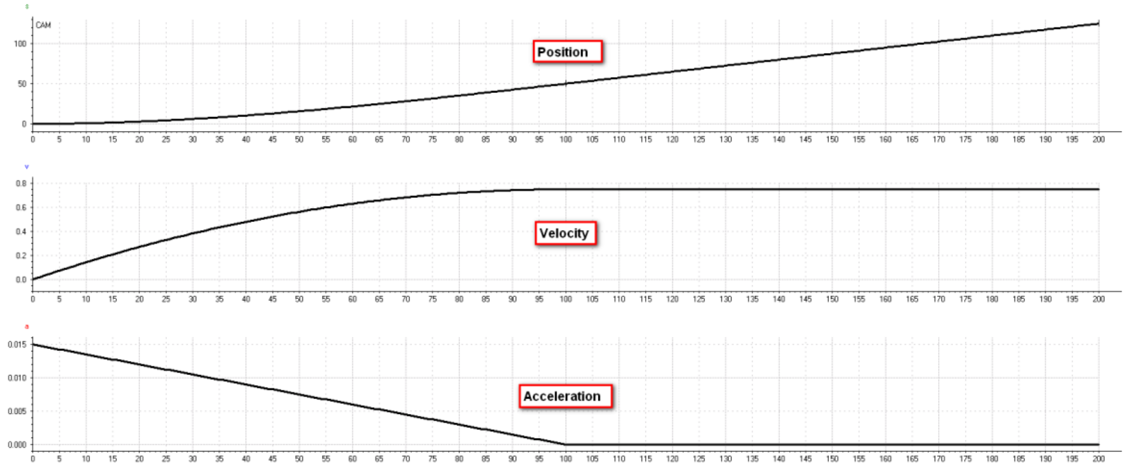
$$Y_2 = Y_1 + (X_2 - X_1) \cdot m_1 = 50 + 100 \cdot 0.75 = 125$$

Source code

```
FUNCTION_BLOCK "LCamHdl_Example_1"
{ S7_Optimized_Access := 'TRUE' }
VERSION : 1.0
VAR
   instLCamHdl_CreateCam : "LCamHdl_CreateCamAdvanced";
   statCamProfile : Array[0..1] of "LCamHdl_typeAdvancedElement";
   statNumberOfElements : Int;
END_VAR

BEGIN

// 1st element
#statCamProfile[0].inflectionPointParameter := 0.5;
#statCamProfile[0].camProfileType := "LCAMHDL_PROFILE_POLY_3";
#statCamProfile[0].leadingValueStart := 0.0;
#statCamProfile[0].leadingValueEnd := 100.0;
#statCamProfile[0].followingValueStart :=0.0;
#statCamProfile[0].followingValueEnd := 50.0;
#statCamProfile[0].geoVeloStart := 0.0;
#statCamProfile[0].geoVeloEnd := 0.75;
#statCamProfile[0].geoAccelStart := 0.0;
#statCamProfile[0].geoAccelEnd := 0.0;
#statCamProfile[0].geoJerkStart := 0.0;
#statCamProfile[0].geoJerkEnd := 0.0;

// 2nd element
#statCamProfile[1].inflectionPointParameter := 0.5;
#statCamProfile[1].camProfileType := "LCAMHDL_PROFILE_CONST_VELO";
#statCamProfile[1].leadingValueStart := 100.0;
#statCamProfile[1].leadingValueEnd := 200.0;
#statCamProfile[1].followingValueStart := 50.0;
#statCamProfile[1].followingValueEnd := 125.0;
#statCamProfile[1].geoVeloStart := 0.75;
#statCamProfile[1].geoVeloEnd := 0.75;
#statCamProfile[1].geoAccelStart := 0.0;
#statCamProfile[1].geoAccelEnd := 0.0;
#statCamProfile[1].geoJerkStart := 0.0;
#statCamProfile[1].geoJerkEnd := 0.0;

#statNumberOfElements := 2;

// Function block call
# instLCamHdl_CreateCam (execute := TRUE,
              cam := "Cam_1",
              camProfile := #statCamProfile,
              numberOfElements := #statNumberOfElements);
END_FUNCTION_BLOCK
```

