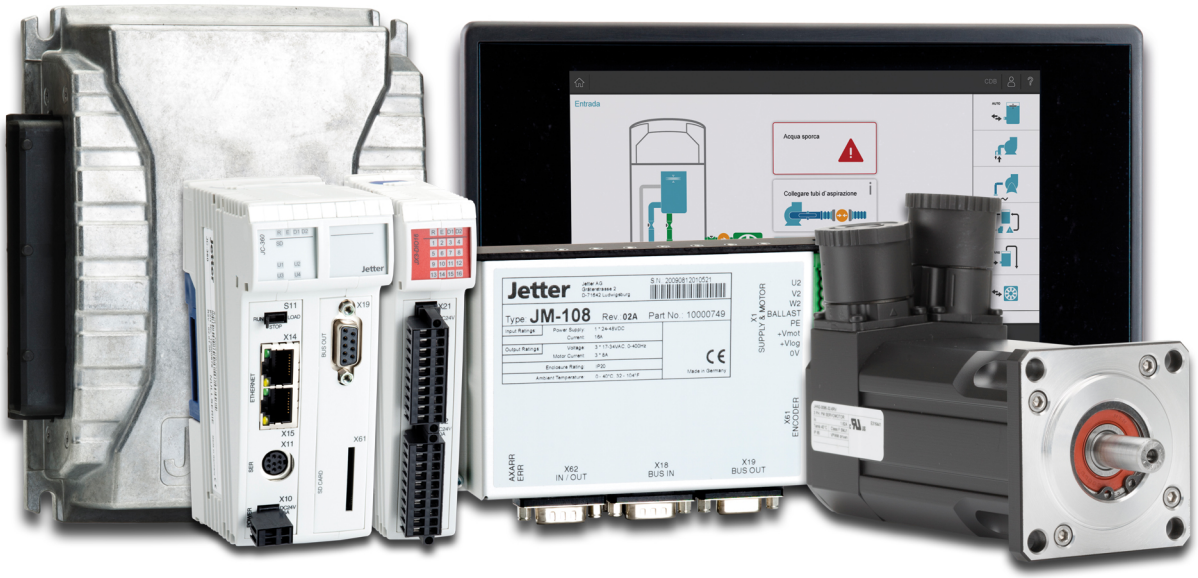
Jetter automation



Application Note - APN 047

Description of the Motion-API command "Read cam discs"

60880723

We automate your success.

Application Note: 047
Item # 60880723
Revision 1.01
August 2015 / Printed in Germany

This document has been compiled by Jetter AG with due diligence, and based on the known state of the art.

In the case of modifications, further developments or enhancements to products shipped in the past, a revised document will be supplied only if required by law, or deemed appropriate by Jetter AG. Jetter AG shall not be liable for errors in form or content, or for missing updates, as well as for damages or disadvantages resulting from such failure.

The logos, brand names, and product names mentioned in this document are trademarks or registered trademarks of Jetter AG, of associated companies or other title owners and must not be used without consent of the respective title owner.

Table of Contents

1 Reading cam discs 5

1.1 Definitions referring to cam disc 6
CamSegmentData 7
CamEvaluateData 11
CamAnalyseData 13
CamAnalyseDerivatesData 16

1 Reading cam discs

Introduction

This chapter describes how to read out a cam disc by means of the Motion API. In the Motion API, there are two structures for cam discs:

- Structures for defining cam discs
- Structures for reading cam discs

Contents

Topic	Page
Definitions referring to cam disc	6

1.1 Definitions referring to cam disc

Introduction

The Motion API library contains functions for evaluation of defined cam discs. These functions return data as values within corresponding structures. This chapter describes the applied structures in detail.

The structures in detail

In the Motion API, structures for reading cam discs have got the following names:

```
{
    // ...
    Struct CamSegmentData
    Struct CamEvaluateData
    Struct CamAnalyseData
    Struct CamAnalyseDerivatesData
    //...
}
```

Access to the structures

You can read these data by attaching to each cam disc function both the profile number (CamID) and the segment number (SegmentID). For accessing the `MCTechnoCam` structures, declare the variables. You can have these variables displayed in a setup screen, for example.

```
Var
    MyCamSegmentData            : CamSegmentData;
    MyCamEvaluateData           : CamEvaluateData;
    MyCamAnalyseData            : CamAnalyseData;
    MyCamAnalyseDerivatesData   : CamAnalyseDerivatesData;
End_Var;
```

Contents

Topic	Page
CamSegmentData	7
CamEvaluateData	11
CamAnalyseData	13
CamAnalyseDerivatesData	16

CamSegmentData

Introduction

The feedback value of the function `ReadSegment` supplies the following data structure for being read:

- Segment start
- Offset of the leading and shadow axis
- Coefficients of the cam disc
- Range of the leading axis

If you have to check data, for example, you can read the coefficients of a previously defined polynomial or the offset position of a shadow axis by means of the structure `CamSegmentData`.

