Creation of cam disks at runtime for S7-1500T

Library LCamHdl – Advanced cam creation – Additional profiles

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 blocks LCamHdl_AddCamSine, LCamHdl_AddCamInvSine, LCamHdl_AddCamPolynomial and LCamHdl_AddCamDblHarmonic of the LCamHdl block library. The before mentioned blocks expand the function block LCamHdl_CreateCamAdvanced by several additional profiles. 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.

NOTE Support for the function block LCamHdl_CreateCamAdvanced can be found in the manual “LCamHdl Advanced” \2\.

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.

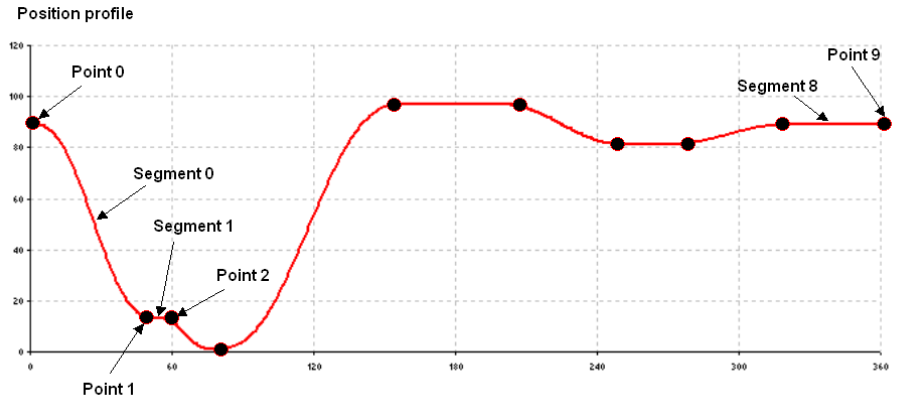
Within the scope of the SIMATIC S7-1500T, there are two ways to configure cam disks:

- 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.

NOTE For further details please refer to the manuals “LCamHdl Basic” \2\ and “LCamHdl Advanced” \2\.

1.2 Profile types

The function blocks **LCamHdl_AddCamSine**, **LCamHdl_AddCamInvSine**, **LCamHdl_AddCamPolynomial** and **LCamHdl_AddCamDblHarmonic** of the LCamHdl block library expand the function block **LCamHdl_CreateCamAdvanced** by the following profiles:

- sine profile
- inverse sine profile (arc sine)
- polynomial profile with and without trigonometric portion
- double harmonic sine profile (cosine^4)

Sine profile

A sine element describes a motion according to the sine function. The sine function can be adjusted with the phase angle in the start point and end point, the period length, the amplitude as well as the oscillation zero point (offset). It is also possible to realize an inclined sine element. The incline sine function can be adjusted with the displacement at the start point and end point or the inclination.

Inverse sine (approximated)

An inverse sine describes a motion according to the arcsine function. The arcsine function is the inverse function of the sine function and is approximated using interpolation points of the arcsine function.

Polynomial with or without trigonometric portion

A polynomial with trigonometric portion describes a motion according to the polynomial function with trigonometric portion.

Double harmonic element (approximated)

A double harmonic element describes a motion according to the cosine^4 function. A double harmonic element is approximated using interpolation points of the cosine^4 function.

Overview of the profile types and modes

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 / modes

Profile type	Profile mode	Additional mode	Supported parameters
EMPTY (default): LCAMHDL_PROFILE_EMPTY	-	-	Empty element – not used in current cam
Sine: LCAMHDL_ADD_PROFILE_SINE **LCamHdl_AddCamSine**	Definition by phase in start point and phase in end point LCAMHDL_SINE_MODE_PHASE_START_AND_END	-	*leadingValueStart* *leadingValueEnd* *amplitude* *phaseStart* *phaseEnd* *displacementStart*
	Definition by phase in start point and period length LCAMHDL_SINE_MODE_PHASE_START_AND_PERIOD_LENGTH	-	*leadingValueStart* *leadingValueEnd* *amplitude* *phaseStart* *periodLength* *displacementStart*
	Definition by phase in start point and frequency LCAMHDL_SINE_MODE_PHASE_START_AND_FREQUENCY	-	*leadingValueStart* *leadingValueEnd* *amplitude* *phaseStart* *frequency* *displacementStart*
	Definition by period length and phase in end point LCAMHDL_SINE_MODE_PERIOD_LENGTH_AND_PHASE_END	-	*leadingValueStart* *leadingValueEnd* *amplitude* *phaseEnd* *periodLength* *displacementStart*
	Definition by frequency and phase in end point LCAMHDL_SINE_MODE_FREQUENCY_AND_PHASE_END	-	*leadingValueStart* *leadingValueEnd* *amplitude* *phaseEnd* *frequency* *displacementStart*
	Requirement: *leadingValueStart < leadingValueEnd* *phaseStart <> phaseEnd* *periodLength <> 0.0* *frequency <> 0.0*		

Profile type	Profile mode	Additional mode	Supported parameters
Inclined sine: LCAMHDL_ADD_PROFILE_INCLINED_SINE **LCamHdl_AddCamSine**	Definition by phase in start point and phase in end point LCAMHDL_SINE_MODE_PHASE_START_AND_END	Inclination defined by displacement at start and end point LCAMHDL_INCL_SINE_MODE_DISPLACEMENT_START_AND_END	*leadingValueStart* *leadingValueEnd* *amplitude* *phaseStart* *phaseEnd* *displacementStart* *displacementEnd*
		Inclination defined by displacement at start point and inclination LCAMHDL_INCL_SINE_MODE_DISPLACEMENT_START_AND_INCLINATION	*leadingValueStart* *leadingValueEnd* *amplitude* *phaseStart* *phaseEnd* *displacementStart* *inclination*
		Inclination defined by inclination and displacement at end point LCAMHDL_INCL_SINE_MODE_INCLINATION_AND_DISPLACEMENT_END	*leadingValueStart* *leadingValueEnd* *amplitude* *phaseStart* *phaseEnd* *inclination* *displacementEnd*
	Definition by phase in start point and period length LCAMHDL_SINE_MODE_PHASE_START_AND_PERIOD_LENGTH	Inclination defined by displacement at start and end point LCAMHDL_INCL_SINE_MODE_DISPLACEMENT_START_AND_END	*leadingValueStart* *leadingValueEnd* *amplitude* *phaseStart* *periodLength* *displacementStart* *displacementEnd*
		Inclination defined by displacement at start point and inclination LCAMHDL_INCL_SINE_MODE_DISPLACEMENT_START_AND_INCLINATION	*leadingValueStart* *leadingValueEnd* *amplitude* *phaseStart* *periodLength* *displacementStart* *inclination*
		Inclination defined by inclination and displacement at end point LCAMHDL_INCL_SINE_MODE_INCLINATION_AND_DISPLACEMENT_END	*leadingValueStart* *leadingValueEnd* *amplitude* *phaseStart* *periodLength* *inclination* *displacementEnd*
	Definition by phase in start point and frequency LCAMHDL_SINE_MODE_PHASE_START_AND_FREQUENCY	Inclination defined by displacement at start and end point LCAMHDL_INCL_SINE_MODE_DISPLACEMENT_START_AND_END	*leadingValueStart* *leadingValueEnd* *amplitude* *phaseStart* *frequency* *displacementStart* *displacementEnd*

