



# DOKUMENTATION ISG-kernel

## Handbuch Zyklen - Zusatzzyklen

Kurzbezeichnung:  
CYCLES-ADC

# Vorwort

## Rechtliche Hinweise

---

Diese Dokumentation wurde sorgfältig erstellt. Die beschriebenen Produkte und der Funktionsumfang werden jedoch ständig weiterentwickelt. Wir behalten uns das Recht vor, die Dokumentation jederzeit und ohne Ankündigung zu überarbeiten und zu ändern.

Aus den Angaben, Abbildungen und Beschreibungen in dieser Dokumentation können keine Ansprüche auf Änderung bereits gelieferter Produkte geltend gemacht werden.

## Qualifikation des Personals

---

Diese Beschreibung wendet sich ausschließlich an ausgebildetes Fachpersonal der Steuerungs-, Automatisierungs- und Antriebstechnik, das mit den geltenden Normen, der zugehörigen Dokumentation und der Aufgabenstellung vertraut ist.

Zur Installation und Inbetriebnahme ist die Beachtung der Dokumentation, der nachfolgenden Hinweise und Erklärungen unbedingt notwendig. Das Fachpersonal ist verpflichtet, für jede Installation und Inbetriebnahme die zum betreffenden Zeitpunkt veröffentlichte Dokumentation zu verwenden.

Das Fachpersonal hat sicherzustellen, dass die Anwendung bzw. der Einsatz der beschriebenen Produkte alle Sicherheitsanforderungen, einschließlich sämtlicher anwendbarer Gesetze, Vorschriften, Bestimmungen und Normen erfüllt.

## Weiterführende Informationen

---

Unter den Links (DE)

<https://www.isg-stuttgart.de/produkte/softwareprodukte/isg-kernel/dokumente-und-downloads>

bzw. (EN)

<https://www.isg-stuttgart.de/en/products/softwareproducts/isg-kernel/documents-and-downloads>

finden Sie neben der aktuellen Dokumentation weiterführende Informationen zu Meldungen aus dem NC-Kern, Onlinehilfen, SPS-Bibliotheken, Tools usw.

## Haftungsausschluss

---

Änderungen der Software-Konfiguration, die über die dokumentierten Möglichkeiten hinausgehen, sind unzulässig.

## Marken und Patente

---

Der Name ISG®, ISG kernel®, ISG virtuos®, ISG dirigent® und entsprechende Logos sind eingetragene und lizenzierte Marken der ISG Industrielle Steuerungstechnik GmbH.

Die Verwendung anderer in dieser Dokumentation enthaltene Marken oder Kennzeichen durch Dritte kann zu einer Verletzung von Rechten der Inhaber der entsprechenden Bezeichnungen führen.

## Copyright

---

© ISG Industrielle Steuerungstechnik GmbH, Stuttgart, Deutschland.

Weitergabe sowie Vervielfältigung dieses Dokuments, Verwertung und Mitteilung seines Inhalts sind verboten, soweit nicht ausdrücklich gestattet. Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte für den Fall der Patent-, Gebrauchsmuster oder Geschmacksmustereintragung vorbehalten.

# Allgemeine- und Sicherheitshinweise

## Verwendete Symbole und ihre Bedeutung

In der vorliegenden Dokumentation werden die folgenden Symbole mit nebenstehendem Sicherheitshinweis und Text verwendet. Die (Sicherheits-) Hinweise sind aufmerksam zu lesen und unbedingt zu befolgen!

## Symbole im Erklärtext

- Gibt eine Aktion an.
- ⇒ Gibt eine Handlungsanweisung an.



### **GEFAHR**

#### **Akute Verletzungsgefahr!**

Wenn der Sicherheitshinweis neben diesem Symbol nicht beachtet wird, besteht unmittelbare Gefahr für Leben und Gesundheit von Personen!



### **VORSICHT**

#### **Schädigung von Personen und Maschinen!**

Wenn der Sicherheitshinweis neben diesem Symbol nicht beachtet wird, können Personen und Maschinen geschädigt werden!



### **Achtung**

#### **Einschränkung oder Fehler**

Dieses Symbol beschreibt Einschränkungen oder warnt vor Fehlern.



### **Hinweis**

#### **Tipps und weitere Hinweise**

Dieses Symbol kennzeichnet Informationen, die zum grundsätzlichen Verständnis beitragen oder zusätzliche Hinweise geben.



### **Beispiel**

#### **Allgemeines Beispiel**

Beispiel zu einem erklärten Sachverhalt.



### **Programmierbeispiel**

#### **NC-Programmierbeispiel**

Programmierbeispiel (komplettes NC-Programm oder Programmsequenz) der beschriebenen Funktionalität bzw. des entsprechenden NC-Befehls.



### **Versionshinweis**

#### **Spezifischer Versionshinweis**

Optionale, ggf. auch eingeschränkte Funktionalität. Die Verfügbarkeit dieser Funktionalität ist von der Konfiguration und dem Versionsumfang abhängig.

# Inhaltsverzeichnis

<b>Vorwort .....</b>	<b>2</b>
<b>Allgemeine- und Sicherheitshinweise .....</b>	<b>3</b>
<b>1 Zyklen-Einleitung .....</b>	<b>7</b>
1.1 Hinweise zur Installation der Zyklen.....	7
<b>2 Übersicht .....</b>	<b>9</b>
<b>3 Positionierungszyklen .....</b>	<b>10</b>
3.1 Einleitung .....	10
3.2 SysPosRotationToPlane - Auf Ebene Einschwenken .....	11
3.2.1 Ablauf .....	11
3.2.2 Definition der Ebene über Referenzpunkte .....	12
3.2.3 Definition der Ebene über Winkel .....	12
3.2.4 Parametrierung .....	13
3.2.5 Syntax .....	14
3.2.6 Programmierbeispiele .....	14
3.3 SysPosPattern - Aufrufmuster erstellen .....	17
3.3.1 Definition eines Aufrufmusters im NC-Programmcode .....	17
3.3.2 Ablauf .....	19
3.3.3 Parametrierung .....	19
3.3.4 Syntax .....	19
3.3.5 Programmierbeispiel .....	19
<b>4 Berechnungszyklen .....</b>	<b>21</b>
4.1 Einleitung .....	21
4.2 Zyklus Berechnung eines Kreises 2D .....	22
4.2.1 Ablauf .....	22
4.2.2 Parameter .....	22
4.2.3 Syntax .....	23
4.2.4 Ausgabevariablen .....	23
4.2.5 Programmierbeispiel .....	23
4.3 Zyklus Berechnung einer Ebene .....	24
4.3.1 Ablauf .....	24
4.3.2 Parameter .....	24
4.3.3 Syntax .....	25
4.3.4 Ausgabevariablen .....	25
4.3.5 Programmierbeispiel .....	26
4.4 Zyklus Berechnung einer Geraden 3D .....	27
4.4.1 Ablauf .....	27
4.4.2 Parameter .....	27
4.4.3 Syntax .....	28
4.4.4 Ausgabevariablen .....	28
4.4.5 Programmierbeispiel .....	29
4.5 Zyklus Berechnung einer Kugel .....	30
4.5.1 Ablauf .....	30
4.5.2 Parameter .....	30
4.5.3 Syntax .....	31

4.5.4	Ausgabevariablen .....	31
4.5.5	Programmierbeispiel .....	32
4.6	Zyklus Berechnung eines Kreises 3D .....	32
4.6.1	Ablauf .....	33
4.6.2	Parameter .....	33
4.6.3	Syntax .....	34
4.6.4	Ausgabevariablen .....	34
4.6.5	Programmierbeispiel .....	34
<b>5</b>	<b>High Speed Settings .....</b>	<b>36</b>
5.1	Einleitung .....	36
5.2	Zyklus SysHscSettings - High Speed Cutting Settings .....	36
5.2.1	Ablauf .....	36
5.2.2	Parametrierung .....	37
5.2.3	Syntax .....	37
5.2.4	Programmierbeispiel .....	37
<b>6</b>	<b>Anhang .....</b>	<b>40</b>
6.1	Anregungen, Korrekturen und neueste Dokumentation .....	40

## Abbildungsverzeichnis

Abb. 1:	SysPosRotationToPlane.....	11
Abb. 2:	Definition der Ebene .....	12
Abb. 3:	SysPosRotationToPlaneEx1 .....	14
Abb. 4:	SysPosRotationToPlaneEx2 .....	15
Abb. 5:	Aufrufmuster erstellen .....	17
Abb. 6:	Berechnung eines Kreises 2D .....	22
Abb. 7:	Berechnung einer Ebene .....	24
Abb. 8:	Berechnung einer Geraden 3D.....	27
Abb. 9:	Berechnung einer Kugel .....	30
Abb. 10:	Berechnung eines Kreises 3D .....	32
Abb. 11:	Wirkungsweise HSC Settings .....	36

# 1 Zyklen-Einleitung



## Hinweis

Zyklen sind eine lizenzpflichtige Zusatzoption.

## Allgemeine Informationen

### Zyklen-Aufruf

Der Aufruf der ISG-Zyklen erfolgt über einen Zyklusaufruf:

```
L CYCLE[ NAME="..." @P1 = .. @P2 = .. ...]
```

Durch die Vorgabe des Zyklennamens wird der entsprechende Zyklus ausgewählt. Zusätzlich ist es möglich, durch die Versorgungsparameter den Zyklus zu parametrieren und so das Verhalten des Zyklus an den speziellen Anwendungsfall anzupassen.

Die folgende Dokumentation enthält für jeden Zyklus ein eigenes Unterkapitel, in welchem das Verhalten des Zyklus genauer spezifiziert ist. Hier findet sich auch eine Liste der für den Zyklus verwendbaren Versorgungsparameter. Ein einfaches Programmierbeispiel beschreibt zusätzlich, wie der Zyklus aufgerufen wird.