Corresponding Motion API command

The structure `CamSegmentData` corresponds with the following function:

- MCTechnoCam.**ReadSegment**()

`ReadSegment` is a means for controlling defined cam discs. This way you can check, whether the cam disc takes the planned course. JetSym, for example, lets you read and graphically present this course.

Definition of the structure CamSegmentData

The structure `CamSegmentData` has been defined in the Motion API file **MCTechnoCamStructs.stxp**.

```
CamSegmentData : Struct
    SegmentType    : Int;
    SegmentStart   : Double;
    Option         : Int;
    MasterOffset   : Double;
    FollowerOffset : Double;
    Coefficient    : Array[0..7] Of Double;
    MasterCamRange : Double;
End_Struct;
```

CamSegmentData - Memory structure

Members	Description	Data type
SegmentType	Type of the segment function: 1: Straight line 2: 3rd order polynomial 3: 5th order polynomial 4: 7th order polynomial 5: 7th order autopolynomial 6: Sin 7: 3rd order autopolynomial 8: 5th order autopolynomial 9: 1st order autopolynomial (straight line)	Int
SegmentStart	Segment start	Double
Options	Possible options: 0: None (default) 1: Automatically appending the segment 2: User-specific cam range of the leading axis	Int
MasterOffset	Offset of the leading axis	Double
FollowerOffset	Offset of the shadow axis	Double
Coefficient	0: 1. Coefficient/leading axis range 1: 2. Coefficient 2: 3. Coefficient 3: 4. Coefficient 4: 5. Coefficient 5: 6. Coefficient 6: 7. Coefficient 7: 8. Coefficient	Double
MasterCamRange	Range of the leading axis	Double

The function `MCTechnoCam.ReadSegment()` reads data and saves these as return values to the structure `CamSegmentData`.

Notes

- In case of the function AutopolyX, not all basic parameters (derivatives), but the polynomial coefficients are read back.
- For the first segment, the leading axis range is read, for further segments the leading axis modulo range is read.

Example of an equation including a straight line

You have defined a cam disc by means of function type **straight line**. If you read back the coefficients, two coefficients for the functions will be stored to `CamSegmentData.Coefficient[...]`.

$$y(x) = a \cdot x + b$$

Saved coefficients: a, b

Example: 7th order polynomial

You have defined a cam disc by means of function type **7th order polynomial**. If you read back the coefficients, eight coefficients for the functions will be stored to C`amSegmentData.Coefficient[...]`.

$$y(x) = ax^7 + bx^6 + cx^5 + dx^4 + ex^3 + fx^2 + gx + h$$

Saved coefficients: a, b, c, d, e, f, g, h

Order of the coefficients

The coefficients have been stored in ascending order. This means that, for example, in case of a **7th order polynomial** the eighth coefficient has got the highest order:

```
CamSegmentData.Coefficient[0] = Coefficient h
CamSegmentData.Coefficient[1] = Coefficient g
...
CamSegmentData.Coefficient[6] = Coefficient b
CamSegmentData.Coefficient[7] = Coefficient a
```

In case of a **straight line,** the coefficients have been stored as follows:

```
CamSegmentData.Coefficient[0] = Coefficient b
CamSegmentData.Coefficient[1] = Coefficient a
```

Coefficients that are not needed contain value zero.

JetSym STX program

By means of Hardware Manager, you have created a technology group `tecGruppe` consisting of the leading axis`axLeitachse` and the cam disc shadow axis `axFolgeachse`. The short program outputs the value of the segment:

```
Task tReadSegment
Var
    CamSegData:    CamSegmentData;
    i:             Int;
End_Var;

// Reading the data 'ReadSegment':
tecGruppe.Coupling.Cam.ReadSegment(axFolgeachse,1,1);

// Data output:
Trace('Segmenttyp: ' + FloatToStr(CamSegData.SegmentType));
Trace('Segmenstart: ' + FloatToStr(CamSegData.SegmentStart));
Trace('Optionen: ' + FloatToStr(CamSegData.Option));
Trace('Offset der Leitachse: ' +
FloatToStr(CamSegData.MasterOffset));
Trace('Offset der Folgeachse: ' +
FloatToStr(CamSegData.FollowerOffset));
For i := 0 To 7 Do
    Trace('Koeffizient ' + IntToStr(i) + ' :' +
    FloatToStr(CamSegData.Coefficient[i]));
End_For;
Trace('Leitachsbereich: ' +
FloatToStr(CamSegData.MasterCamRange));
End_Task;
```