Profile type	Profile mode	Additional mode	Supported parameters
		Inclination defined by displacement at start point and inclination LCAMHDL_INCL_SINE_MODE_DISPLACEMENT_START_AND_INCLINATION	*leadingValueStart* *leadingValueEnd* *amplitude* *phaseStart* *frequency* *displacementStart* *inclination*
		Inclination defined by inclination and displacement at end point LCAMHDL_INCL_SINE_MODE_INCLINATION_AND_DISPLACEMENT_END	*leadingValueStart* *leadingValueEnd* *amplitude* *phaseStart* *frequency* *inclination* *displacementEnd*
	Definition by period length and phase in end point LCAMHDL_SINE_MODE_PERIOD_LENGTH_AND_PHASE_END	Inclination defined by displacement at start and end point LCAMHDL_INCL_SINE_MODE_DISPLACEMENT_START_AND_END	*leadingValueStart* *leadingValueEnd* *amplitude* *phaseEnd* *periodLength* *displacementStart* *displacementEnd*
		Inclination defined by displacement at start point and inclination LCAMHDL_INCL_SINE_MODE_DISPLACEMENT_START_AND_INCLINATION	*leadingValueStart* *leadingValueEnd* *amplitude* *phaseEnd* *periodLength* *displacementStart* *inclination*
		Inclination defined by inclination and displacement at end point LCAMHDL_INCL_SINE_MODE_INCLINATION_AND_DISPLACEMENT_END	*leadingValueStart* *leadingValueEnd* *amplitude* *phaseEnd* *periodLength* *inclination* *displacementEnd*
	Definition by frequency and phase in end point LCAMHDL_SINE_MODE_FREQUENCY_AND_PHASE_END	Inclination defined by displacement at start and end point LCAMHDL_INCL_SINE_MODE_DISPLACEMENT_START_AND_END	*leadingValueStart* *leadingValueEnd* *amplitude* *phaseEnd* *frequency* *displacementStart* *displacementEnd*
		Inclination defined by displacement at start point and inclination LCAMHDL_INCL_SINE_MODE_DISPLACEMENT_START_AND_INCLINATION	*leadingValueStart* *leadingValueEnd* *amplitude* *phaseEnd* *frequency* *displacementStart* *inclination*

Profile type	Profile mode	Additional mode	Supported parameters
		Inclination defined by inclination and displacement at end point LCAMHDL_INCL_SINE_MODE_INCLINATION_AND_DISPLACEMENT_END	*leadingValueStart* *leadingValueEnd* *amplitude* *phaseEnd* *frequency* *inclination* *displacementEnd*
	Requirement: *leadingValueStart* < *leadingValueEnd* *phaseStart* <> *phaseEnd* *periodLength* <> *0.0* *frequency* <> *0.0*		
Inverse sine: LCAMHDL_ADD_PROFILE_INVERSE_SINE **LCamHdl_AddCamInvSine**	**Requirement:** leadingValueStart < leadingValueEnd 2 <= numberOfInterpolationPoints <= 1000 -1 <= definitionRangeStart <= 1 -1 <= definitionRangeEnd <= 1 definitionRangeStart <> definitionRangeEnd		*leadingValueStart* *leadingValueEnd* *followingValueMaximum* *followingValueMinimum* *definitionRangeStart* *definitionRangeEnd* *mirror* *numberOfInterpolationPoints*
Polynomial: LCAMHDL_ADD_PROFILE_POLY **LCamHdl_AddCamPolynomial**	Definition by boundary values and point of inflection LCAMHDL_POLY_MODE_BOUNDARY_VALUES_AND_POINT_OF_INFLECTION	-	*leadingValueStart* *leadingValueEnd* *followingValueStart* *followingValueEnd* *geoVeloStart* *geoVeloEnd* *geoAccelStart* *geoAccelEnd* *inflectionPointParameter*
	Definition by boundary values and start jerk LCAMHDL_POLY_MODE_BOUNDARY_VALUES_AND_START_JERK	-	*leadingValueStart* *leadingValueEnd* *followingValueStart* *followingValueEnd* *geoVeloStart* *geoVeloEnd* *geoAccelStart* *geoAccelEnd* *geoJerkStart*
	Definition by boundary values and end jerk LCAMHDL_POLY_MODE_BOUNDARY_VALUES_AND_END_JERK	-	*leadingValueStart* *leadingValueEnd* *followingValueStart* *followingValueEnd* *geoVeloStart* *geoVeloEnd* *geoAccelStart* *geoAccelEnd* *geoJerkEnd*
	Requirement: *leadingValueStart* < *leadingValueEnd*		

Profile type	Profile mode	Additional mode	Supported parameters
Polynomial with trigonometric portion: LCAMHDL_ADD_PROFILE_POLY_TRIGONOMETRIC **LCamHdl_AddCamPolynomial**	Definition by boundary values and point of inflection LCAMHDL_POLY_MODE_BOUNDARY_VALUES_AND_POINT_OF_INFLECTION	Definition by phase in start point and phase in end point LCAMHDL_SINE_MODE_PHASE_START_AND_END	*leadingValueStart* *leadingValueEnd* *followingValueStart* *followingValueEnd* *geoVeloStart* *geoVeloEnd* *geoAccelStart* *geoAccelEnd* *inflectionPointParameter* *amplitude* *phaseStart* *phaseEnd*
		Definition by phase in start point and period length LCAMHDL_SINE_MODE_PHASE_START_AND_PERIOD_LENGTH	*leadingValueStart* *leadingValueEnd* *followingValueStart* *followingValueEnd* *geoVeloStart* *geoVeloEnd* *geoAccelStart* *geoAccelEnd* *inflectionPointParameter* *amplitude* *phaseStart* *periodLength*
		Definition by phase in start point and frequency LCAMHDL_SINE_MODE_PHASE_START_AND_FREQUENCY	*leadingValueStart* *leadingValueEnd* *followingValueStart* *followingValueEnd* *geoVeloStart* *geoVeloEnd* *geoAccelStart* *geoAccelEnd* *inflectionPointParameter* *amplitude* *phaseStart* *frequency*
		Definition by period length and phase in end point LCAMHDL_SINE_MODE_PERIOD_LENGTH_AND_PHASE_END	*leadingValueStart* *leadingValueEnd* *followingValueStart* *followingValueEnd* *geoVeloStart* *geoVeloEnd* *geoAccelStart* *geoAccelEnd* *inflectionPointParameter* *amplitude* *phaseEnd* *periodLength*

Profile type	Profile mode	Additional mode	Supported parameters
		Definition by frequency and phase in end point LCAMHDL_SINE_MODE_FREQUENCY_AND_PHASE_END	*leadingValueStart* *leadingValueEnd* *followingValueStart* *followingValueEnd* *geoVeloStart* *geoVeloEnd* *geoAccelStart* *geoAccelEnd* *inflectionPointParameter* *amplitude* *phaseEnd* *frequency*
	Definition by boundary values and start jerk LCAMHDL_POLY_MODE_BOUNDARY_VALUES_AND_START_JERK	Definition by phase in start point and phase in end point LCAMHDL_SINE_MODE_PHASE_START_AND_END	*leadingValueStart* *leadingValueEnd* *followingValueStart* *followingValueEnd* *geoVeloStart* *geoVeloEnd* *geoAccelStart* *geoAccelEnd* *geoJerkStart* *amplitude* *phaseStart* *phaseEnd*
		Definition by phase in start point and period length LCAMHDL_SINE_MODE_PHASE_START_AND_PERIOD_LENGTH	*leadingValueStart* *leadingValueEnd* *followingValueStart* *followingValueEnd* *geoVeloStart* *geoVeloEnd* *geoAccelStart* *geoAccelEnd* *geoJerkStart* *amplitude* *phaseStart* *periodLength*
		Definition by phase in start point and frequency LCAMHDL_SINE_MODE_PHASE_START_AND_FREQUENCY	*leadingValueStart* *leadingValueEnd* *followingValueStart* *followingValueEnd* *geoVeloStart* *geoVeloEnd* *geoAccelStart* *geoAccelEnd* *geoJerkStart* *amplitude* *phaseStart* *frequency*

Profile type	Profile mode	Additional mode	Supported parameters
		Definition by period length and phase in end point LCAMHDL_SINE_MODE_PERIOD_LENGTH_AND_PHASE_END	*leadingValueStart* *leadingValueEnd* *followingValueStart* *followingValueEnd* *geoVeloStart* *geoVeloEnd* *geoAccelStart* *geoAccelEnd* *geoJerkStart* *amplitude* *phaseEnd* *periodLength*
		Definition by frequency and phase in end point LCAMHDL_SINE_MODE_FREQUENCY_AND_PHASE_END	*leadingValueStart* *leadingValueEnd* *followingValueStart* *followingValueEnd* *geoVeloStart* *geoVeloEnd* *geoAccelStart* *geoAccelEnd* *geoJerkStart* *amplitude* *phaseEnd* *frequency*
	Definition by boundary values and end jerk LCAMHDL_POLY_MODE_BOUNDARY_VALUES_AND_END_JERK	Definition by phase in start point and phase in end point LCAMHDL_SINE_MODE_PHASE_START_AND_END	*leadingValueStart* *leadingValueEnd* *followingValueStart* *followingValueEnd* *geoVeloStart* *geoVeloEnd* *geoAccelStart* *geoAccelEnd* *geoJerkEnd* *amplitude* *phaseStart* *phaseEnd*
		Definition by phase in start point and period length LCAMHDL_SINE_MODE_PHASE_START_AND_PERIOD_LENGTH	*leadingValueStart* *leadingValueEnd* *followingValueStart* *followingValueEnd* *geoVeloStart* *geoVeloEnd* *geoAccelStart* *geoAccelEnd* *geoJerkEnd* *amplitude* *phaseStart* *periodLength*