### Zyklen-Ebenenwahl

Ein Zyklus ist unabhängig von der aktuell gültigen Ebene (G17, G18, G19) und unabhängig von den im NC-Kanal konfigurierten Achsnamen programmiert.

Zur verbesserten Lesbarkeit werden die Achsen in der Zyklendokumentation über folgende Namen beschrieben:

- Die X-Achse beschreibt die 1. Hauptachse
- Die Y-Achse beschreibt die 2. Hauptachse
- Die Z-Achse beschreibt die 3. Hauptachse
- A-Achse ist die Bezeichnung für die Rotationsachse um die 1. Hauptachse
- B-Achse ist die Bezeichnung für die Rotationsachse um die 2. Hauptachse
- C-Achse ist die Bezeichnung für die Rotationsachse um die 3. Hauptachse



## Hinweis

Zyklen können auch in verschobenen und gedrehten Koordinatensystemen verwendet werden. Die Definition dieser Koordinatensysteme sollte ausschließlich mit Hilfe des #CS-Befehls erfolgen. Der Befehl #ROTATION wird in Kombination mit Zyklen nicht unterstützt.

## 1.1 Hinweise zur Installation der Zyklen

### Inhalt des Zyklenpakets

Folgende Dateien und Verzeichnisse sind im Zyklenpaket enthalten:

- Setup-Datei zur Ausführung des Zyklen-Setups.
- Dependency-Graph: Beschreibt die Abhängigkeiten (Unterprogrammaufrufe) der einzelnen Zyklen.
- Releasenotes: Beschreibung relevanter Änderungen, welche im Release enthalten sind.
- Dokumentationsverzeichnis: Enthält die Dokumentation der Zyklen.
- Fehlerverzeichnis: Enthält die Zyklen-Fehlermeldungen.

- Komprimiertes Zyklenverzeichnis: Dieses enthält die CNC-Zyklen sowie Beispiel-Aufrufprogramme. Durch Ausführung des Setups werden die entsprechenden Dateien entpackt und am angegebenen Pfad hinterlegt.

## Ausführen des Setups

Zur Installation der Zyklen muss die im Zyklenpaket enthaltene Datei *isg-cnc-cycles-setup.exe* ausgeführt werden.

Durch das Ausführen des Zyklen-Setups werden die CNC-Zyklen sowie die Beispiel-Aufrufprogramme entpackt und am angegebenen Pfad hinterlegt.

Um die Zyklen ausführen zu können, muss der angegebene Pfad als Unterprogrammpfad in der Steuerung hinterlegt sein.



### Achtung

**Während der Ausführung des Zyklensetups wird der Inhalt eines vorher installierten Zyklenpakets überschrieben.**

## Eingliedern der Fehlermeldungen

Damit bei falscher Ausführung der CNC-Zyklen die korrekten Fehlermeldungen ausgegeben werden, müssen die Fehlerdateien in der TwinCAT-Installation hinterlegt werden.

Zur Ausgabe der Fehlermeldung in der HMI wird die Datei *TcCncCycleErrors.xml* aus dem Fehlerverzeichnis (*error\_files*) im TwinCAT-Installationsordner (hier *C:\TwinCat*) unter folgendem Pfad hinterlegt: *C:\TwinCAT\3.1\Target\Resource*.

Zur korrekten Ausgabe der Fehlermeldungen in der Log-Datei wird die Datei *err\_text\_cycles\_eng.txt* (bzw. *err\_text\_cycles.txt*) aus dem Fehlerverzeichnis (*error\_files*) im TwinCAT-Installationsordner (hier *C:\TwinCat*) unter folgendem Pfad hinterlegt: *C:\TwinCAT\3.1\Components\Mc\CNC\Diagnostics*.

Alternativ kann der Name der Fehlermeldungsdatei auch über den Kanalparameter P-STUP-00200 angepasst werden.



## 2 Übersicht



### Hinweis

Zyklen sind eine lizenzpflichtige Zusatzoption.

Die Zusatzzyklen können genutzt werden, sofern eine der drei Zyklen-Hauptlizenzen (Bearbeitung, Messen und Kalibrieren, Maschinenkalibrierung) vorhanden ist.

### Aufgabe

Zusätzlich zu den bestehenden Zyklen bieten die Zusatzzyklen folgenden Funktionsumfang:

1. Die Positionierungszyklen [► 10] dienen der Berechnung und Vorpositionierung des Werkzeugs. Sie werden üblicherweise im Verbund mit Bearbeitungs-, Mess- oder Kalibrierzyklen aufgerufen.
2. Die Berechnungszyklen [► 21] dienen der Durchführung komplexer Berechnungen, welche vom Anwender viel Fachwissen und Programmierarbeit erfordern.
3. Mit Hilfe des Zyklus für die High Speed Settings [► 36] kann das Maschinenverhalten über Kanalparameter allgemein eingestellt werden. Dadurch wird es möglich, maschinenspezifisch auf die Oberflächengüte, Geschwindigkeit und Genauigkeit Einfluss zu nehmen.

## 3 Positionierungszyklen

### 3.1 Einleitung



#### Hinweis

Zyklen sind eine lizenzpflichtige Zusatzoption.

#### Übersicht

Die Positionierungszyklen dienen der Berechnung und Vorpositionierung des Werkzeugs. Sie werden üblicherweise im Verbund mit Bearbeitungs-, Mess- oder Kalibrierzyklen aufgerufen.

#### Einsatzmöglichkeiten

Folgende Einsatzmöglichkeiten sind dabei denkbar:

- Berechnung einer Ebene durch Vorgabe von Referenzpunkten
- Einschwenken des Bearbeitungskoordinatensystems auf eine schräge Ebene
- Einschwenken des Werkzeugs auf eine schräge Ebene
- Verwendung von vordefinierten Aufrufmustern zur Zyklusausführung

#### Programmierung

Die Zyklen werden mit der Funktion `L CYCLE[...]` und den benötigten Parametern direkt aus dem NC-Programm aufgerufen. Eine genaue Beschreibung des Aufrufs wird in den Unterkapiteln der einzelnen Zyklen dargestellt.

## 3.2

### SysPosRotationToPlane - Auf Ebene Einschwenken

Mit diesem Zyklus wird das Werkzeug und/oder das Bearbeitungskoordinatensystem [BKS] auf eine über die Versorgungsparameter definierte Ebene eingeschwengt.

Die gewünschte Ebene wird durch Rotationswinkel oder Referenzpunkte definiert. Optional kann das resultierende Bearbeitungskoordinatensystem im Voraus durch eine Ausgabe überprüft werden. Dieser Zyklus kann nach einem Teach-In Zyklus verwendet werden, um auf das gemessene BKS einzuschwenken.

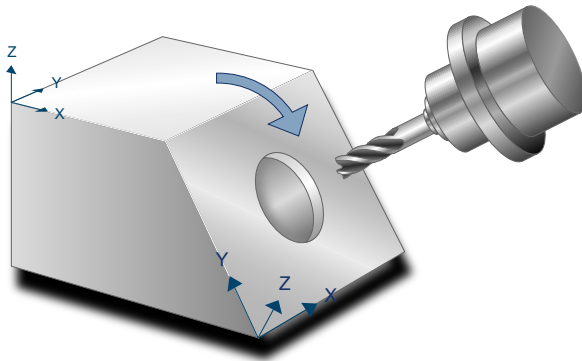


Abb. 1: SysPosRotationToPlane

### 3.2.1

#### Ablauf



#### Hinweis

Soll das Werkzeug auf die definierte Ebene eingeschwengt werden, so muss vor dem Zyklus eine vollständige kinematische Transformation aktiv sein.

Zu Beginn des Zyklus wird das Werkzeug um die über @P2 eingestellte Strecke entlang der Z-Achse zurückgezogen.

Über den Schwenkmodus @P1 können folgende Abläufe eingestellt werden:

- @P1=1: Das Bearbeitungskoordinatensystem und Werkzeug wird auf die definierte Ebene eingeschwengt
- @P1=2: Nur das Werkzeug wird auf die definierte Ebene eingeschwengt
- @P1=3: Nur das Bearbeitungskoordinatensystem wird auf die definierte Ebene eingeschwengt
- @P1=4: Es erfolgt keine Bewegung / Änderung des BKS. Dieser Modus kann genutzt werden, um das aus den Versorgungsparametern resultierende BKS zu berechnen und ausgeben zu lassen.

Wurde das Bearbeitungskoordinatensystem eingeschwengt (@P1=1 oder @P1=3), so ist nach dem Zyklus ein eingeschwenktes BKS mit Nullpunktverschiebung auf den über @P3@P5 definierten Ursprung aktiv. Dieses kann mit #CS OFF wieder abgewählt werden.

Zur Überprüfung der korrekten Definition der gewünschten Ebene kann das resultierende BKS mit @P20=1 als Textdatei ausgegeben werden. Dabei werden die berechneten Achsen des BKS, dargestellt im aktuellen WKS, in die Textdatei SysPosRotationToPlaneResult.txt geschrieben.