CamEvaluateData

Introduction

This structure lets you read the following data:

- Position of the shadow axis at set leading axis position
- Segment number
- Range of the leading axis

Corresponding Motion API command

The structure `CamEvaluateData` corresponds with the following function:

- MCTechnoCam.**Evaluate**()

The Motion API command `Evaluate` reads the position of the shadow axis. You set the position of the leading axis.

Definition of the structure CamEvaluateData

The structure `CamEvaluateData` has been defined in the Motion API file **MCTechnoCamStructs.stxp**.

```
CamEvaluateData : Struct
    FollowerPosition  : Double;
    SegmentNr         : Int;
    MasterCamRange    : Double;
End_Struct;
```

CamEvaluateData - Memory structure

Members	Description	Data type
FollowerPosition	Position of the shadow axis	Double
SegmentNr	Segment start	Int
MasterCamRange	Range of the leading axis	Double

The function `MCTechnoCam.Evaluate()` reads data and saves these as return values to the structure `CamEvaluateData`.

JetSym STX program

By means of Hardware Manager, you have created a technology group `tecGruppe` consisting of the leading axis`axLeitachse` and the cam disc shadow axis `axFolgeachse`. The task reads data of the shadow axis at set leading axis.

```
Var
    CamEvalData:        CamEvaluateData;
    tecGruppe:          Pointer To MCTechno;
End_Var;

Task tEvalData
// Coupling.Cam.Evaluate is called by the following parameters:
// Coupling.Cam.Evaluate(FollowerAxis, CamId, MasterPosition)
// The leading axis position is set, the shadow axis position,
// segment number and leading axis range are read.
// Function call-up:
// Reading the values for cam disc 2 at leading axis position 150.0:
CamEvalData:= tecGruppe.Coupling.Cam.Evaluate(axFolgeachse, 2,
150.0);
// Results output:
Trace('Position der Folgeachse: '+
FloatToStr(CamEvalData.FollowerPosition);
Trace('SegmentNr: ', + IntToStr(CamEvalData.SegmentNr));
Trace('Leitachsbereich: ' +
FloatToStr(CamEvalData.MasterCamRange));
End_Task;
```

CamAnalyseData

Introduction

This structure lets you read the following data:

- Segment start
- Selected option in the analyzed segment
- Offsets of the leading axis
- Offset of the shadow axis
- Minimum value of the segment
- Maximum value of the segment

Corresponding Motion API command

The structure `CamAnalyseData` corresponds with the following function:

- MCTechnoCam.**CamAnalyseSegment()**

Definition of the structure CamAnalyseData

The structure `CamAnalyseSegment` has been defined in the Motion API file **MCTechnoCamStructs.stxp**.

```
CamAnalyseData : Struct
    SegmentType    : Int;
    SegmentStart   : Double;
    Option         : Int;
    MasterOffset   : Double;
    FollowerOffset : Double;
    MinValue       : Double;
    MaxValue       : Double;
End_Struct;
```

CamAnalyseData - Memory structure

Members	Description	Data type
SegmentType	Type of the segment function: 1: Straight line 2: 3rd order polynomial 3: 5th order polynomial 4: 7th order polynomial 5: 7th order autopolynomial 6: Sin 7: 3rd order autopolynomial 8: 5th order autopolynomial 9: 1st order autopolynomial (straight line)	Int
SegmentStart	Segment start	Double
Options	Possible options: 0: None (default) 1: Automatically appending the segment 2: User-specific range of the leading axis	Int (0 ... 2)
MasterOffset	Offset of the leading axis	Double
FollowerOffset	Offset of the shadow axis	Double
MinValue	Smallest value in the segment	Double
MaxValue	Greatest value in the segment	Double

The function `MCTechnoCam.AnalyseSegment()` reads values for the structure `CamAnalyseData`.