Profile type	Profile mode	Additional mode	Supported parameters
		Definition by phase in start point and frequency LCAMHDL_SINE_MODE_PHASE_START_AND_FREQUENCY	*leadingValueStart* *leadingValueEnd* *followingValueStart* *followingValueEnd* *geoVeloStart* *geoVeloEnd* *geoAccelStart* *geoAccelEnd* *geoJerkEnd* *amplitude* *phaseStart* *frequency*
		Definition by period length and phase in end point LCAMHDL_SINE_MODE_PERIOD_LENGTH_AND_PHASE_END	*leadingValueStart* *leadingValueEnd* *followingValueStart* *followingValueEnd* *geoVeloStart* *geoVeloEnd* *geoAccelStart* *geoAccelEnd* *geoJerkEnd* *amplitude* *phaseEnd* *periodLength*
		Definition by frequency and phase in end point LCAMHDL_SINE_MODE_FREQUENCY_AND_PHASE_END	*leadingValueStart* *leadingValueEnd* *followingValueStart* *followingValueEnd* *geoVeloStart* *geoVeloEnd* *geoAccelStart* *geoAccelEnd* *geoJerkEnd* *amplitude* *phaseEnd* *frequency*
	Requirement: *leadingValueStart* < *leadingValueEnd* *phaseStart* <> *phaseEnd* *periodLength* <> *0.0* *frequency* <> *0.0*		
Double harmonic: LCAMHDL_ADD_PROFILE_DOUBLE_HARMONIC_D_R LCAMHDL_ADD_PROFILE_DOUBLE_HARMONIC_R_D **LCamHdl_AddCamDblHarmonic**	**Application:** dwell – reversal reversal – dwell **Requirement:** *leadingValueStart* < *leadingValueEnd* 2 <= numberOfInterpolationPoints <= 1000		*leadingValueStart* *leadingValueEnd* *followingValueStart* *followingValueEnd* *numberOfInterpolationPoints*

1.3 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

1.4 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) without the LCamHdl_CreateCamAdvanced function block in the code work-memory is 48 Kbytes, in the data work-memory 12 Kbytes and 688 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 31150	LCamHdl_AddCamSine	7		116
FB 31160	LCamHdl_AddCam10kSine	7		116
FB 31151	LCamHdl_AddCamInvSine	5		97
FB 31161	LCamHdl_AddCam10kInvSine	5		97
FB 31152	LCamHdl_AddCamPolynomial	31		354
FB 31162	LCamHdl_AddCam10kPolynomial	31		354
FB 31153	LCamHdl_AddCamDblHarmonic	5		91
FB 31163	LCamHdl_AddCam10kDblHarmonic	5		91
DB 31150	InstLCamHdl_AddCamSine		5	10
DB 31160	InstLCamHdl_AddCam10kSine		5	10
DB 31151	InstLCamHdl_AddCamInvSine		1	5
DB 31161	InstLCamHdl_AddCam10kInvSine		1	5
DB 31152	InstLCamHdl_AddCamPolynomial		5	10
DB 31162	InstLCamHdl_AddCam10kPolynomial		5	10
DB 31153	InstLCamHdl_AddCamDblHarmonic		1	5
DB 31163	InstLCamHdl_AddCam10kDblHarmonic		1	5

[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 with additional profiles.

Table 2-1: List of blocks

Block	Symbol	Classification
FB 31150	LCamHdl_AddCamSine	In-house development
FB 31160	LCamHdl_AddCam10kSine	In-house development
FB 31151	LCamHdl_AddCamInvSine	In-house development
FB 31161	LCamHdl_AddCam10kInvSine	In-house development
FB 31152	LCamHdl_AddCamPolynomial	In-house development
FB 31162	LCamHdl_AddCam10kPolynomial	In-house development
FB 31153	LCamHdl_AddCamDblHarmonic	In-house development
FB 31163	LCamHdl_AddCam10kDblHarmonic	In-house development

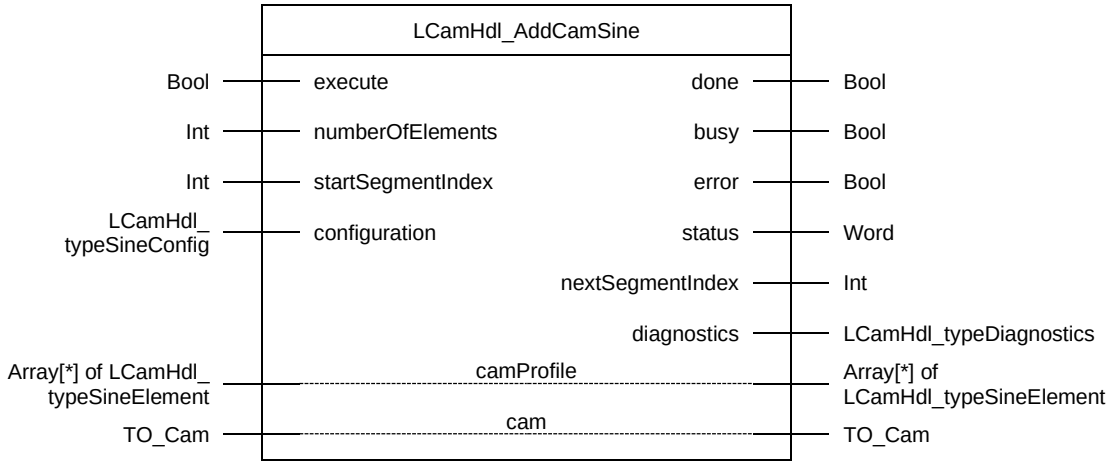
2.2 Explanation of the blocks

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

2.2.1 FB LCamHdl_AddCamSine (FB 31150)

Figure

Figure 2-1: Block diagram of *LCamHdl_AddCamSine*



Principle of operation

A sine or inclined sine segment can be created at runtime with a SIMATIC S7-1500T CPU with this type of function block. Gaps between the segments are interpolated by the runtime system with the interpolation method chosen in the selected technology object cam.

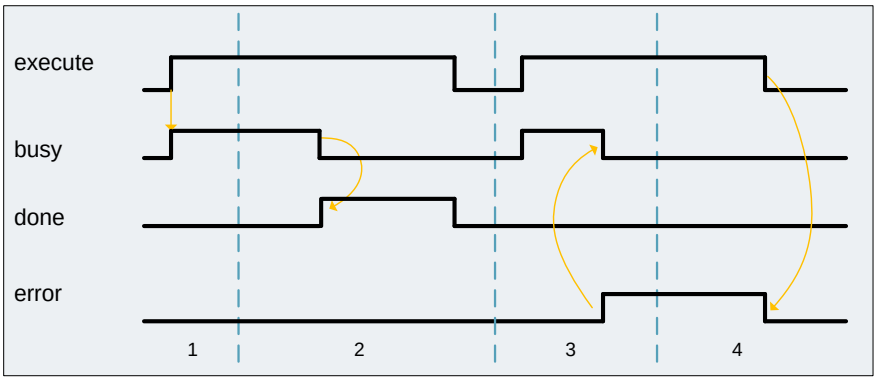
A maximum number of 50 segments in a cam profile can be used to define a cam.

The FB **LCamHdl_AddCamSine** can be configured to delete preceding or successive cam segments and cam points. In addition it is possible to interpolate the cam disk at the end with the *interpolateCam* configuration bit.

For more information about the connection of the CamHdl function blocks see chapter 4 Notes and Support.