### 3.2.2

#### Definition der Ebene über Referenzpunkte

Die Ebene, auf welche eingeschungen werden soll, kann über 3 Punkte (@P6@P14) im aktuellen Werkstück-KS definiert werden. Werden diese Parameter übergeben, so wird die Zielebene damit definiert. Das resultierende Bearbeitungskoordinatensystem wird in diesem Fall so berechnet, dass alle 3 Punkte in der XY-Ebene liegen. Zusätzlich gilt:

- Die X-Achse des resultierenden BKS ist parallel zu dem Vektor vom 1. Referenzpunkt zum 2. Referenzpunkt
- Die Z-Achse liegt senkrecht zur XY-Ebene und wird so gelegt, dass ihr Wert in Z positiv ist
- Die Y-Achse wird so definiert, dass sie ein Rechtssystem mit der X- und Z-Achse bildet

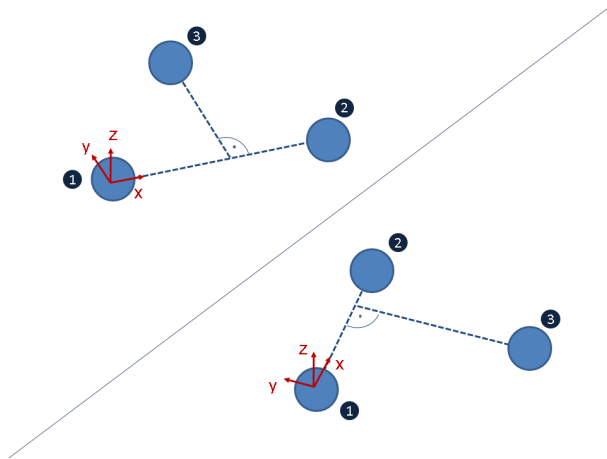


Abb. 2: Definition der Ebene

### 3.2.3

#### Definition der Ebene über Winkel

Die Ebene, auf die eingeschungen werden soll, kann zusätzlich über die Drehwinkel (@P15@P17) definiert werden. Werden diese Parameter übergeben, so wird die Zielebene damit definiert. Die Ebene wird dabei durch die Drehung um die Achsen des aktuellen WKS realisiert. Dabei ist eine Rotation mittels Projektionswinkel (Rotationsachsen drehen mit, @P18=1) und Raumwinkel (Rotationsachsen bleiben fest, @P18=2) möglich. Die Drehreihenfolge kann über @P19 definiert werden.

Weitere Informationen zum Drehen eines WKS um Koordinatensystem-Achsen finden sich in der Dokumentation zum #CS Befehl.

Soll der Schwenkzyklus mit einem Teach-in Zyklus zur Vermessung schiefer Ebenen kombiniert werden, so bietet sich die Definition der Ebene über Winkel an. Hier ist es ausreichend, die Ausgabevariablen des Teach-In Zyklus direkt zu übertragen: @P15=V.CYC.SYS\_A, @P16=V.CYC.SYS\_B, @P17=V.CYC.SYS\_C. Der Rotationsmodus und die Rotationsreihenfolge können auf den Standardwerten belassen werden.

### 3.2.4 Parametrierung

Folgende Versorgungsparameter sind für den Zyklusaufwurf erforderlich. Die Positionsangaben beziehen sich dabei auf das aktuell (vor dem Zyklusaufwurf) wirksame Werkstück-Koordinatensystem.

Versorgungsparameter	Beschreibung
@P1	Schwenkmodus 1: Bearbeitungskoordinatensystem und Werkzeug wird eingeschwenkt 2: Werkzeug wird eingeschwenkt 3: Bearbeitungskoordinatensystem wird eingeschwenkt 4: Kein Bewegung, lediglich Berechnung und Ausgabe (nur sinnvoll mit @P20=1)
@P2	Rückzugsstrecke (relativ)
@P3	Position in X des neuen Ursprungs
@P4	Position in Y des neuen Ursprungs
@P5	Position in Z des neuen Ursprungs
@P6	Position in X des ersten Referenzpunkt der Zielebene
@P7	Position in Y des ersten Referenzpunkt der Zielebene
@P8	Position in Z des ersten Referenzpunkt der Zielebene
@P9	Position in X des zweiten Referenzpunkt der Zielebene
@P10	Position in Y des zweiten Referenzpunkt der Zielebene
@P11	Position in Z des zweiten Referenzpunkt der Zielebene
@P12	Position in X des dritten Referenzpunkt der Zielebene
@P13	Position in Y des dritten Referenzpunkt der Zielebene
@P14	Position in Z des dritten Referenzpunkt der Zielebene
@P15	Drehwinkel für die Rotation um die X-Achse (Angabe in Grad)
@P16	Drehwinkel für die Rotation um die Y-Achse (Angabe in Grad)
@P17	Drehwinkel für die Rotation um die Z-Achse (Angabe in Grad)
@P18 (optional)	Rotationsmodus 1: Intrinsisch / Projektionswinkel (Standard) 2: Extrinsisch / Raumwinkel
@P19 (optional)	Drehreihenfolge (Tait-Bryan-Winkel) 1: XYZ 2: XZY 3: YXZ 4: YZX 5: ZXY 6: ZYX (Standard)
@P20 (optional)	Ergebnisausgabe 0: Keine Ausgabe (Standard) 1: Ergebnisausgabe über SysPosRotationToPlaneResult.txt

Zur Überprüfung der korrekten Belegung der Versorgungsparameter empfiehlt sich die Anwendung des Syntaxchecks.

### 3.2.5 Syntax

```
L CYCLE [ NAME = SysPosRotationToPlane.ecy @P.. = .. ]
```

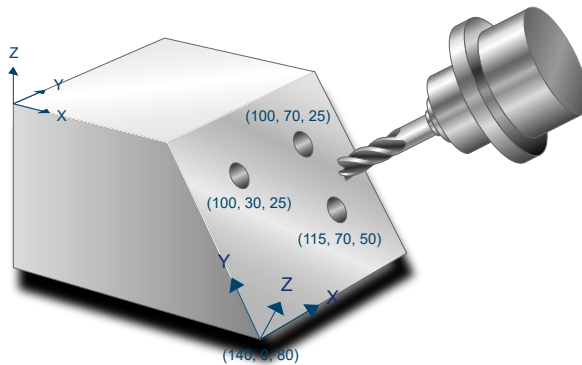
### 3.2.6 Programmierbeispiele

#### Beispiel 1 - Ebenendefinition durch Referenzpunkte



#### Programmierbeispiel

#### Ebenendefinition durch Referenzpunkte



**Abb. 3: SysPosRotationToPlaneEx1**

Folgende Parametrierung wird in diesem Beispiel angewendet:

- Werkzeug und Bearbeitungskoordinatensystem werden eingeschungen: @P1=1
- Die Rückzugsstrecke entlang der Z-Achse beträgt 50mm: @P2 = 50
- Der neue Ursprung liegt bei (140, 0, 80): @P3=140 @P4=0 @P5=80
- Der erste Referenzpunkt liegt bei (100, 30, 25): @P6=100 @P7=30 @P8=25
- Der zweite Referenzpunkt liegt bei (100, 70, 25): @P9=100 @P10=70 @P11=25
- Der dritte Referenzpunkt liegt bei (115, 70, 50): @P12=115 @P13=70 @P14=50

```
#TRAFO ON
```

```
L CYCLE [NAME=SysPosRotationToPlane.ecy \
    @P1 = 1 \
    @P2 = 50 \
    @P3 = 140 \
    @P4 = 0 \
    @P5 = 80 \
    @P6 = 100 \
    @P7 = 30 \
    @P8 = 25 \
    @P9 = 100 \
    @P10 = 70 \
    @P11 = 25 \
    @P12 = 115 \
```

```

        @P13 = 70
        @P14 = 50
    ]

(-----)
(---- Bearbeitung der Ebene ----)
(-----)

#CS OFF ; Deaktivierung des berechneten BKS

#TRAFO OFF

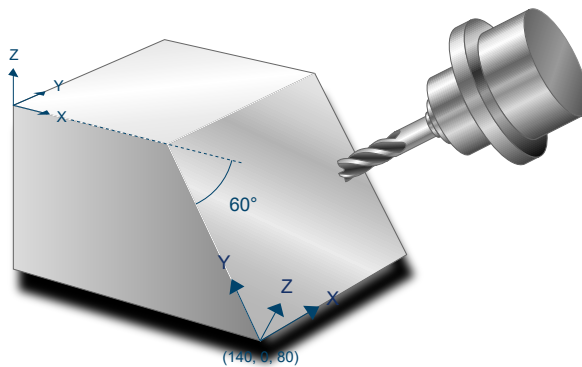
```

## Beispiel 2 - Ebenendefinition durch Drehwinkel



### Programmierbeispiel

#### Ebenendefinition durch Drehwinkel



**Abb. 4: SysPosRotationToPlaneEx2**

Folgende Parametrierung wird in diesem Beispiel angewendet:

- Werkzeug und Bearbeitungskoordinatensystem werden eingeschungen: @P1=1
- Die Rückzugsstrecke entlang der Z-Achse beträgt 50mm: @P2 = 50
- Der neue Ursprung liegt bei (140, 0, 80): @P3=140 @P4=0 @P5=80
- Die erste Drehung um die Z-Achse beträgt 90°: @P17=90
- Die zweite Drehung um die Y-Achse beträgt 0°: @P16=0
- Die dritte Drehung um die X-Achse beträgt -60°: @P15=-60

```

#TRAFO ON

L CYCLE [NAME=SysPosRotationToPlane.ecy \
    @P1 = 1
    @P2 = 50
    @P3 = 0
    @P4 = 0
    @P5 = 0
    @P15 = -60
    @P16 = 0
    @P17 = 90
]

(-----)
(---- Bearbeitung der Ebene ----)
(-----)

```

(-----)

#CS OFF ; Deaktivierung des berechneten BKS

#TRAFO OFF

M30



### 3.3 SysPosPattern - Aufrufmuster erstellen



#### Hinweis

Diese Funktionalität ist verfügbar ab CNC-Version V3.01.3079.37

Der Pattern-Zyklus beschreibt eine übergeordnete Funktionalität, die es ermöglicht, Zyklen mehrfach an vordefinierten Positionen auszuführen. Im Gegensatz zur vorhandenen Funktionalität MODAL\_MOVE / MODAL\_BLOCK können zudem Rotationen definiert werden, welche während der Zyklusausführung in Form eines Bearbeitungskoordinatensystems aktiv sind.