JetSym STX program

By means of Hardware Manager, you have created a technology group `tecGruppe` consisting of the leading axis`axLeitachse` and the cam disc shadow axis `axFolgeachse`.

```
Task tAnalyseCAM
Var
    CamAnalyseSegData:    CamAnalyseData;
End_Var;

// Function call-up
// Reading the values for cam disc 1, segment 1:
tecGruppe.Coupling.Cam.AnalyseSegment(axFolgeachse,1,1);
// Evaluation:
Trace('Segmenttyp: ' + IntToStr(CamAnalyseSegData.SegmentType));
Trace('Beginn des Segments: ' +
FloatToStr(CamAnalyseSegData.SegmentStart));
Trace('Option: ' + IntToStr(CamAnalyseSegData.Option ));
Trace('Offset der Leitachse: ' +
FloatToStr(CamAnalyseSegData.MasterOffset));
Trace('Offset der Folgeachse: ' +
FloatToStr(CamAnalyseSegData.FollowerOffset));
Trace('Kleinster Wert im Segment: ' +
FloatToStr(CamAnalyseSegData.MinValue));
Trace('Größter Wert im Segment: ' +
FloatToStr(CamAnalyseSegData.MaxValue));
End_Task;
```

CamAnalyseDerivatesData

Introduction

This structure lets you read the following data:

- Segment start
- Selected option in the analyzed segment
- Offsets of the leading axis
- Offset of the shadow axis
- Minimum values of the derivatives
- Maximum values of the derivatives
- Range of the leading axis

Application example

If you have used the function type **Autopolynomial** in the definition of the cam disc, read the derivatives segment by segment. This way, expensive software updates are not necessary anymore.

Corresponding Motion API command

The structure `CamAnalyseDerivatesData` corresponds with the following function:

- MCTechnoCam.**AnalyseSegmentDerivates()**

Definition of the structure CamAnalyseDerivatesData

The structure `CamAnalyseDerivatesData` has been defined in the Motion API file **MCTechnoCamStructs.stxp**.

```
CamAnalyseDerivatesData: Struct
    SegmentType    : Int;
    SegmentStart   : Double;
    Option         : Int;
    MasterOffset   : Double;
    FollowerOffset : Double;
    MinValue       : Array[0..2] Of Double;
    MaxValue       : Array[0..2] Of Double;
    MasterCamRange : Double;
End_Struct;
```

CamAnalyseDerivates Data - Memory structure

Members	Description	Data type
SegmentType	Type of the segment function 1: Straight line 2: 3rd order polynomial 3: 5th order polynomial 4: 7th order polynomial 5: 7th order autopolynomial 6: Sin 7: 3rd order autopolynomial 8: 5th order autopolynomial 9: 1st order autopolynomial (straight line)	Int
SegmentStart	Segment start	Double
Options	Possible options: 0: None (default) 1: Automatically appending the segment 2: User-specific cam range of the leading axis	Int (0 ... 2)
MasterOffset	Offset of the leading axis	Double
FollowerOffset	Offset of the shadow axis	Double
MinValue	Smallest value of derivatives	Array[0..2] of Double
MaxValue	Greatest value of derivatives	Array[0..2] of Double

The function `MCTechnoCam.AnalyseSegmentDerivates()` reads values for the structure `CamAnalyseDerivates`.

JetSym STX program

By means of Hardware Manager, you have created a technology group `tecGruppe` consisting of the leading axis`axLeitachse` and the cam disc shadow axis `axFolgeachse`.

```
Task tAnalyseCAMDerivates

Var
    CamAnalyseDerivates:    CamAnalysederivatesData;
    i:                      Int;
End_Var;

// Function call-up
// Reading the values for cam disc 1, segment 1:
tecGruppe.Coupling.Cam.AnalyseSegmentDerivates(axFolgeachse,1,1)
;

// Evaluation:
Trace('Segmenttyp: ' +
IntToStr(CamAnalyseDerivates.SegmentType));
Trace('Beginn des Segments: ' +
FloatToStr(CamAnalyseDerivates.SegmentStart));
Trace('Option: ' + IntToStr(CamAnalyseDerivates.Option));
Trace('Offset der Leitachse: ' +
FloatToStr(CamAnalyseDerivates.MasterOffset));
Trace('Offset der Folgeachse: ' +
FloatToStr(CamAnalyseDerivates.FollowerOffset));
For i := 0 To 2 Do
    Trace('Kleinster Wert der ' + IntToStr(i) + ' Ableitung ist: '
    + FloatToStr(CamAnalyseDerivates.MinValue[i]));
    Trace('Größter Wert der ' + IntToStr(i) + ' Ableitung ist: ' +
    FloatToStr(CamAnalyseDerivates.MaxValue[i]));
End_For;
End_Task;
```

Jetter automation

Jetter AG
Graeterstrasse 2
71642 Ludwigsburg | Germany

Phone +49 7141 2550-0
Fax +49 7141 2550-425
info@jetter.de
www.jetter.de

We automate your success.