Function characteristics

Figure 2-2: Timing diagram of the *LCamHdl_AddCamSine* 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_AddCamSine*

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)
configuration	LCamHdl_typeSineConfig	Configuration for interpolating the cam disk and deleting preceding or successive cam points/ segments

Output parameters

Table 2-3: Output parameters of *LCamHdl_AddCamSine*

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)
diagnostics	LCamHdl_typeDiagnostics	Diagnostics information of FB

InOut parameters

Table 2-4: InOut parameters of *LCamHdl_AddCamSine*

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

Status and error displays

Table 2-5: *LCamHdl_AddCamSine* 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_AddCamSine* error displays

Status	Meaning	Remedy / notes
16#8200	ERR_INVALID_PROFILE_TYPE	Invalid profile type in element no., see diagnostics.errorElementNo
16#8202	ERR_CAM_SEGMENTS_OUT_OF_BOUNDS	Maximum number of cam segments (50) of the technology object was exceeded
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#820B	ERR_INVALID_NUMBER_OF_ELEMENTS	Invalid number of cam elements
16#820D	ERR_INVALID_PERFORMANCE_MODE	Invalid performance mode parameter
16#8220	ERR_INVALID_DELETE_OPTION	Invalid delete option - check configuration
16#8222	ERR_INVALID_START_SEGMENT_INDEX	Invalid *startSegmentIndex*, 1<=*startSegmentIndex* <=50
16#8225	ERR_MISSING_PROFILE_DATA	No profile data available
16#8240	ERR_INVALID_SINE_MODE	Invalid profile mode in element no., see diagnostics.errorElementNo
16#8245	ERR_INVALID_PHASE	Invalid phase in element no., see diagnostics.errorElementNo
16#8246	ERR_INVALID_PERIOD_LENGTH	Invalid period length in element no., see diagnostics.errorElementNo
16#8247	ERR_INVALID_FREQUENCY	Invalid frequency in element no., see diagnostics.errorElementNo
16#824A	ERR_INVALID_INCLINATION_MODE	Invalid inclination mode in element no., see diagnostics.errorElementNo
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)

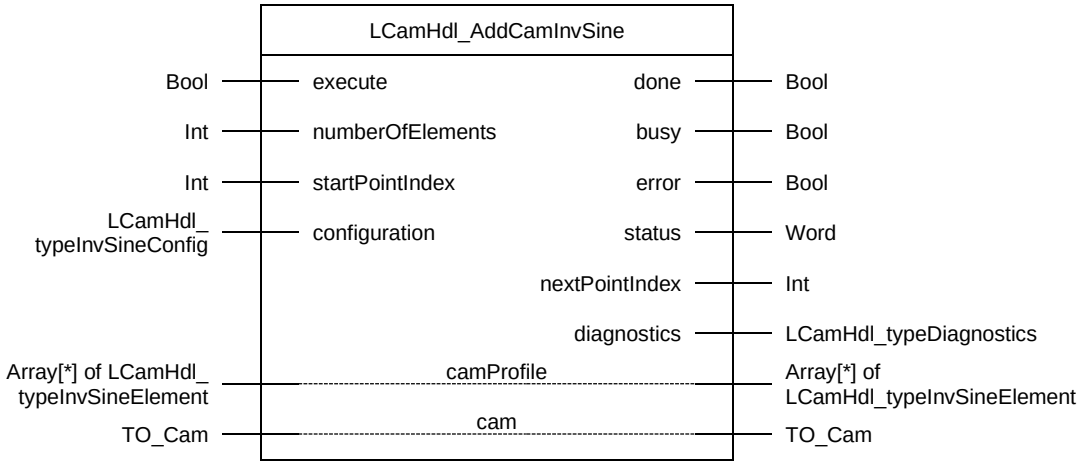
2.2.2 FB LCamHdl_AddCam10kSine (FB 31160)

The function block LCamHdl_AddCam10kSine is a copy of the function block LCamHdl_AddCamSine. The "..10k.." version enables using a cam technology object of type TO_Cam_10k instead of TO_Cam.

2.2.3 FB LCamHdl_AddCamInvSine (FB 31151)

Figure

Figure 2-3: Block diagram of *LCamHdl_AddCamInvSine*



Principle of operation

An inverse sine segment can be created at runtime with a SIMATIC S7-1500T CPU with this type of function block. An inverse sine is approximated using interpolation points of the arcsine function. Gaps between the points are interpolated by the runtime system with the interpolation method chosen in the selected technology object cam.

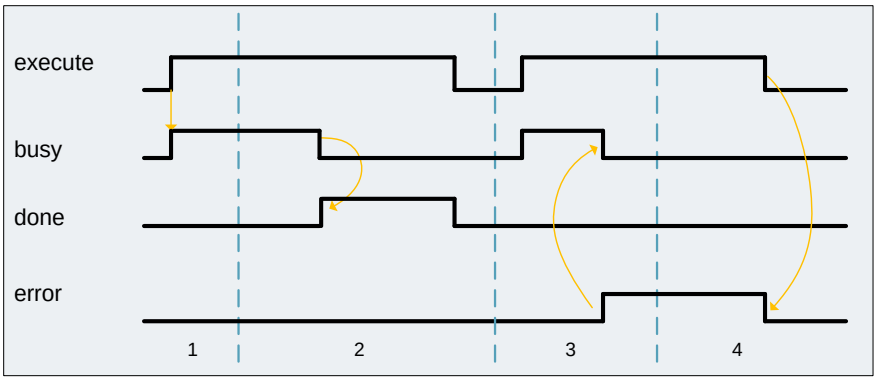
The generated points will be added as points in the cam disk. A maximum number of 1000 points in a cam profile can be used to define a cam.

The FB **LCamHdl_AddCamInvSine** 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.

For more information about the connection of the CamHdl function blocks see chapter 4 Notes and Support.

Function characteristics

Figure 2-4: Timing diagram of the *LCamHdl_AddCamInvSine* 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 2-7: Input parameters of *LCamHdl_AddCamInvSine*

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)
startPointIndex	Int	Start index of cam point data (1-1000) (default: 1)
configuration	LCamHdl_typeInvSineConfig	Configuration for interpolating the cam disk and deleting preceding or successive cam points/ segments

Output parameters

Table 2-8: Output parameters of *LCamHdl_AddCamInvSine*

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)
nextPointIndex	Int	Next "empty" point index (default: 0)
diagnostics	LCamHdl_typeDiagnostics	Diagnostics information of FB

InOut parameters

Table 2-9: InOut parameters of *LCamHdl_AddCamInvSine*

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

Status and error displays

Table 2-10: *LCamHdl_AddCamInvSine* 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 2-11: *LCamHdl_AddCamInvSine* 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#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#820B	ERR_INVALID_NUMBER_OF_ ELEMENTS	Invalid number of cam elements
16#820D	ERR_INVALID_PERFORMANCE_ MODE	Invalid performance mode parameter
16#8220	ERR_INVALID_DELETE_OPTION	Invalid delete option - check configuration
16#8223	ERR_INVALID_START_POINT_ INDEX	Invalid *startPointIndex*, 1 <= *startPointIndex* <= 1000
16#8225	ERR_MISSING_PROFILE_DATA	No profile data available
16#8227	ERR_INVALID_NUMBER_OF_ INTERPOLATION_POINTS	Invalid number of interpolation points in element no., see diagnostics.errorElementNo
16#8250	ERR_INVALID_DEFINITION_ RANGE	Invalid definition range in element no., see diagnostics.errorElementNo
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)

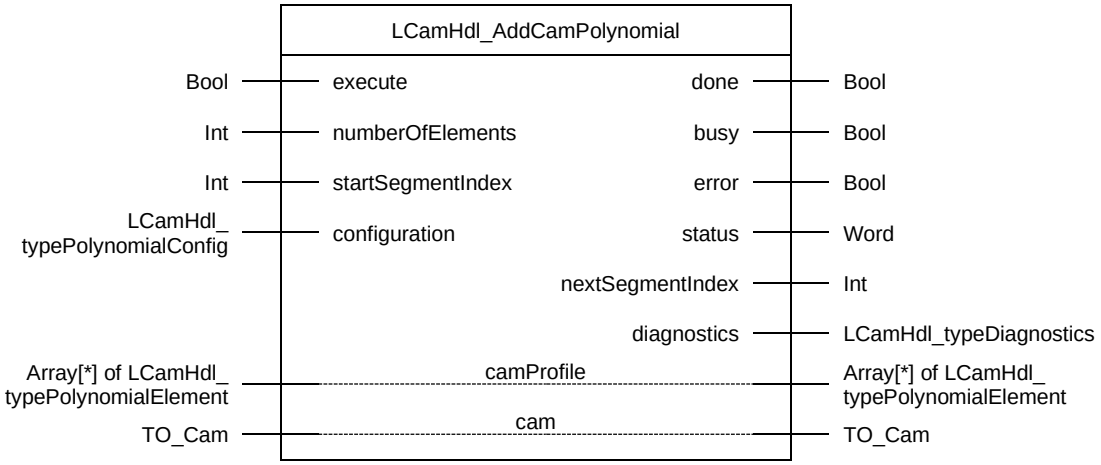
2.2.4 FB LCamHdl_AddCam10kInvSine (FB 31161)

The function block LCamHdl_AddCam10kInvSine is a copy of the function block LCamHdl_AddCamInvSine. 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.