Folgende Einsatzmöglichkeiten sind im Zusammenhang mit dem Pattern-Zyklus denkbar:

- Ausführung der ISG-Zyklen an mehreren vordefinierten Positionen
- Ausführung eigener Zyklen an mehreren vordefinierten Positionen
- Definition von zusätzlichen Rotationen, welche bei der Zyklusausführung aktiv sind

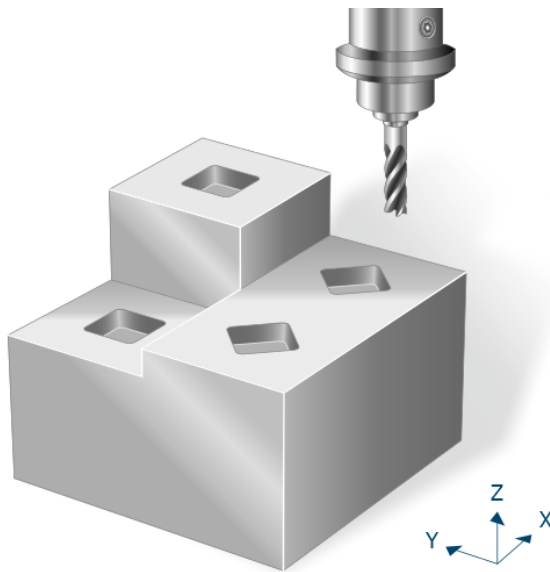


Abb. 5: Aufrufmuster erstellen

#### 3.3.1 Definition eines Aufrufmusters im NC-Programmcode

##### Zeitpunkt der Definition

Bei Hochlauf der Steuerung sind keine Aufrufmuster vordefiniert. Eine Definition in den Konfigurationslisten ist nicht möglich.

Aufrufmuster werden ausschließlich direkt im NC-Programm in einer von Klartextbefehlen eingeschlossenen Sequenz von Positions- und Rotationsdefinitionen erstellt. Die verwendeten Aufrufmuster müssen dabei vor Aufruf des entsprechenden Zyklus definiert werden. Die Definition des Aufrufmusters ist bis zum Überschreiben, Löschen oder bis zum Programmende gültig.

##### Beginn einer Aufrufmusterdefinition

```
# PATTERN BEGIN [ID<expr>]
```

ID <expr> Identifikationsnummer des Aufrufmusters.

Eine Aufrufmusterdefinition wird durch `#PATTERN BEGIN [ID <expr>]` aktiviert. Dabei wird die frei wählbare Identifikationsnummer übergeben. Ist bereits ein Aufrufmuster mit der gewünschten Identifikationsnummer vorhanden, wird dieses durch das neue Aufrufmuster überschrieben.

## Ende einer Aufrufmusterdefinition

```
# PATTERN END
```

Jede Aufrufmusterdefinition muss mit `#PATTERN END` abgeschlossen werden. Erst nach Abschluss der Aufrufmusterdefinition können wieder die Standardbefehle verwendet werden.

## Programmierbeispiel - Beschreibung eines Aufrufmusters

```
#PATTERN BEGIN [ID=1]
#AT [100, 0, 0]
#AT [100,100, 0,0,0,90]
#AT [0 ,100,100]
#PATTERN END
```

Jede Aufrufmusterdefinition beginnt mit `#PATTERN BEGIN` und muss mit `#PATTERN END` abgeschlossen sein. Zwischen diesen Befehlen werden die gewünschten Positionen und Rotationen beschrieben.

Mit dem Befehl `#AT` werden die Positionen und Rotationen definiert. Die ersten 3 Eingabewerte beschreiben die Position, von welcher der Zyklus ausgeführt werden soll. Die letzten 3 Eingabewerte beschreiben die Rotationen, welche bei der Ausführung des Zyklus aktiviert werden sollen. Die Winkeldefinition muss dabei gemäß der Drehung eines Bearbeitungskoordinatensystems gewählt werden. Aktuell ist hier nur eine Rotation um die Z-Achse erlaubt.

Werden keine Rotationen definiert, so ist es ausreichend, beim `#AT`-Befehl lediglich die ersten 3 Werte zu definieren.

## Aufrufmuster löschen

Der Befehl zum Löschen eines Aufrufmusters umfasst folgende Syntaxelemente:

```
# PATTERN DELETE [ID<expr>]
```

ID <expr> Identifikationsnummer des Aufrufmusters

Zudem ist es möglich, alle derzeit definierten Aufrufmuster zu löschen und den Speicherplatz für neue Definitionen frei zu geben:

```
# PATTERN DELETE ALL
```

Informationen von gelöschten Aufrufmustern sind unwiderruflich verloren. Der belegte Speicherplatz wird anschließend für neue Aufrufmusterdefinitionen freigegeben. Es können nur bereits definierte Aufrufmuster gelöscht werden.

## Programmierbeispiel - Aufrufmuster löschen

```
# PATTERN DELETE [ID3] ( Spezifisches Löschen des Aufrufmusters mit ID3)
# PATTERN DELETE ALL ( Löschen aller definierten Aufrufmuster)
```

### 3.3.2 Ablauf

Wurde ein Aufrufmuster definiert [► 17], kann der entsprechende Zyklus aufgerufen werden, um diesen an den vorgegebenen Positionen auszuführen. Dazu muss eine Aufrufmuster-Rückzugsebene `PATTERN_RETRACT` angegeben werden, auf welche das Werkzeug vor jeder neuen Positionierung zurückgezogen wird.

Durch Übergabe der Aufrufmuster-ID beim Zyklusaufruf wird intern der im Zykluspaket hinterlegte Zyklus "SysPosPatternWrapper.ecy" aufgerufen. In diesem wird für jede Positions- und Rotationsdefinition (`#AT`) nun auf der Aufrufmuster-Rückzugsebene die Position in X und Y angefahren. Danach werden die Rotationen sowie die übergebene Z-Position als Bearbeitungskoodinatensystem aktiviert, bevor der gewünschte Zyklus durchgeführt wird.

### 3.3.3 Parametrierung

Die Parametrierung erfolgt im Wesentlichen durch die Definition des Aufrufmusters [► 17].

Zusätzliche Parameter, die bei Zyklusaufruf übergeben werden müssen, sind:

- Die entsprechende Aufrufmuster-ID `PATTERN_ID`
- Die Aufrufmuster-Rückzugsebene `PATTERN_RETRACT` (absolute Angabe), auf welche zur Vorpositionierung zwischen den einzelnen Positionen zurückgezogen wird

Zur Überprüfung der korrekten Belegung der Versorgungsparameter empfiehlt sich die Anwendung des Syntaxchecks.

### 3.3.4 Syntax

```
; pattern data
# PATTERN BEGIN [ID=V.CYC.PatternID]
# AT [ ... ]
...
# PATTERN END

; cycle call
L CYCLE [NAME=... \
  @P.. = .. \
  PATTERN_ID = .. \
  PATTERN_RETRACT = .. \
]
```

### 3.3.5 Programmierbeispiel



#### Programmierbeispiel

##### Zyklusaufruf mit Aufrufmuster

Das folgende Beispiel beschreibt das Fräsen einer Rechtecktasche an drei verschiedenen Positionen mit Hilfe des Aufrufmuster-Aufrufs. Die zweite Aufrufmusterdefinition beinhaltet zudem eine Rotation, wodurch die Rechtecktasche an dieser Position um 60 Grad um die Z-Achse gedreht wird.

```
T8 D8          ( Tool data )
M6             ( Tool change )

G54 G90 S6000 M03 ( Technology data )

#VAR
```

```
; parameters for pattern:
V.CYC.PatternRetract      = 100
V.CYC.PatternID           = 1

; input parameters:
V.L.SurfacePosition       = 0      ( Z-Position of workpiece surface )
V.L.RetractionPlane       = 50     ( Z-Position of retraction plane )
V.L.SafetyClearance       = 2      ( relative value of safety clearance in Z)
V.L.DepthOfPocket        = 12     ( depth of pocket )
V.L.MaxIncrementZ         = 4      ( maximal infeed in Z )
V.L.MaxIncrementXY        = V.G.WZ_AKT.R ( maximal infeed in XY )
V.L.FeedRateXY            = 6000   ( machining feedrate in XY )
V.L.FeedRateZ             = 4000   ( plunging feedrate )
V.L.MachiningMode         = 1      ( machining mode )
V.L.PocketLength          = 20     ( length of the pocket )
V.L.PocketWidth           = 20     ( width of the pocket )
#ENDVAR

; pattern data
#PATTERN BEGIN [ID=V.CYC.PatternID]
#AT [100, 0, 0]
#AT [100, 100, 5, 0, 0, 60]
#AT [0 , 100, 10]
#PATTERN END

; polynomial contouring for smooth movements
#CONTOUR MODE [DEV, PATH_DEV = V.G.WZ_AKT.R / 100]
G261

G00 Z60
G00 X50 Y50                      ( Positioning to the starting point )

L CYCLE [NAME=SysMillRectangularPocket.ecy \
  @P1 = V.L.SurfacePosition \
  @P2 = V.L.RetractionPlane \
  @P3 = V.L.SafetyClearance \
  @P4 = V.L.DepthOfPocket \
  @P5 = V.L.MaxIncrementZ \
  @P6 = V.L.MaxIncrementXY \
  @P20 = V.L.FeedRateXY \
  @P21 = V.L.FeedRateZ \
  @P31 = V.L.MachiningMode \
  @P72 = V.L.PocketLength \
  @P73 = V.L.PocketWidth \
  PATTERN_ID = V.CYC.PatternID \
  PATTERN_RETRACT = V.CYC.PatternRetract \
]

G260
M05
M30
```

## 4 Berechnungszyklen

### 4.1 Einleitung



#### Hinweis

Zyklen sind eine lizenzpflichtige Zusatzoption.