Cam disk

Figure 4-3: Resulting cam disk and derivations



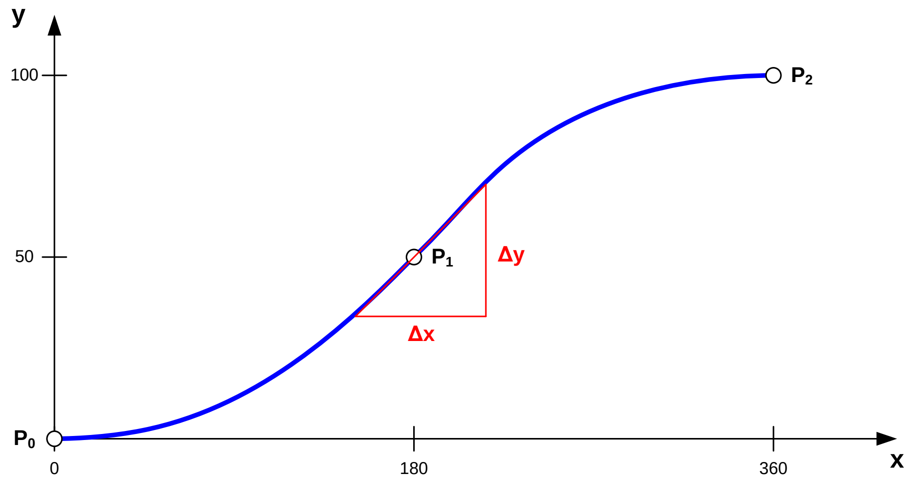
4.2.2 Example 2

Element 1: dwell – constant velocity → modified sine line

Element 2: constant velocity – dwell → 5th degree polynomial

P_0:	$X_0 = 0$	$Y_0 = 0$	$m_0 = 0$
P_1:	$X_1 = 180$	$Y_1 = 50$	$m_1 = ?$
P_2 :	$X_2 = 360$	$Y_2 = 100$	$m_2 = 0$

Figure 4-4: Example 2



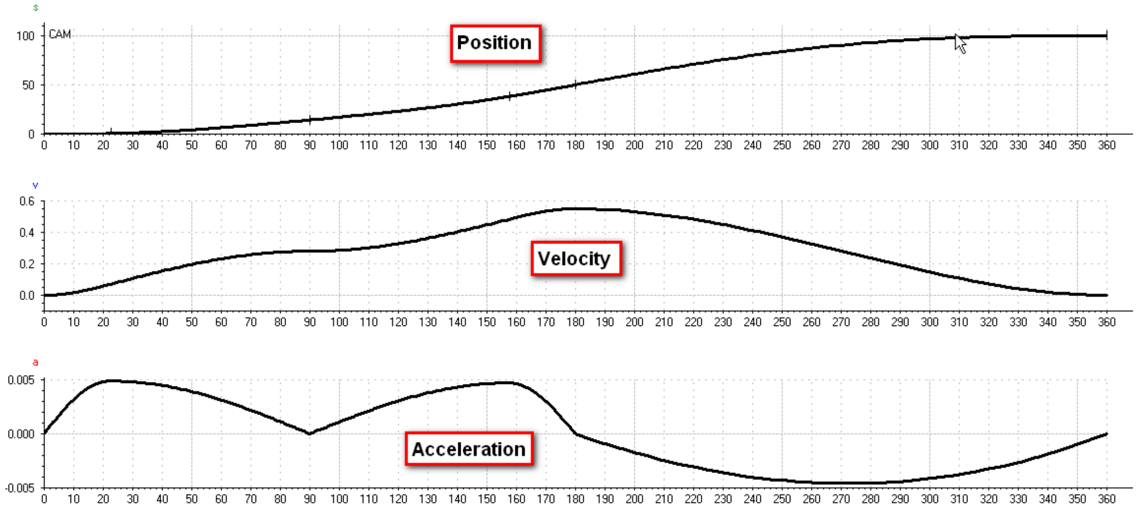
$$m_1 = \frac{\Delta y}{\Delta x} = 0{,}55$$

Source code:

```
FUNCTION_BLOCK "LCamHdl_Example_2"
{ S7_Optimized_Access := 'TRUE' }
VERSION : 1.0
VAR
   instLCamHdl_CreateCam: "LCamHdl_CreateCamAdvanced";
   statCamProfile : Array[0..1] of "LCamHdl_typeAdvancedElement";
   statNumberOfElements : Int;
END_VAR

BEGIN

// 1st element
#statCamProfile[0].inflectionPointParameter := 0.5;
#statCamProfile[0].camProfileType := "LCAMHDL_PROFILE_D_C_MOD_SINE";
#statCamProfile[0].leadingValueStart := 0.0;
#statCamProfile[0].leadingValueEnd := 180;
#statCamProfile[0].followingValueStart := 0.0;
#statCamProfile[0].followingValueEnd := 50.0;
#statCamProfile[0].geoVeloStart := 0.0;
#statCamProfile[0].geoVeloEnd := 0.55;
#statCamProfile[0].geoAccelStart := 0.0;
#statCamProfile[0].geoAccelEnd := 0.0;
#statCamProfile[0].geoJerkStart := 0.0;
#statCamProfile[0].geoJerkEnd := 0.0;

// 2nd element
#statCamProfile[1].inflectionPointParameter := 0.5;
#statCamProfile[1].camProfileType := "LCAMHDL_PROFILE_POLY_5";
#statCamProfile[1].leadingValueStart := 180.0;
#statCamProfile[1].leadingValueEnd := 360.0;
#statCamProfile[1].followingValueStart := 50.0;
#statCamProfile[1].followingValueEnd := 100.0;
#statCamProfile[1].geoVeloStart := 0.55;
#statCamProfile[1].geoVeloEnd := 0.0;
#statCamProfile[1].geoAccelStart := 0.0;
#statCamProfile[1].geoAccelEnd := 0.0;
#statCamProfile[1].geoJerkStart := 0.0;
#statCamProfile[1].geoJerkEnd := 0.0;

#statNumberOfElements := 2;

// Function block call
# instLCamHdl_CreateCam (execute := TRUE,
                   cam := "Cam_1",
                   camProfile := #statCamProfile,
                   numberOfElements := #statNumberOfElements);
END_FUNCTION_BLOCK
```