2.2.5 FB LCamHdl_AddCamPolynomial (FB 31152)

Figure

Figure 2-5: Block diagram of *LCamHdl_AddCamPolynomial*



Principle of operation

A polynomial element with or without trigonometric portion can be created at runtime with a SIMATIC S7-1500T CPU with this type of function block. Gaps between the segments are interpolated by the runtime system with the interpolation method chosen in the selected technology object cam.

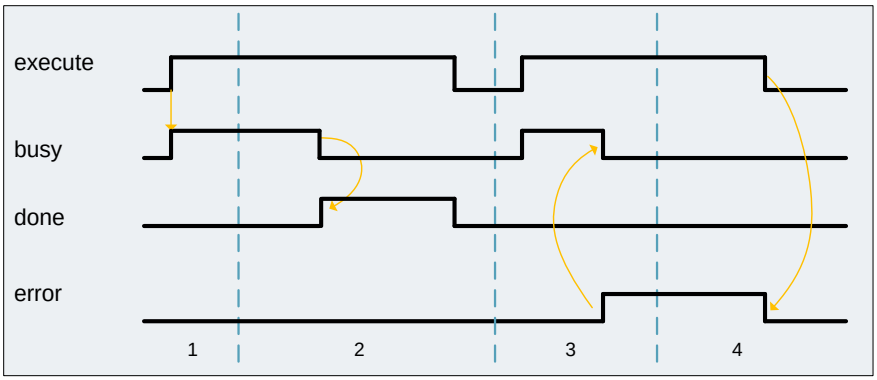
A maximum number of 50 segments in a cam profile can be used to define a cam.

The FB **LCamHdl_AddCamPolynomial** can be configured to delete preceding or successive cam segments and cam points. In addition it is possible to interpolate the cam disk at the end with the *interpolateCam* configuration bit.

For more information about the connection of the CamHdl function blocks see chapter 4 Notes and Support.

Function characteristics

Figure 2-6: Timing diagram of the *LCamHdl_AddCamPolynomial* 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 2-12: Input parameters of *LCamHdl_AddCamPolynomial*

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)
configuration	LCamHdl_typePolynomialConfig	Configuration for interpolating the cam disk and deleting preceding or successive cam points/ segments

Output parameters

Table 2-13: Output parameters of *LCamHdl_AddCamPolynomial*

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)
diagnostics	LCamHdl_typeDiagnostics	Diagnostics information of FB

InOut parameters

Table 2-14: InOut parameters of *LCamHdl_AddCamPolynomial*

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

Status and error displays

Table 2-15: *LCamHdl_AddCamPolynomial* 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 2-16: *LCamHdl_AddCamPolynomial* error displays

Status	Meaning	Remedy / notes
16#8200	ERR_INVALID_PROFILE_TYPE	Invalid profile type in element no., see diagnostics.errorElementNo
16#8202	ERR_CAM_SEGMENTS_OUT_OF_BOUNDS	Maximum number of cam segments (50) of the technology object was exceeded
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#820B	ERR_INVALID_NUMBER_OF_ELEMENTS	Invalid number of cam elements
16#820D	ERR_INVALID_PERFORMANCE_MODE	Invalid performance mode parameter
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#8225	ERR_MISSING_PROFILE_DATA	No profile data available
16#8230	ERR_INVALID_POLYNOMIAL_MODE	Invalid polynomial mode in element no., see diagnostics.errorElementNo
16#8240	ERR_INVALID_SINE_MODE	Invalid profile mode in element no., see diagnostics.errorElementNo
16#8245	ERR_INVALID_PHASE	Invalid phase in element no., see diagnostics.errorElementNo
16#8246	ERR_INVALID_PERIOD_LENGTH	Invalid period length in element no., see diagnostics.errorElementNo
16#8247	ERR_INVALID_FREQUENCY	Invalid frequency in element no., see diagnostics.errorElementNo
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)

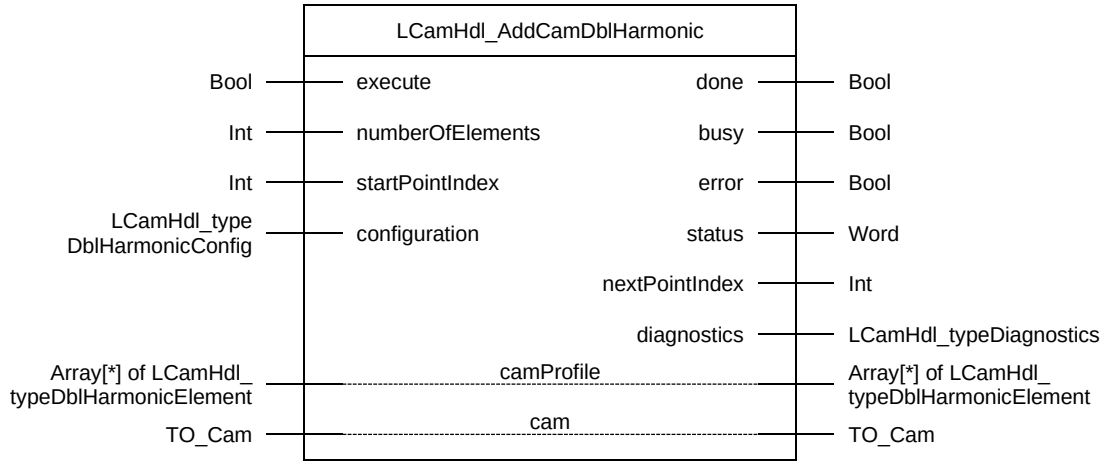
2.2.6 FB LCamHdl_AddCam10kPolynomial (FB 31162)

The function block LCamHdl_AddCam10kPolynomial is a copy of the function block LCamHdl_AddCamPolynomial. The "..10k.." version enables using a cam technology object of type TO_Cam_10k instead of TO_Cam.

2.2.7 FB LCamHdl_AddCamDblHarmonic (FB 31153)

Figure

Figure 2-7: Block diagram of *LCamHdl_AddCamDblHarmonic*



Principle of operation

A double harmonic segment can be created at runtime with a SIMATIC S7-1500T CPU with this type of function block. A double harmonic element is approximated using interpolation points of the cosine^4 function. Gaps between the points are interpolated by the runtime system with the interpolation method chosen in the selected technology object cam.

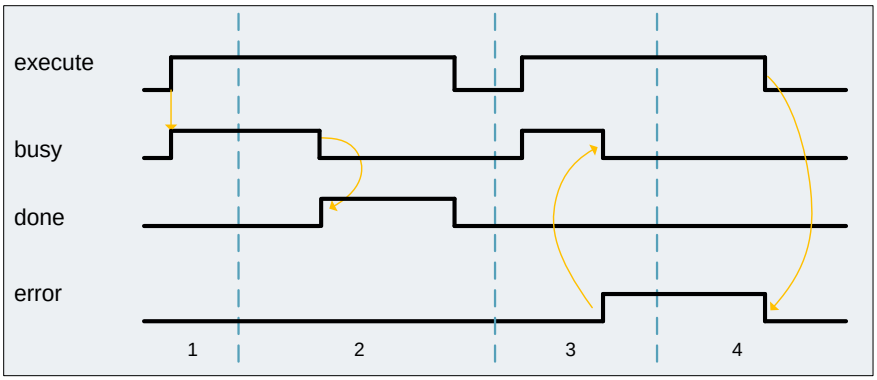
The generated points will be added as points in the cam disk. A maximum number of 1000 points in a cam profile can be used to define a cam.

The FB **LCamHdl_AddCamDblHarmonic** 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.

For more information about the connection of the CamHdl function blocks see chapter 4 Notes and Support.