#### Aufgabe

Berechnungszyklen dienen der Durchführung komplexer Berechnungen, welche vom Anwender viel Fachwissen und Programmierarbeit erfordern.

Folgende Berechnungszyklen stehen aktuell zur Verfügung:

- Berechnung des Durchmessers und Mittelpunkts eines zweidimensionalen Kreises aus mind. 3 Abtastpunkten
- Berechnung des Durchmessers und Mittelpunkts einer Kugel aus mind. 4 Abtastpunkten
- Berechnung einer Ebene aus mind. 3 Abtastpunkten
- Berechnung eines Kreises im 3D-Raum aus mind. 3 Abtastpunkten
- Berechnung einer Geraden im 3D-Raum aus mind. 2 Abtastpunkten

Die Berechnungszyklen verwenden die "Methode der kleinsten Quadrate", um aus einer gegebenen Anzahl an Abtastpunkten das wahrscheinlichste Ergebnis zu berechnen.

#### Programmierung und Parametrierung

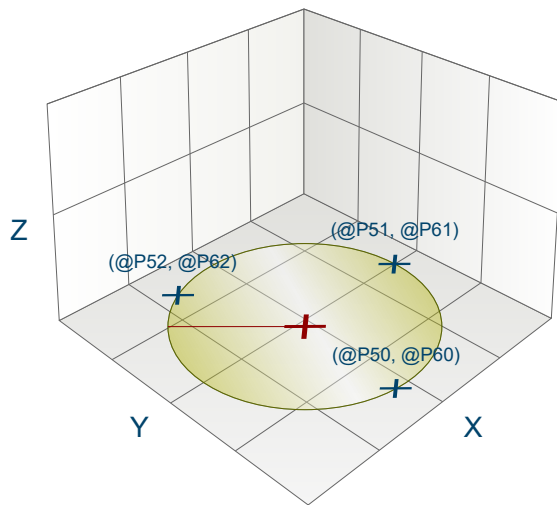
Die Programmierung und Parametrierung der Zyklen ist unter jedem der Hauptkapitel in der jeweiligen Übersicht beschrieben.

#### Handhabung von Ausgabevariablen

Wird die V.CYC-Ausgabevariable im Hauptprogramm (bzw. im zyklusaufrufenden Unterprogramm) erstellt, so wird der Wert innerhalb des Zyklus beschrieben und ist nach der Berechnung im Hauptprogramm verfügbar.

Die im folgenden beschriebenen Berechnungszyklen beinhalten Programmierbeispiel, mit welchen das Auslesen der Ausgabevariablen nachvollzogen werden kann.

## 4.2 Zyklus Berechnung eines Kreises 2D



**Abb. 6: Berechnung eines Kreises 2D**

### 4.2.1 Ablauf

Beim Berechnen eines Kreises in der XY-Ebene werden aus einer gegebenen Anzahl (mindestens 3, maximal 10) von Punkten auf dem Kreisbogen Radius und Mittelpunkt des Kreises bestimmt. Die Punkte dürfen nicht kollinear sein, sonst kann keine Berechnung durchgeführt werden!

Zum Erhalt guter Berechnungsergebnisse ist es empfehlenswert, möglichst weit von einander entfernte Punkte zu verwenden.

### 4.2.2 Parameter

Folgende Parameter sind für den Zyklusaufwurf erforderlich:

Zyklusparameter	Beschreibung
@P50	X-Koordinate Punkt 1
@P51	X-Koordinate Punkt 2
@P52	X-Koordinate Punkt 3
@P53 (optional)	X-Koordinate Punkt 4
@P54 (optional)	X-Koordinate Punkt 5
@P55 (optional)	X-Koordinate Punkt 6
@P56 (optional)	X-Koordinate Punkt 7
@P57 (optional)	X-Koordinate Punkt 8
@P58 (optional)	X-Koordinate Punkt 9
@P59 (optional)	X-Koordinate Punkt 10
@P60	Y-Koordinate Punkt 1
@P61	Y-Koordinate Punkt 2
@P62	Y-Koordinate Punkt 3

@P63 (optional)	Y-Koordinate Punkt 4
@P64 (optional)	Y-Koordinate Punkt 5
@P65 (optional)	Y-Koordinate Punkt 6
@P66 (optional)	Y-Koordinate Punkt 7
@P67 (optional)	Y-Koordinate Punkt 8
@P68 (optional)	Y-Koordinate Punkt 9
@P69 (optional)	Y-Koordinate Punkt 10

Zur Überprüfung der korrekten Belegung der Versorgungsparameter empfiehlt sich die Anwendung des Syntaxchecks.

### 4.2.3 Syntax

```
L CYCLE [ NAME = SysCalcCircle.ecy @P.. = .. ]
```

### 4.2.4 Ausgabevariablen

Variable	Wert
V.CYC.SysRetCenterX	Berechneter Mittelpunkt des Kreises in X
V.CYC.SysRetCenterY	Berechneter Mittelpunkt des Kreises in Y
V.CYC.SysRetRadius	Berechneter Radius des Kreises
V.CYC.SysRetVariance	Berechnete Varianz des Abstands der übergebenen Punkte zum berechneten Kreismittelpunkt

Siehe Informationen zur Verwendung der Ausgabevariablen [► 21].

### 4.2.5 Programmierbeispiel



#### Programmierbeispiel

#### Berechnungszyklus

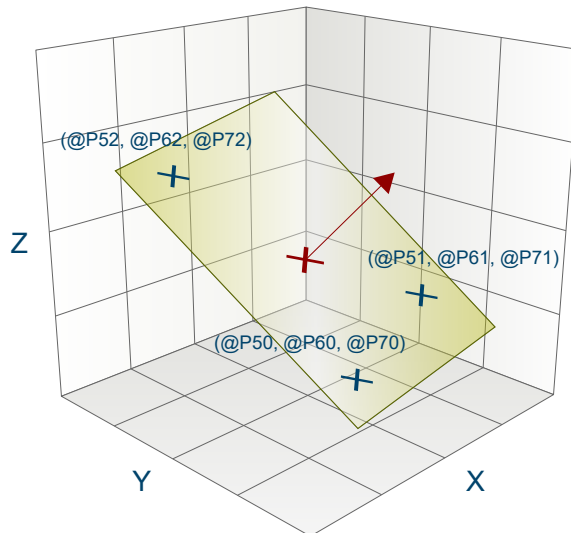
```
; Definition of return variables
#VAR
  V.CYC.SysRetRadius
  V.CYC.SysRetCenterX
  V.CYC.SysRetCenterY
  V.CYC.SysRetVariance
#ENDVAR

; cycle call
L CYCLE [NAME=SysCalcCircle.ecy \
@P50 = 0 @P60 = -1 \
@P51 = 0 @P61 = 1 \
@P52 = 1 @P62 = 0 \
@P53 = -1 @P63 = 0 ]
```

```
; print results
#FILE NAME [MSG="SysCalcCircleResult.txt"]
#MSG SAVE EXCLUSIVE["Center X = %f",V.CYC.SysRetCenterX]
#MSG SAVE EXCLUSIVE["Center Y = %f",V.CYC.SysRetCenterY]
#MSG SAVE EXCLUSIVE["Radius    = %f", V.CYC.SysRetRadius]
#MSG SAVE EXCLUSIVE["Variance = %f", V.CYC.SysRetVariance]
```

M30

## 4.3 Zyklus Berechnung einer Ebene



**Abb. 7: Berechnung einer Ebene**

### 4.3.1 Ablauf

Beim Berechnen einer Ebene im dreidimensionalen Raum werden aus einer gegebenen Anzahl (mindestens 3, maximal 10) von Punkten der Normalen- und Stützvektor einer Ebene bestimmt. Die Punkte dürfen nicht kollinear sein, sonst kann keine Berechnung durchgeführt werden!

Zum Erhalt guter Berechnungsergebnisse ist es empfehlenswert, möglichst weit von einander entfernte Punkte zu verwenden.

### 4.3.2 Parameter

Folgende Parameter sind für den Zyklusauf Ruf erforderlich:

Zyklusparameter	Beschreibung
@P50	X-Koordinate Punkt 1
@P51	X-Koordinate Punkt 2
@P52	X-Koordinate Punkt 3
@P53 (optional)	X-Koordinate Punkt 4
@P54 (optional)	X-Koordinate Punkt 5
@P55 (optional)	X-Koordinate Punkt 6
@P56 (optional)	X-Koordinate Punkt 7



@P57 (optional)	X-Koordinate Punkt 8
@P58 (optional)	X-Koordinate Punkt 9
@P59 (optional)	X-Koordinate Punkt 10
@P60	Y-Koordinate Punkt 1
@P61	Y-Koordinate Punkt 2
@P62	Y-Koordinate Punkt 3
@P63 (optional)	Y-Koordinate Punkt 4
@P64 (optional)	Y-Koordinate Punkt 5
@P65 (optional)	Y-Koordinate Punkt 6
@P66 (optional)	Y-Koordinate Punkt 7
@P67 (optional)	Y-Koordinate Punkt 8
@P68 (optional)	Y-Koordinate Punkt 9
@P69 (optional)	Y-Koordinate Punkt 10
@P70	Z-Koordinate Punkt 1
@P71	Z-Koordinate Punkt 2
@P72	Z-Koordinate Punkt 3
@P73 (optional)	Z-Koordinate Punkt 4
@P74 (optional)	Z-Koordinate Punkt 5
@P75 (optional)	Z-Koordinate Punkt 6
@P76 (optional)	Z-Koordinate Punkt 7
@P77 (optional)	Z-Koordinate Punkt 8
@P78 (optional)	Z-Koordinate Punkt 9
@P79 (optional)	Z-Koordinate Punkt 10

Zur Überprüfung der korrekten Belegung der Versorgungsparameter empfiehlt sich die Anwendung des Syntaxchecks.