Cam disk

Figure 4-5: Resulting cam disk and derivations

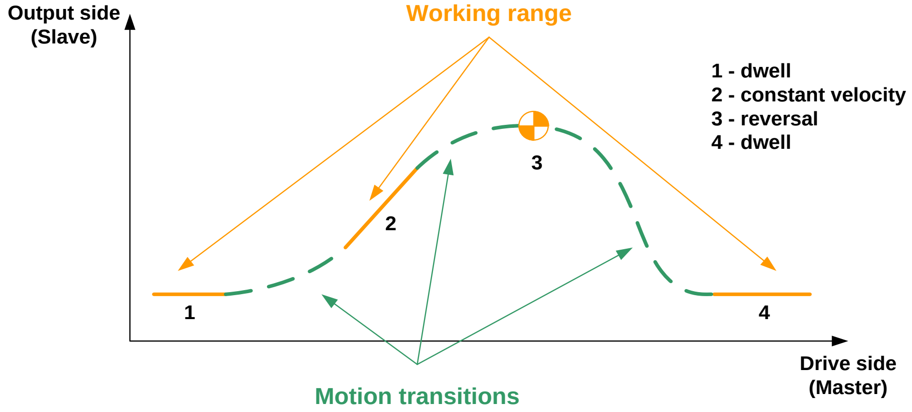


4.3 Laws of motion

4.3.1 Motion transitions according to VDI 2143

A cam disk consists of segments that can either be a working range or a motion transition. The working ranges are defined by the technological application (e.g. flying saw) within the machine.

Figure 4-6: Working ranges and motion transitions in a cam disk



A motion transition between two consecutive working ranges has to fulfill boundary conditions (Figure 4-7), e.g. velocity and acceleration has to be continuous. This guarantees that the drive runs smoothly, e.g. without jerks. The elements are described by mathematic functions between start and end point, e.g. polynomials.