Function characteristics

Figure 2-8: Timing diagram of the *LCamHdl_AddCamDblHarmonic* 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 2-17: Input parameters of *LCamHdl_AddCamDblHarmonic*

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)
startPointIndex	Int	Start index of cam point data (1-1000) (default: 1)
configuration	LCamHdl_type DblHarmonicConfig	Configuration for interpolating the cam disk and deleting preceding or successive cam points/ segments

Output parameters

Table 2-18: Output parameters of *LCamHdl_AddCamDblHarmonic*

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)
nextPointIndex	Int	Next "empty" point index (default: 0)
diagnostics	LCamHdl_typeDiagnostics	Diagnostics information of FB

InOut parameters

Table 2-19: InOut parameters of *LCamHdl_AddCamDblHarmonic*

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

Status and error displays

Table 2-20: *LCamHdl_AddCamDblHarmonic* 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 2-21: *LCamHdl_AddCamDblHarmonic* 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#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#820B	ERR_INVALID_NUMBER_OF_ ELEMENTS	Invalid number of cam elements
16#820D	ERR_INVALID_PERFORMANCE_ MODE	Invalid performance mode parameter
16#8220	ERR_INVALID_DELETE_OPTION	Invalid delete option - check configuration
16#8223	ERR_INVALID_START_POINT_ INDEX	Invalid *startPointIndex*, 1 <= *startPointIndex* <= 1000
16#8225	ERR_MISSING_PROFILE_DATA	No profile data available
16#8227	ERR_INVALID_NUMBER_OF_ INTERPOLATION_POINTS	Invalid number of interpolation points in element no., see diagnostics.errorElementNo
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)

2.2.8 FB LCamHdl_AddCam10kDblHarmonic (FB 31163)

The function block LCamHdl_AddCam10kDblHarmonic is a copy of the function block LCamHdl_AddCamDblHarmonic. 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.

2.2.9 PLC tags

LCamHdl – profile constants - segments

Table 2-22: User constants in *LCamHdl_ProfileConstants*

Name	Data type	Value	Comment
LCAMHDL_PROFILE_EMPTY	DInt	0	Empty
LCAMHDL_ADD_PROFILE_POLY	DInt	44	Polynomial
LCAMHDL_ADD_PROFILE_POLY_ TRIGONOMETRIC	DInt	45	Polynomial with trigonometric portion
LCAMHDL_ADD_PROFILE_SINE	DInt	51	Sine
LCAMHDL_ADD_PROFILE_INCLINED _SINE	DInt	52	Inclined sine

LCamHdl – profile constants - points

Table 2-23: User constants in *LCamHdl_ProfileConstants*

Name	Data type	Value	Comment
LCAMHDL_PROFILE_EMPTY	DInt	0	Empty
LCAMHDL_ADD_PROFILE_INVERSE _SINE	DInt	55	Inverse sine
LCAMHDL_ADD_PROFILE_D_R_ DOUBLE_HARMONIC	DInt	60	Dwell → reversal - double harmonic
LCAMHDL_ADD_PROFILE_R_D_ DOUBLE_HARMONIC	DInt	61	Reversal → dwell - double harmonic

LCamHdl – Config constants

Table 2-24: 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

LCamHdl – Additional configuration constants

Table 2-25: User constants in *LCamHdl_AdditionalConstants*

Name	Data type	Value	Comment
LCAMHDL_SINE_MODE_PHASE_START_AND_END	DInt	0	Sine mode: Phase at start and at end
LCAMHDL_SINE_MODE_PHASE_START_AND_PERIOD_LENGTH	DInt	1	Sine mode: Phase at start and period length
LCAMHDL_SINE_MODE_PHASE_START_AND_FREQUENCY	DInt	2	Sine mode: Phase at start and frequency
LCAMHDL_SINE_MODE_PERIOD_LENGTH_AND_PHASE_END	DInt	3	Sine mode: Period length and phase at end
LCAMHDL_SINE_MODE_FREQUENCY_AND_PHASE_END	DInt	4	Sine mode: Frequency and phase at end
LCAMHDL_SINE_INCL_MODE_DISPLACEMENT_START_AND_END	DInt	0	Sine inclination mode: Displacement at start and end
LCAMHDL_SINE_INCL_MODE_DISPLACEMENT_START_AND_INCLINATION	DInt	1	Sine inclination mode: Displacement at start and inclination
LCAMHDL_SINE_INCL_MODE_INCLINATION_AND_DISPLACEMENT_END	DInt	2	Sine inclination mode: Inclination and displacement at end
LCAMHDL_POLY_MODE_BOUNDARY_VALUES_AND_POINT_OF_INFLECTION	DInt	0	Polynomial definition by boundary values and point of inflection
LCAMHDL_POLY_MODE_BOUNDARY_VALUES_AND_START_JERK	DInt	1	Polynomial definition by boundary values and start jerk
LCAMHDL_POLY_MODE_BOUNDARY_VALUES_AND_END_JERK	DInt	2	Polynomial definition by boundary values and end jerk

2.2.10 PLC data types

LCamHdl_typeSineConfig

Table 2-26: Parameter of *LCamHdl_typeSineConfig*

Name	Data type	Value	Comment
deletePreceding CamSegmentData	DInt	0	Delete preceding cam segment data 0: Delete no data 1: Delete until first invalid data found 2: Delete all data
deleteSuccessive CamSegmentData	DInt	1	Delete successive cam segment data 0: Delete no data 1: Delete until first invalid data found 2: Delete all data
deleteCamPointData	DInt	0	Delete 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_typeSineElement

Table 2-27: Parameter of *LCamHdl_typeSineElement*

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
amplitude	LReal	0.0	Sine amplitude
periodLength	LReal	0.0	Sine period length
frequency	LReal	0.0	Sine frequency
phaseStart	LReal	0.0	Phase at start
phaseEnd	LReal	0.0	Phase at end
inclination	LReal	0.0	Inclination of inclined sine
displacementStart	LReal	0.0	Oscillation midpoint offset of the sine element / Displacement at start of the inclined sine
displacementEnd	LReal	0.0	Displacement at end of the inclined sine - only inclined sine
camProfileType	DInt	51	Profile type of the cam disk element , 51: LCAMHDL_ADD_PROFILE_SINE (default)
sineMode	DInt	0	Profile mode , 0: LCAMHDL_SINE_MODE_PHASE_START_AND_END (default)
inclinationMode	DInt	0	Inclination mode , 0: LCAMHDL_SINE_INCL_MODE_DISPLACEMENT_START_AND_END (default)

LCamHdl_typeInvSineConfig

Table 2-28: Parameter of *LCamHdl_typeInvSineConfig*

Name	Data type	Value	Comment
deletePrecedingCamPointData	DInt	0	Delete preceding cam point data 0: Delete no data 1: Delete until first invalid data found 2: Delete all data
deleteSuccessiveCamPointData	DInt	1	Delete successive cam point data 0: Delete no data 1: Delete until first invalid data found 2: Delete all data
deleteCamSegmentData	DInt	0	Delete cam segment 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_typeInvSineElement

Table 2-29: Parameter of *LCamHdl_typeInvSineElement*

Name	Data type	Value	Comment
numberOfInterpolationPoints	Int	32	Approximation - number of interpolation points
leadingValueStart	LReal	0.0	Leading value at the beginning of the element
leadingValueEnd	LReal	0.0	Leading value at the end of the element
followingValueMin	LReal	0.0	Following value minimum
followingValueMax	LReal	0.0	Following value maximum
definitionRangeStart	LReal	-0.95	Start point in the definition range of the arcsine function that is to be used
definitionRangeEnd	LReal	0.95	End point in the definition range of the arcsine function that is to be used
mirrored	Bool	FALSE	Select whether or not the inverse sine is to be mirrored about the abscissa
camProfileType	DInt	55	Profile type of the cam disk element , 55: LCAMHDL_ADD_PROFILE_INVERSE_SINE (default)

LCamHdl_typePolynomialConfig

Table 2-30: Parameter of *LCamHdl_typePolynomialConfig*

Name	Data type	Value	Comment
deletePrecedingCamSegmentData	DInt	0	Delete preceding cam segment data 0: Delete no data 1: Delete until first invalid data found 2: Delete all data
deleteSuccessiveCamSegmentData	DInt	1	Delete successive cam segment data 0: Delete no data 1: Delete until first invalid data found 2: Delete all data
deleteCamPointData	DInt	0	Delete 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_typePolynomialElement