### 4.3.3 Syntax

```
L CYCLE [ NAME = SysCalcPlane.ecy @P.. = .. ]
```

### 4.3.4 Ausgabevariablen

Variable	Wert
V.CYC.SysRetNormalX	Berechneter Normalenvektor in X
V.CYC.SysRetNormalY	Berechneter Normalenvektor in Y
V.CYC.SysRetNormalZ	Berechneter Normalenvektor in Z
V.CYC.SysRetSupVectorX	Berechneter Stützvektor der Ebene in X (entspricht Punkt auf der Ebene)

V.CYC.SysRetSupVectorY	Berechneter Stützvektor der Ebene in Y (entspricht Punkt auf der Ebene)
V.CYC.SysRetSupVectorZ	Berechneter Stützvektor der Ebene in Z (entspricht Punkt auf der Ebene)

Siehe Informationen zur Verwendung der Ausgabevariablen [► 21].

### 4.3.5 Programmierbeispiel



#### Programmierbeispiel

#### Berechnungszyklus

```

; creation of return variables
#VAR
    V.CYC.SysRetNormalX
    V.CYC.SysRetNormalY
    V.CYC.SysRetNormalZ
    V.CYC.SysRetSupVectorX
    V.CYC.SysRetSupVectorY
    V.CYC.SysRetSupVectorZ
#ENDVAR

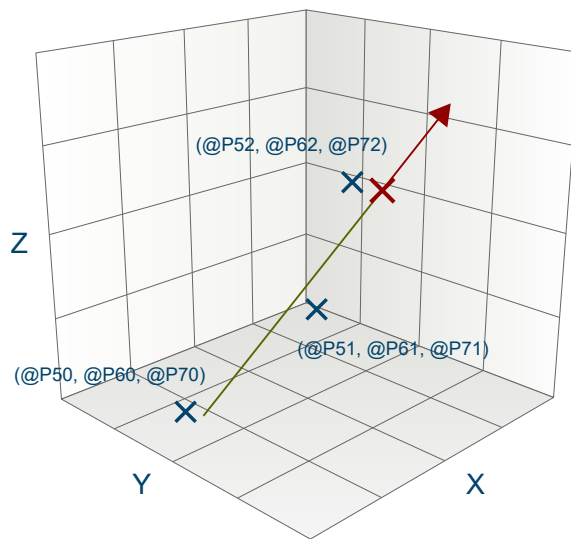
; calculation of the XY-plane
L CYCLE [NAME=SysCalcPlane.ecy \
@P50 = 1   @P60 = 0   @P70 = 0   \
@P51 = 0   @P61 = 1   @P71 = 0   \
@P52 = 1   @P62 = 1   @P72 = 0   ]

; print result
#FILE NAME[MSG="SysCalcPlaneResult.txt" ]
#MSG SAVE ["Normal X = %f", V.CYC.SysRetNormalX ]
#MSG SAVE ["Normal Y = %f", V.CYC.SysRetNormalY ]
#MSG SAVE ["Normal Z = %f", V.CYC.SysRetNormalZ ]
#MSG SAVE ["Point X  = %f", V.CYC.SysRetSupVectorX ]
#MSG SAVE ["Point Y  = %f", V.CYC.SysRetSupVectorY ]
#MSG SAVE ["Point Z  = %f", V.CYC.SysRetSupVectorZ ]

M30

```

## 4.4 Zyklus Berechnung einer Geraden 3D



**Abb. 8: Berechnung einer Geraden 3D**

### 4.4.1 Ablauf

Beim Berechnen einer Geraden im dreidimensionalen Raum werden aus einer gegebenen Anzahl (mindestens 2, maximal 10) von Punkten auf der Geraden Abstände der Punkte, sowie Richtungs- und Stützvektor bestimmt. Die Punkte dürfen nicht identisch sein, sonst kann keine Berechnung durchgeführt werden.

Zum Erhalt guter Berechnungsergebnisse ist es empfehlenswert, möglichst weit von einander entfernte Punkte zu verwenden.

### 4.4.2 Parameter

Folgende Parameter sind für den Zyklusaufwurf erforderlich:

Zyklusparameter	Beschreibung
@P50	X-Koordinate Punkt 1
@P51	X-Koordinate Punkt 2
@P52 (optional)	X-Koordinate Punkt 3
@P53 (optional)	X-Koordinate Punkt 4
@P54 (optional)	X-Koordinate Punkt 5
@P55 (optional)	X-Koordinate Punkt 6
@P56 (optional)	X-Koordinate Punkt 7
@P57 (optional)	X-Koordinate Punkt 8
@P58 (optional)	X-Koordinate Punkt 9
@P59 (optional)	X-Koordinate Punkt 10
@P60	Y-Koordinate Punkt 1
@P61	Y-Koordinate Punkt 2
@P62 (optional)	Y-Koordinate Punkt 3

@P63 (optional)	Y-Koordinate Punkt 4
@P64 (optional)	Y-Koordinate Punkt 5
@P65 (optional)	Y-Koordinate Punkt 6
@P66 (optional)	Y-Koordinate Punkt 7
@P67 (optional)	Y-Koordinate Punkt 8
@P68 (optional)	Y-Koordinate Punkt 9
@P69 (optional)	Y-Koordinate Punkt 10
@P70	Z-Koordinate Punkt 1
@P71	Z-Koordinate Punkt 2
@P72 (optional)	Z-Koordinate Punkt 3
@P73 (optional)	Z-Koordinate Punkt 4
@P74 (optional)	Z-Koordinate Punkt 5
@P75 (optional)	Z-Koordinate Punkt 6
@P76 (optional)	Z-Koordinate Punkt 7
@P77 (optional)	Z-Koordinate Punkt 8
@P78 (optional)	Z-Koordinate Punkt 9
@P79 (optional)	Z-Koordinate Punkt 10

Zur Überprüfung der korrekten Belegung der Versorgungsparameter empfiehlt sich die Anwendung des Syntaxchecks.

### 4.4.3 Syntax

**L CYCLE [ NAME = SysCalcLine.ecy @P.. = .. ]**

### 4.4.4 Ausgabevariablen

Variable	Wert
V.CYC.SysRetDirVectorX	Berechneter Richtungsvektor der Geraden in X
V.CYC.SysRetDirVectorY	Berechneter Richtungsvektor der Geraden in Y
V.CYC.SysRetDirVectorZ	Berechneter Richtungsvektor der Geraden in Z
V.CYC.SysRetSupVectorX	Berechneter Stützvektor der Geraden in X (entspricht Punkt auf der Geraden)
V.CYC.SysRetSupVectorY	Berechneter Stützvektor der Geraden in Y (entspricht Punkt auf der Geraden)
V.CYC.SysRetSupVectorZ	Berechneter Stützvektor der Geraden in Z (entspricht Punkt auf der Geraden)
V.CYC.SysRetVariance	Berechnete Varianz des Abstands der Punkte zur berechneten Geraden

Siehe Informationen zur Verwendung der Ausgabevariablen [► 21].

## 4.4.5

## Programmierbeispiel



## Programmierbeispiel

## Berechnungszyklus

```
; creation of return variables
#VAR
  V.CYC.SysRetDirVectorX
  V.CYC.SysRetDirVectorY
  V.CYC.SysRetDirVectorZ
  V.CYC.SysRetSupVectorX
  V.CYC.SysRetSupVectorY
  V.CYC.SysRetSupVectorZ
  V.CYC.SysRetVariance
#ENDVAR

; calc line of X-axis from points
L CYCLE [NAME=SysCalcLine.ecy \
@P50 = 0 @P60 = 0 @P70 = 0 \
@P51 = 1 @P61 = 0 @P71 = 0 \
@P52 = 2 @P62 = 0 @P72 = 0 \
]

#FILE NAME[MSG="SysCalcLineResult.txt" ]
#MSG SAVE ["Direction X      = %f", V.CYC.SysRetDirVector[0]]
#MSG SAVE ["Direction Y      = %f", V.CYC.SysRetDirVector[1]]
#MSG SAVE ["Direction Z      = %f", V.CYC.SysRetDirVector[2]]
#MSG SAVE ["Point on Line X = %f", V.CYC.SysRetSupVector[0]]
#MSG SAVE ["Point on Line Y = %f", V.CYC.SysRetSupVector[1]]
#MSG SAVE ["Point on Line Z = %f", V.CYC.SysRetSupVector[2]]
#MSG SAVE ["Variance         = %f", V.CYC.SysRetVariance ]

M30
```

## 4.5 Zyklus Berechnung einer Kugel

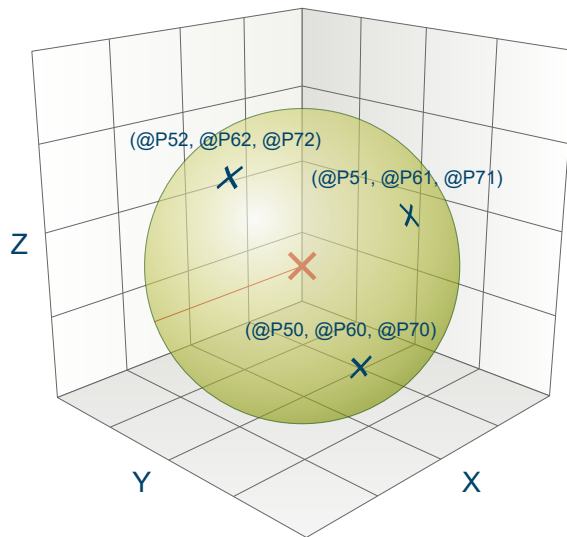


Abb. 9: Berechnung einer Kugel

### 4.5.1 Ablauf

Beim Berechnen einer Kugel im dreidimensionalen Raum werden aus einer gegebenen Anzahl (mindestens 4, maximal 10) von Punkten auf der Kugeloberfläche Radius und Mittelpunkt der Kugel bestimmt. Die Punkte dürfen nicht kollinear sein oder auf einer Ebene liegen, sonst kann keine Berechnung durchgeführt werden.