Figure 4-7: Boundary conditions of a motion transition

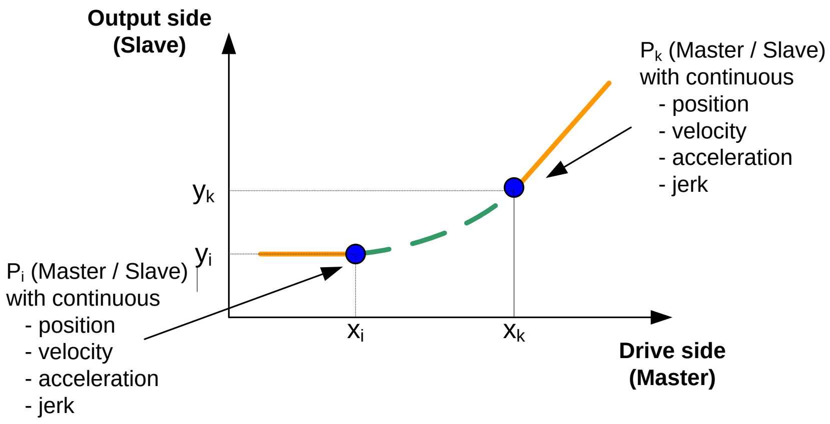


Table 4-1: Selection of a motion transition based on consecutive segments

Boundary condition start \ Boundary condition end	**D** (dwell)	**C** (constant velocity)	**R** (reversal)	**M** (motion)
D (dwell)	$v=0$ $a=0$ / $v=0$ $a=0$	$v=0$ $a=0$ / $v\neq0$ $a=0$	$v=0$ $a=0$ / $v=0$ $a\neq0$	$v=0$ $a=0$ / $v\neq0$ $a\neq0$
C (constant velocity)	$v\neq0$ $a=0$ / $v=0$ $a=0$	$v\neq0$ $a=0$ / $v\neq0$ $a=0$	$v\neq0$ $a=0$ / $v=0$ $a\neq0$	$v\neq0$ $a=0$ / $v\neq0$ $a\neq0$
R (reversal)	$v=0$ $a\neq0$ / $v=0$ $a=0$	$v=0$ $a\neq0$ / $v\neq0$ $a=0$	$v=0$ $a\neq0$ / $v=0$ $a\neq0$	$v=0$ $a\neq0$ / $v\neq0$ $a\neq0$
M (motion)	$v\neq0$ $a\neq0$ / $v=0$ $a=0$	$v\neq0$ $a\neq0$ / $v\neq0$ $a=0$	$v\neq0$ $a\neq0$ / $v=0$ $a\neq0$	$v\neq0$ $a\neq0$ / $v\neq0$ $a\neq0$

4.3.2 Selection of profiles according to motion transitions

Table 4-2 shows appropriate laws of motion for the 16 possible transitions of motion according to VDI 2143, without any judgment of quality.

Table 4-2: Appropriate laws of motion for the possible motion transitions

	Dwell D	**Constant velocity C**	**Reversal R**	**Motion M**
Dwell D	- Straight line - Quadratic parabola - Basic sine line - 5th degree polynomial - Inclined sine line - Mod. accel.trapezoid - Modified sine line	- 5th degree polynomial - Modified sine line	- Mod. accel. trapezoid - 5th degree polynomial - Harmonic combination	- 5th degree polynomial
Constant velocity C	- 5th degree polynomial - Modified sine line	- Straight line - 5th degree polynomial - Modified sine line - Quadratic parabola	- Harmonic combination - 5th degree polynomial	- 5th degree polynomial
Reversal R	- Mod. accel.trapezoid - 5th degree polynomial - Harmonic combination	- Harmonic combination - 5th degree polynomial	- Sine - straight line combination (velocity trapezoid) - Basic sine line - 5th degree polynomial	- 5th degree polynomial
Motion M	- 5th degree polynomial	- 5th degree polynomial	- 5th degree polynomial	- 5th degree polynomial

You can find a detailed explanation of the above described profiles and transitions in the guideline „VDI 2143 Motion rules for cam mechanisms“ /3/.

5 Appendix

5.1 Service and support

Industry Online Support

Do you have any questions or need assistance?

Siemens Industry Online Support offers round the clock access to our entire service and support know-how and portfolio.

The Industry Online Support is the central address for information about our products, solutions and services.

Product information, manuals, downloads, FAQs, application examples and videos – all information is accessible with just a few mouse clicks:
support.industry.siemens.com

Technical Support

The Technical Support of Siemens Industry provides you fast and competent support regarding all technical queries with numerous tailor-made offers – ranging from basic support to individual support contracts. Please send queries to Technical Support via Web form:
www.siemens.com/industry/supportrequest

SITRAIN – Training for Industry

We support you with our globally available training courses for industry with practical experience, innovative learning methods and a concept that's tailored to the customer's specific needs.

For more information on our offered trainings and courses, as well as their locations and dates, refer to our web page:
www.siemens.com/sitrain

Service offer

Our range of services includes the following:

- Plant data services
- Spare parts services
- Repair services
- On-site and maintenance services
- Retrofitting and modernization services
- Service programs and contracts

You can find detailed information on our range of services in the service catalog web page:
support.industry.siemens.com/cs/sc

Industry Online Support app

You will receive optimum support wherever you are with the "Siemens Industry Online Support" app. The app is available for iOS and Android:
support.industry.siemens.com/cs/ww/en/sc/2067

5.2 Application support

Siemens AG
Digital Industries
Factory Automation
Production Machines
DI FA PMA APC
Frauenauracher Str. 80
91056 Erlangen, Germany

mailto: tech.team.motioncontrol@siemens.com

5.3 Links and literature

Table 5-1

No.	Topic
\1\	Siemens Industry Online Support https://support.industry.siemens.com
\2\	Link to this entry page of this application example https://support.industry.siemens.com/cs/ww/en/view/105644659
\3\	VDI 2143 Page. 1: Motion rules for cam mechanisms; theoretical fundamentals

5.4 Change documentation

Table 5-2

Version	Date	Modifications
V1.0.0	10/2016	First version
V1.0.1	04/2017	Added error ERR_CALCULATED_PARAMETER_C, Lambda calculation for profile types harmonic combination
V1.1	11/2017	Additional polynomial profiles. Change of block interface: order of busy and done reversed; added configuration type and interface to connect additional LCamHdl_AddCam blocks
V1.2	03/2020	Updated, e.g. chapter 1 - scenario 3
V1.3	05/2021	Scope of application is now STEP 7 Professional V17 (Motion Control V6.0, firmware V2.9) New block for handling the new cam technology object of type TO_Cam_10k