Table 2-31: Parameter of *LCamHdl_typePolynomialElement*

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
amplitude	LReal	0.0	Sine amplitude
periodLength	LReal	0.0	Sine period length
frequency	LReal	0.0	Sine frequency
phaseStart	LReal	0.0	Sine phase at start

Name	Data type	Value	Comment
phaseEnd	LReal	0.0	Sine phase at end
camProfileType	DInt	45	Profile type of the cam disk element , 45: LCAMHDL_ADD_PROFILE_POLY_TRIGONOMETRIC (default)
polynomialMode	DInt	0	Polynomial mode , 0: LCAMHDL_POLY_MODE_BOUNDARY_VALUES_AND_POINT_OF_INFLECTION (default)
sineMode	DInt	0	Sine mode , 0: LCAMHDL_SINE_MODE_PHASE_START_AND_END (default)

LCamHdl_typeDblHarmonicConfig

Table 2-32: Parameter of *LCamHdl_typeDblHarmonicConfig*

Name	Data type	Value	Comment
deletePrecedingCamPointData	DInt	0	Delete preceding cam point data 0: Delete no data 1: Delete until first invalid data found 2: Delete all data
deleteSuccessiveCamPointData	DInt	1	Delete successive cam point data 0: Delete no data 1: Delete until first invalid data found 2: Delete all data
deleteCamSegmentData	DInt	0	Delete cam segment 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_typeDblHarmonicElement

Table 2-33: Parameter of *LCamHdl_typeDblHarmonicElement*

Name	Data type	Value	Comment
numberOfInterpolationPoints	Int	32	Approximation - number of interpolation points
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
camProfileType	DInt	60	Profile type of the cam disk element , 60: LCAMHDL_ADD_PROFILE_D_R_DOUBLE_HARMONIC (default)

LCamHdl_typeDiagnostics

Table 2-34: 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

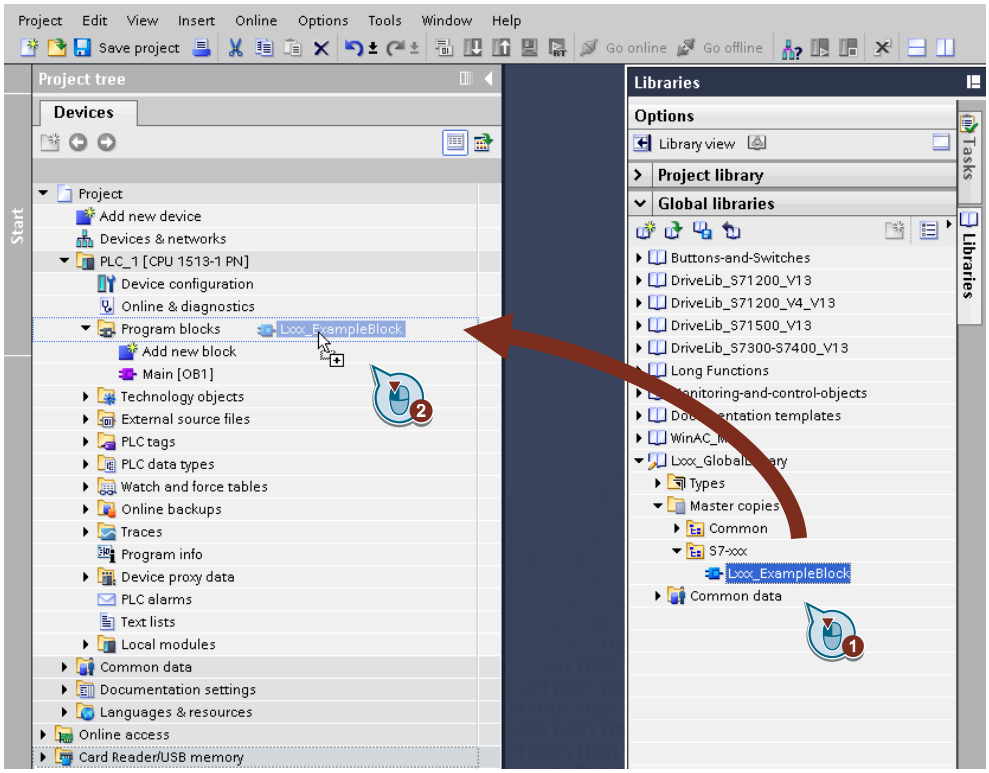


Table 3-2: Integrating the library blocks into STEP 7

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_AddCamXXX* 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)'

Connection of the CamHdl function blocks

For using the additional function blocks it is not necessary to use the *LCamHdl_CreateCamAdvanced* function block. The additional function blocks as described in this documentation could be used as a standalone function.

To add one of the cam elements of the additional function blocks to the elements of the advanced or other additional function blocks, the following parameters must be connected:

Table 4-1: Connection parameters

Preceding function block	Successive function block
nextPointIndex/nextSegmentIndex	startPointIndex/startSegmentIndex
done	execute

After successfully adding the cam points and segments for example with the *LCamHdl_CreateCamAdvanced* which is indicated with *done,* the next function block can be executed. The *startPointIndex* or *startSegmentIndex* of the following function block is the nextPointIndex / nextSegmentIndex of the preceding function block. For performance reasons, only the last called block should interpolate the cam disk.

The default configuration for deleting preceding segments/ points is "LCAMHDL_DELETE_NO_DATA". For the successive segments/ points it is "LCAMHDL_DELETE_DATA_UNTIL_FIRST_INVALID_DATA_FOUND". For performance reasons the configuration of the successive segments/ points must be changed. Only the last called segment based function block (**LCamHdl_AddCamSine**, **LCamHdl_AddCamPolynomial**) must delete the successive segments and the last called point based function block (**LCamHdl_AddCamDbLHarmonic**, **LCamHdl_AddCamInvSine**) must delete all successive points.

NOTE For the best runtime performance only the last called function block must delete successive segments or points.

4.1 Application example

The example requires a technology object cam named "Cam_Additional" in the runtime system of the SIMATIC S7-1500T.

The cam consists of two sine elements, a double harmonic dwell – reversal and reversal –dwell element. These elements got a transition with 5th degree polynomials.

Element 1: dwell – constant velocity → 5th degree polynomial

Element 2: sine element → **LCamHdl_AddCamSine**

Element 3: constant velocity – dwell → 5th degree polynomial

Element 4: dwell – reversal → Double harmonic element

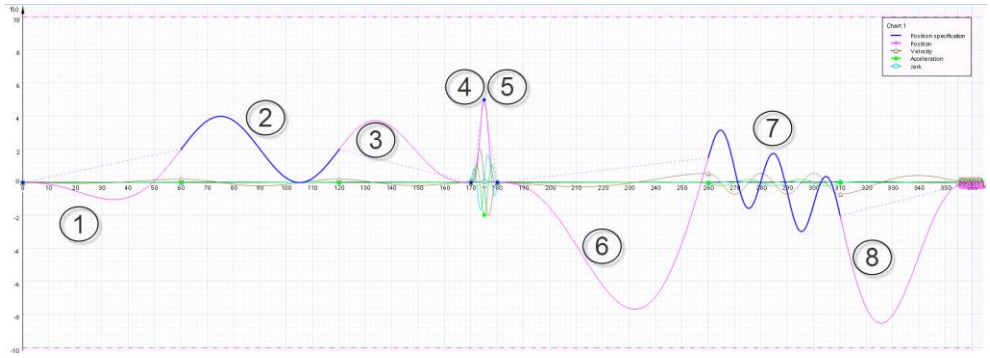
Element 5: reversal – dwell → Double harmonic element

Element 6: dwell – motion → 5th degree polynomial

Element 7: inclined sine element → **LCamHdl_AddCamSine**

Element 8: motion – dwell → 5th degree polynomial

Figure 4-1



For creation of this type of cam disk you need the **LCamHdl_CreateCamAdvanced**, **LCamHdl_AddCamSine** and **LCamHdl_AddCamDblHarmonic** function block.