Zum Erhalt guter Berechnungsergebnisse ist es empfehlenswert, möglichst weit von einander entfernte Punkte zu verwenden.

### 4.5.2 Parameter

Folgende Parameter sind für den Zyklusaufwurf erforderlich:

Zyklusparameter	Beschreibung
@P50	X-Koordinate Punkt 1
@P51	X-Koordinate Punkt 2
@P52	X-Koordinate Punkt 3
@P53	X-Koordinate Punkt 4
@P54 (optional)	X-Koordinate Punkt 5
@P55 (optional)	X-Koordinate Punkt 6
@P56 (optional)	X-Koordinate Punkt 7
@P57 (optional)	X-Koordinate Punkt 8
@P58 (optional)	X-Koordinate Punkt 9
@P59 (optional)	X-Koordinate Punkt 10
@P60	Y-Koordinate Punkt 1
@P61	Y-Koordinate Punkt 2
@P62	Y-Koordinate Punkt 3

@P63	Y-Koordinate Punkt 4
@P64 (optional)	Y-Koordinate Punkt 5
@P65 (optional)	Y-Koordinate Punkt 6
@P66 (optional)	Y-Koordinate Punkt 7
@P67 (optional)	Y-Koordinate Punkt 8
@P68 (optional)	Y-Koordinate Punkt 9
@P69 (optional)	Y-Koordinate Punkt 10
@P70	Z-Koordinate Punkt 1
@P71	Z-Koordinate Punkt 2
@P72	Z-Koordinate Punkt 3
@P73	Z-Koordinate Punkt 4
@P74 (optional)	Z-Koordinate Punkt 5
@P75 (optional)	Z-Koordinate Punkt 6
@P76 (optional)	Z-Koordinate Punkt 7
@P77 (optional)	Z-Koordinate Punkt 8
@P78 (optional)	Z-Koordinate Punkt 9
@P79 (optional)	Z-Koordinate Punkt 10

Zur Überprüfung der korrekten Belegung der Versorgungsparameter empfiehlt sich die Anwendung des Syntaxchecks.

### 4.5.3 Syntax

**L CYCLE [ NAME = SysCalcSphere.ecy @P.. = .. ]**

### 4.5.4 Ausgabevariablen

Variable	Wert
V.CYC.SysRetCenterX	Berechneter Mittelpunkt der Kugel in X
V.CYC.SysRetCenterY	Berechneter Mittelpunkt der Kugel in Y
V.CYC.SysRetCenterZ	Berechneter Mittelpunkt der Kugel in Z
V.CYC.SysRetRadius	Berechneter Radius der Kugel
V.CYC.SysRetVariance	Berechnete Varianz des Abstands der übergebenen Punkte zur berechneten Kugel

Siehe Informationen zur Verwendung der Ausgabevariablen [► 21].

## 4.5.5

## Programmierbeispiel



### Programmierbeispiel

### Berechnungszyklus

```
; creation of return variables
#VAR
  V.CYC.SysRetCenterX
  V.CYC.SysRetCenterY
  V.CYC.SysRetCenterZ
  V.CYC.SysRetRadius
  V.CYC.SysRetVariance
#ENDVAR

; calculation of the unit sphere
L CYCLE [NAME=SysCalcSphere.ecy \
@P50 = 1 @P60 = 0 @P70 = 0 \
@P51 = -1 @P61 = 0 @P71 = 0 \
@P52 = 0 @P62 = 1 @P72 = 0 \
@P53 = 0 @P63 = -1 @P73 = 0 \
@P54 = 0 @P64 = 0 @P74 = 1 \
]

; print result
#FILE NAME[MSG="SysCalcSphereResult.txt"]
#MSG SAVE EXCLUSIVE["Center X = %f", V.CYC.SysRetCenterX ]
#MSG SAVE EXCLUSIVE["Center Y = %f", V.CYC.SysRetCenterY ]
#MSG SAVE EXCLUSIVE["Center Z = %f", V.CYC.SysRetCenterZ ]
#MSG SAVE EXCLUSIVE["Radius   = %f", V.CYC.SysRetRadius   ]
#MSG SAVE EXCLUSIVE["Variance = %f", V.CYC.SysRetVariance]

M30
```

## 4.6

## Zyklus Berechnung eines Kreises 3D

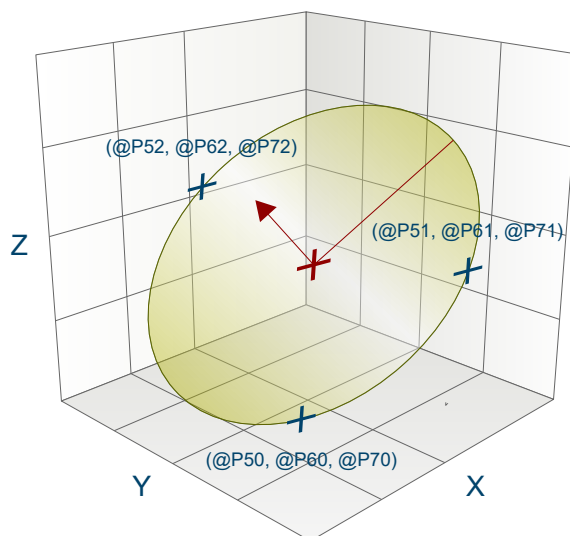


Abb. 10: Berechnung eines Kreises 3D



## 4.6.1 Ablauf

Beim Berechnen eines Kreises im dreidimensionalen Raum werden aus einer gegebenen Anzahl (mindestens 3, maximal 10) von Punkten auf dem Kreisbogen Radius, Mittelpunkt, Abstand zum Ursprung und Normalenvektor des Kreises bestimmt. Die Punkte dürfen nicht kollinear sein, sonst kann keine Berechnung durchgeführt werden!

Zum Erhalt guter Berechnungsergebnisse ist es empfehlenswert, möglichst weit von einander entfernte Punkte zu verwenden.

## 4.6.2 Parameter

Folgende Parameter sind für den Zyklusaufwurf erforderlich:

Zyklusparameter	Beschreibung
@P50	X-Koordinate Punkt 1
@P51	X-Koordinate Punkt 2
@P52	X-Koordinate Punkt 3
@P53 (optional)	X-Koordinate Punkt 4
@P54 (optional)	X-Koordinate Punkt 5
@P55 (optional)	X-Koordinate Punkt 6
@P56 (optional)	X-Koordinate Punkt 7
@P57 (optional)	X-Koordinate Punkt 8
@P58 (optional)	X-Koordinate Punkt 9
@P59 (optional)	X-Koordinate Punkt 10
@P60	Y-Koordinate Punkt 1
@P61	Y-Koordinate Punkt 2
@P62	Y-Koordinate Punkt 3
@P63 (optional)	Y-Koordinate Punkt 4
@P64 (optional)	Y-Koordinate Punkt 5
@P65 (optional)	Y-Koordinate Punkt 6
@P66 (optional)	Y-Koordinate Punkt 7
@P67 (optional)	Y-Koordinate Punkt 8
@P68 (optional)	Y-Koordinate Punkt 9
@P69 (optional)	Y-Koordinate Punkt 10
@P70	Z-Koordinate Punkt 1
@P71	Z-Koordinate Punkt 2
@P72	Z-Koordinate Punkt 3
@P73 (optional)	Z-Koordinate Punkt 4
@P74 (optional)	Z-Koordinate Punkt 5
@P75 (optional)	Z-Koordinate Punkt 6

@P76 (optional)	Z-Koordinate Punkt 7
@P77 (optional)	Z-Koordinate Punkt 8
@P78 (optional)	Z-Koordinate Punkt 9
@P79 (optional)	Z-Koordinate Punkt 10

Zur Überprüfung der korrekten Belegung der Versorgungsparameter empfiehlt sich die Anwendung des Syntaxchecks.

### 4.6.3 Syntax

```
L CYCLE [ NAME = SysCalcPlaneCircle.ecy @P.. = .. ]
```

### 4.6.4 Ausgabevariablen

Variable	Wert
V.CYC.SysRetCenterX	Berechneter Mittelpunkt des Kreises in X
V.CYC.SysRetCenterY	Berechneter Mittelpunkt des Kreises in Y
V.CYC.SysRetCenterZ	Berechneter Mittelpunkt des Kreises in Z
V.CYC.SysRetRadius	Berechneter Radius des Kreises
V.CYC.SysRetNormalX	Berechneter Normalenvektor in X
V.CYC.SysRetNormalY	Berechneter Normalenvektor in Y
V.CYC.SysRetNormalZ	Berechneter Normalenvektor in Z
V.CYC.SysRetRadVariance	Berechnete Varianz der Punkte zum Radius

Siehe Informationen zur Verwendung der Ausgabevariablen [► 21].