The interconnection between these three function blocks, configuration and parameterizing is shown in the following source code:

Source code

```
FUNCTION_BLOCK "LCamHdl_Example_3"
{ S7_Optimized_Access := 'TRUE' }
VERSION : 1.1
   VAR
      statAdvancedProfiles : Array[1..4] of
"LCamHdl_typeAdvancedElement";   // Cam profile for advanced
function block
      statAdvancedConfiguration : "LCamHdl_typeAdvancedConfig";   //
Configuration for advanced function block
      statSineProfiles : Array[1..2] of "LCamHdl_typeSineElement";
// Cam profile for additional sine function block
```

```
      statSineConfiguration : "LCamHdl_typeSineConfig";   //
Configuration for add sine function block
      statDblHarmonicProfiles : Array[1..2] of
"LCamHdl_typeDblHarmonicElement";   // Cam profile for additional
double harmonic function block
      statDblHarmonicConfiguration :
"LCamHdl_typeDblHarmonicConfig";   // Configuration for add double
harmonic function block
      instLCamHdl_CreateCamAdvanced : "LCamHdl_CreateCamAdvanced";
// Instance of create cam advanced function block
      instLCamHdl_AddCamSine : "LCamHdl_AddCamSine";  // Instance of
add cam sine function block
      instLCamHdl_AddCamDblHarmonic : "LCamHdl_AddCamDblHarmonic";
// Instance of add cam double harmonic function block
   END_VAR


BEGIN
   // Configuration
   #statAdvancedConfiguration.deleteSuccessiveCamPointData :=
"LCAMHDL_DELETE_NO_DATA"; // No deletion because
LCamHdl_AddCamDblHarmonic function block will delete cam disk points
   #statAdvancedConfiguration.deleteSuccessiveCamSegmentData :=
"LCAMHDL_DELETE_NO_DATA"; // No deletion because LCamHdl_AddCamSine
function block will delete cam disk segments
   #statAdvancedConfiguration.interpolateCam := FALSE; // No
interpolation because LCamHdl_AddCamDblHarmonic will interpolate the
cam disk
   #statSineConfiguration.interpolateCam := FALSE; // No
interpolation because LCamHdl_AddCamDblHarmonic will interpolate the
cam disk
   #statDblHarmonicConfiguration.interpolateCam := TRUE;


   // 1st element
   #statAdvancedProfiles[1].leadingValueStart := 0.0;
   #statAdvancedProfiles[1].leadingValueEnd := 60.0;
   #statAdvancedProfiles[1].followingValueStart := 0.0;
   #statAdvancedProfiles[1].followingValueEnd := 2.0;
   #statAdvancedProfiles[1].geoVeloStart := 0.0;
   #statAdvancedProfiles[1].geoVeloEnd:= 0.20944;
   #statAdvancedProfiles[1].camProfileType :=
"LCAMHDL_PROFILE_POLY_5";


   // 2nd element (Sine)
   #statSineProfiles[1].leadingValueStart := 60.0;
   #statSineProfiles[1].leadingValueEnd := 120.0;
   #statSineProfiles[1].sineMode :=
"LCAMHDL_SINE_MODE_PHASE_START_AND_END";
   #statSineProfiles[1].amplitude := 2.0;
   #statSineProfiles[1].phaseStart := 0.0;
   #statSineProfiles[1].phaseEnd := 360.0;
   #statSineProfiles[1].displacementStart := 2.0;
   #statSineProfiles[1].camProfileType :="LCAMHDL_PROFILE_SINE";
```

```
   // 3rd element
   #statAdvancedProfiles[2].leadingValueStart := 120.0;
   #statAdvancedProfiles[2].leadingValueEnd := 170.0;
   #statAdvancedProfiles[2].followingValueStart := 2.0;
   #statAdvancedProfiles[2].followingValueEnd := 0.0;
   #statAdvancedProfiles[2].geoVeloStart := 0.20944;
   #statAdvancedProfiles[2].geoVeloEnd := 0.0;
   #statAdvancedProfiles[2].camProfileType :=
"LCAMHDL_PROFILE_POLY_5";

   // 4th element (D_R double harmonic)
   #statDblHarmonicProfiles[1].leadingValueStart :=170.0;
   #statDblHarmonicProfiles[1].leadingValueEnd := 175.0;
   #statDblHarmonicProfiles[1].followingValueStart :=0.0;
   #statDblHarmonicProfiles[1].followingValueEnd :=5.0;
   #statDblHarmonicProfiles[1].numberOfInterpolationPoints := 32;
   #statDblHarmonicProfiles[1].camProfileType :=
"LCAMHDL_PROFILE_D_R_DOUBLE_HARMONIC";

   // 5th element (R_D double harmonic)
   #statDblHarmonicProfiles[2].leadingValueStart := 175.0;
   #statDblHarmonicProfiles[2].leadingValueEnd := 180.0;
   #statDblHarmonicProfiles[2].followingValueStart := 5.0;
   #statDblHarmonicProfiles[2].followingValueEnd := 0.0;
   #statDblHarmonicProfiles[2].numberOfInterpolationPoints := 32;
   #statDblHarmonicProfiles[2].camProfileType :=
"LCAMHDL_PROFILE_R_D_DOUBLE_HARMONIC";

   // 6th element
   #statAdvancedProfiles[3].leadingValueStart := 180.0;
   #statAdvancedProfiles[3].leadingValueEnd := 260.0;
   #statAdvancedProfiles[3].followingValueStart := 0.0;
   #statAdvancedProfiles[3].followingValueEnd := 1.5;
   #statAdvancedProfiles[3].geoVeloStart := 0.0;
   #statAdvancedProfiles[3].geoVeloEnd := 0.558318;
   #statAdvancedProfiles[3].geoAccelStart := 0.0;
   #statAdvancedProfiles[3].geoAccelEnd := -0.000236;
   #statAdvancedProfiles[3].camProfileType :=
"LCAMHDL_PROFILE_POLY_5";

   // 7th element (inclined Sine)
   #statSineProfiles[2].leadingValueStart := 260.0;
   #statSineProfiles[2].leadingValueEnd :=310.0;
   #statSineProfiles[2].sineMode :=
"LCAMHDL_SINE_MODE_PHASE_START_AND_FREQUENCY";
   #statSineProfiles[2].amplitude := 2.0;
   #statSineProfiles[2].phaseStart := 0.0;
```

```
   #statSineProfiles[2].frequency := 0.05;
   #statSineProfiles[2].inclinationMode :=
"LCAMHDL_SINE_INCL_MODE_DISPLACEMENT_START_AND_END";
   #statSineProfiles[2].displacementStart := 1.5;
   #statSineProfiles[2].displacementEnd := -2.0;
   #statSineProfiles[2].camProfileType :=
"LCAMHDL_PROFILE_INCLINED_SINE";


   // 8th element
   #statAdvancedProfiles[4].leadingValueStart := 310.0;
   #statAdvancedProfiles[4].leadingValueEnd := 360.0;
   #statAdvancedProfiles[4].followingValueStart := -2.0;
   #statAdvancedProfiles[4].followingValueEnd := 0.0;
   #statAdvancedProfiles[4].geoVeloStart := -0.698318;
   #statAdvancedProfiles[4].geoVeloEnd := 0.0;
   #statAdvancedProfiles[4].geoAccelStart := -0.000204;
   #statAdvancedProfiles[4].geoAccelEnd := 0.0;
   #statAdvancedProfiles[4].camProfileType :=
"LCAMHDL_PROFILE_POLY_5";


   // Call of function blocks
   #instLCamHdl_CreateCamAdvanced(execute := TRUE,
                                  configuration :=
#statAdvancedConfiguration,
                                  camProfile:=#statAdvancedProfiles,
                                  cam:="Cam_Additional");
   #instLCamHdl_AddCamSine(execute :=
#instLCamHdl_CreateCamAdvanced.done,
                           startSegmentIndex :=
#instLCamHdl_CreateCamAdvanced.nextSegmentIndex,
                           configuration := #statSineConfiguration,
                           camProfile:=#statSineProfiles,
                           cam:="Cam_Additional");
   #instLCamHdl_AddCamDblHarmonic(execute
:=#instLCamHdl_AddCamSine.done,
                                  startPointIndex :=
#instLCamHdl_CreateCamAdvanced.nextPointIndex,
                                  configuration :=
#statDblHarmonicConfiguration,

camProfile:=#statDblHarmonicProfiles,
                                  cam:="Cam_Additional");


END_FUNCTION_BLOCK
```

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
V1.1	11/2017	First version
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 blocks for handling the new cam technology object of type TO_Cam_10k