### 4.6.5 Programmierbeispiel



#### Programmierbeispiel

#### Berechnungszyklus

```
; creation of return variables
#VAR
  V.CYC.SysRetCenterX
  V.CYC.SysRetCenterY
  V.CYC.SysRetCenterZ
  V.CYC.SysRetNormalX
  V.CYC.SysRetNormalY
  V.CYC.SysRetNormalZ
  V.CYC.SysRetRadius
  V.CYC.SysRetRadVariance
#ENDVAR

; calculation of unit circle in the XY-plane with Z=1
L CYCLE [NAME=SysCalcPlaneCircle.ecy \
@P50 = 1 @P60 = 0 @P70 = 1 \
@P51 = 0 @P61 = 1 @P71 = 1 \
```

```
@P52 = -1  @P62 =  0  @P72 =  1      \  
@P53 =  0  @P63 = -1  @P73 =  1      ]  
  
; print result  
#FILE NAME[MSG="SysCalcPlaneCircleResult.txt"      ]  
#MSG SAVE ["Radius    = %f", V.CYC.SysRetRadius      ]  
#MSG SAVE ["Center X = %f", V.CYC.SysRetCenterX      ]  
#MSG SAVE ["Center Y = %f", V.CYC.SysRetCenterY      ]  
#MSG SAVE ["Center Z = %f", V.CYC.SysRetCenterZ      ]  
#MSG SAVE ["Normal X = %f", V.CYC.SysRetNormalX      ]  
#MSG SAVE ["Normal Y = %f", V.CYC.SysRetNormalY      ]  
#MSG SAVE ["Normal Z = %f", V.CYC.SysRetNormalZ      ]  
#MSG SAVE ["Variance = %f", V.CYC.SysRetRadVariance  ]
```

M30

## 5 High Speed Settings

### 5.1 Einleitung

Mit Hilfe des Zyklus für die High Speed Settings kann das Maschinenverhalten über Kanalparameter allgemein eingestellt werden. Dadurch wird es möglich, maschinenspezifisch auf die Oberflächengüte, Geschwindigkeit und Genauigkeit Einfluss zu nehmen.



#### Hinweis

Zyklen sind eine lizenzpflichtige Zusatzoption.

#### Aufgabe

Durch maschinenspezifische Vorgaben von Toleranzen für:

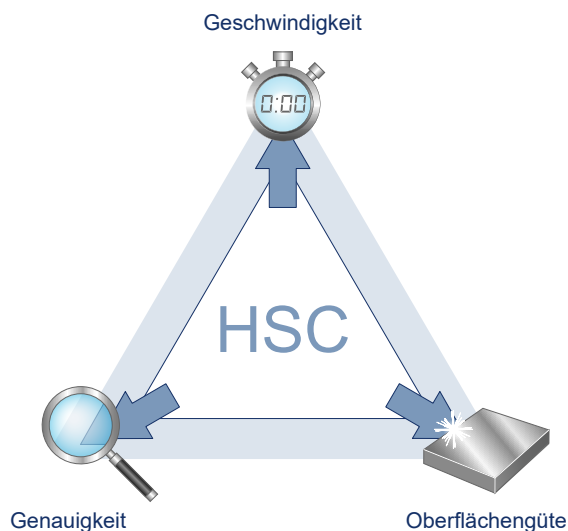
- das Schrappen
- das Vorschlichten
- das Schlichten

wird durch die High Speed Setting Zyklen das Bearbeitungsverhalten der Maschine eingestellt. Der Anwender kann durch den Zyklus den entsprechenden Bearbeitungszustand anwählen und so Einfluss auf das Maschinenverhalten nehmen.

#### Hinweis zur Lizenzierung

Bitte beachten Sie, dass die Zyklen lizenzpflichtige Zusatzoptionen sind.

### 5.2 Zyklus SysHscSettings - High Speed Cutting Settings



**Abb. 11: Wirkungsweise HSC Settings**

#### 5.2.1 Ablauf

Mit dem Zyklus SysHscSettings kann das Maschinenverhalten optimiert werden. In den Kanalspezifischen Listen können NC Programme angegeben werden, die in Abhängigkeit vom angegebenen Modus aufgerufen werden.

Die Parameter dafür heißen:

- hscs.deselect.prog
- hscs.deselect.tolerance
- hscs.rough.prog
- hscs.rough.tolerance
- hscs.prefinish.prog
- hscs.prefinish.tolerance
- hscs.finish.prog
- hscs.finish.tolerance

Die Werte der Listen Daten für die Toleranzen werden intern in mm oder inch umgerechnet.

Diese Toleranzen können im Nc-Programm über die folgenden Variablen gelesen werden:

```
V.G.HSCS.DESELECT.TOLERANCE
V.G.HSCS.ROUGH.TOLERANCE
V.G.HSCS.PREFINISH.TOLERANCE
V.G.HSCS.FINISH.TOLERANCE
```

## 5.2.2 Parametrierung

Folgende Versorgungsparameter sind für den Zyklusaufzuruf erforderlich:

Versorgungsparameter	Beschreibung
@P1	Modus Standardwert = 1 1 = Abwahl 2 = Grob (Schruppen) 3 = Mittel (Vorschlichten) 4 = Fein (Schlichten)

Zur Überprüfung der korrekten Belegung der Versorgungsparameter empfiehlt sich die Anwendung des Syntaxchecks.

## 5.2.3 Syntax

```
L CYCLE [ NAME = SysHscSettings.ecy @P.. = .. ]
```

## 5.2.4 Programmierbeispiel

### Allgemeines Programmierbeispiel



#### Programmierbeispiel

##### Kreistasche fräsen

In diesem Beispiel wird der Modus 4 aufgerufen. Dieser entspricht der Konfiguration Schlichten. Das hat zur Folge, dass das Unterprogramm manufacturerHscFinishSettings.nc, dass in dem Parameter hscs.finish.prg angegeben wurde, aufgerufen wird. Des weiteren ist die Bearbeitungstoleranz 0.01 mm auf der Variable V.G.HSCS.FINISH.TOLERANCE verfügbar und wird im Unterprogramm Beispielhaft für den #HSC Befehl verwendet.

```
; HSC Settings
T9 D9                                ( Tool data )
```

```
M6                                ( Tool change )
G00 G17 G90 F2000 M03 S6000      ( Technology data )
G00 Z150                         ( Go to z start position )
G00 X0 Y0                        ( position over the workpiece )

; Finishing Mode
L CYCLE [NAME=SysHscSettings.ecy \
      @P1 = 4                      \
      ]

; 3D Milling
;...
;...

M30
```

## Programmierbeispiel für die Optimierungs-Programme

Die angegebenen Daten sind Beispielhaft und können nicht einfach übernommen werden. Es können Schäden an der Maschine entstehen.

```
% manufacturerHscDeselectSettings.nc
G133 = 100
G134 = 100
#HSC OFF
#FILTER OFF
M17

% manufacturerHscRoughSettings.nc
; example
; G133 = 180 ; example G133 = 180 set 180%
; G134 = 180
; #SLOPE[TYPE=HSC]
; #HSC ON[SURFACE PATH_DEV=V.G.HSCS.ROUGH.TOLERANCE \
      TRACK_DEV=V.G.HSCS.ROUGH.TOLERANCE]
; #FILTER ON [AX_DEV=V.G.HSCS.ROUGH.TOLERANCE]
M17

% manufacturerHscPrefinishSettings.nc
; example
; G133 = 150 ; example G133 = 150 set 150%
; G134 = 150
; #SLOPE[TYPE=HSC]
; #HSC ON[SURFACE PATH_DEV=V.G.HSCS.PREFINISH.TOLERANCE \
      TRACK_DEV=V.G.HSCS.PREFINISH.TOLERANCE]
; #FILTER ON [AX_DEV=V.G.HSCS.PREFINISH.TOLERANCE]
M17

% manufacturerHscFinishSettings.nc
; example
; G133 = 80 ; example G133 = 80 set 80%
; G134 = 80
; #SLOPE[TYPE=HSC]
; #HSC ON[SURFACE PATH_DEV=V.G.HSCS.FINISH.TOLERANCE \
      TRACK_DEV=V.G.HSCS.FINISH.TOLERANCE]
; #FILTER ON [AX_DEV=V.G.HSCS.FINISH.TOLERANCE]
M17
```



## 6 Anhang

### 6.1 Anregungen, Korrekturen und neueste Dokumentation

Sie finden Fehler, haben Anregungen oder konstruktive Kritik? Gerne können Sie uns unter [documentation@isg-stuttgart.de](mailto:documentation@isg-stuttgart.de) kontaktieren. Die aktuellste Dokumentation finden Sie in unserer Onlinehilfe (DE/EN):



QR-Code Link: <https://www.isg-stuttgart.de/documentation-kernel/>

Der o.g. Link ist eine Weiterleitung zu:

<https://www.isg-stuttgart.de/fileadmin/kernel/kernel-html/index.html>



#### Hinweis

##### Mögliche Änderung von Favoritenlinks im Browser:

Technische Änderungen der Webseitenstruktur betreffend der Ordnerpfade oder ein Wechsel des HTML-Frameworks und damit der Linkstruktur können nie ausgeschlossen werden.

Wir empfehlen, den o.g. „QR-Code Link“ als primären Favoritenlink zu speichern.

#### PDFs zum Download:

DE:

<https://www.isg-stuttgart.de/produkte/softwareprodukte/isg-kernel/dokumente-und-downloads>

EN:

<https://www.isg-stuttgart.de/en/products/softwareproducts/isg-kernel/documents-and-downloads>

E-Mail: [documentation@isg-stuttgart.de](mailto:documentation@isg-stuttgart.de)





© Copyright  
ISG Industrielle Steuerungstechnik GmbH  
STEP, Gropiusplatz 10  
D-70563 Stuttgart  
Alle Rechte vorbehalten  
[www.isg-stuttgart.de](http://www.isg-stuttgart.de)  
[support@isg-stuttgart.de](mailto:support@isg-stuttgart.de)

