



Alle Rechte vorbehalten.
Die Firma Jetter AG behält sich das Recht vor, Änderungen an ihren Produkten vorzunehmen, die der technischen Weiterentwicklung dienen. Diese Änderungen werden nicht notwendigerweise in jedem Einzelfall dokumentiert.

Diese Application Note und die darin enthaltenen Informationen wurden mit der gebotenen Sorgfalt zusammengestellt. Die Firma Jetter AG übernimmt jedoch keine Gewähr für Druckfehler oder andere Fehler oder daraus entstehende Schäden.

Die hier genannten Marken und Produktnamen sind Warenzeichen oder eingetragene Warenzeichen der jeweiligen Titelhälter.

Jetter AG
Gräterstrasse 2
D-71642 Ludwigsburg
Germany

Telefon - Zentrale: 07141/2550-0
Telefon - Vertrieb: 07141/2550-530
Telefon - Technische Hotline: 07141/2550-444

Telefax: 07141/2550-425
E-Mail - Vertrieb: sales@jetter.de
E-Mail - Technische Hotline: hotline@jetter.de
Internetadresse: <http://www.jetter.de>

Inhaltsverzeichnis

1	Einleitung	5
1.1	Voraussetzungen	5
1.1.1	Systemvoraussetzungen	6
1.2	Einschränkungen	6
2	Programminitialisierung	6
2.1	Initialisierungsschritte und -Reihenfolge	6
2.2	Festlegen des Masterpositionsbereichs im Slave	6
2.3	Tabellendefinition	9
3	Produktionsperiode und Sägezyklus	9
3.1	Produktionsperiode	9
3.2	Sägezyklus	9
3.3	Status des Sägezyklus	10
3.4	Anzeige der Materiallänge	10
3.5	Anzeige der Distanz zum Schnittpunkt	10
3.6	Ausklinken	11
3.6.1	Ausklinken mit sofortiger Stopprampe	11
3.6.2	Ausklinken mit gleichbleibender Geschwindigkeit	12
3.6.3	Ausklinken mit Beschleunigung	12
3.7	Abbruch	13
4	Klassische fliegende Säge	14
4.1	Variante 1: Start durch Kopfschnitt	14
4.1.1	Hardwarkonfiguration	15
4.1.2	Beispielprogrammierung	15
4.2	Variante 2: Start mit Ausrichtung von Material und Werkzeugeinheit	16
4.2.1	Rahmenbedingungen	17
4.2.2	Hardwarekonfiguration	17
4.2.3	Beispielprogrammierung	17
5	Fliegende Säge mit Hardwaresignal	18
5.1	Variante 1: Direktes Starten nach Hardwaresignal, Fall 1	18
5.1.1	Rahmenbedingungen	19
5.1.2	Hardwarekonfiguration	20
5.1.3	Beispielprogrammierung	20
5.2	Variante 1: Direktes Starten nach Hardwaresignal, Fall 2	21
5.2.1	Rahmenbedingungen	21
5.2.2	Hardwarekonfiguration	21
5.2.3	Beispielprogrammierung	21
5.3	Variante 2: Auflaufen von mehreren Hardwaresignalen	22
5.3.1	Rahmenbedingungen	23
5.3.2	Hardwarekonfiguration	23
5.3.3	Beispielprogrammierung	23
6	Zusätzliche Anforderungen	24
6.1	Startposition der Sägeachse verändern	24
6.2	Schnitt ohne Rückfahrt	25
6.3	Unterschiedliche Geschwindigkeit zwischen Sägeachse und Motion-Master	26
6.4	Tabellenwechsel	26
7	Schnittgenauigkeit	27
7.1	Ungenauigkeit der Übersetzungsangaben	27
7.2	Schnittpunktkorrektur	27
7.3	Gleichlaufanpassung	28
7.4	Totzeit-Kompensation bei hoher Mastergeschwindigkeit	29

8	Fehlersuche	30
9	Funktionsbibliothek ‚Fliegende Säge‘	31
9.1	Was ist die Funktionsbibliothek „Fliegende Säge“ ?	31
9.2	Was ist zu beachten ?	31
9.3	Funktionen in JetSym-ST	31
9.4	Einbinden der Bibliotheksdateien	32
9.5	Typen	33
9.5.1	MOTION_FS_SLAVE_DATA	35
9.5.2	MOTION_FS_TAB_DATA	35
9.5.3	MOTION_FS_TMP_FLOATVARS	37
9.6	Konstanten	37
9.6.1	c_FS_STARTINSTANT_TIME_OF_PROCESS	39
9.7	Standard-Variablendeklaration für alle Beispiele	39
9.8	Funktionen	39
9.8.1	MotionFS_Ini	41
9.8.2	MotionFS_LoadTab_Linear / MotionFS_LoadTab_Sin2	43
9.8.3	MotionFS_Reset	47
9.8.4	MotionFS_Start	48
9.8.5	MotionFS_StartChangeTab	50
9.8.6	MotionFS_StartInstant	52
9.8.7	MotionFS_StartCapPos	53
9.8.8	MotionFS_GetStartStatus	55
9.8.9	MotionFS_GetActCutLength	57
9.8.10	MotionFS_GetDistToCutPoint	60
9.8.11	MotionFS_GetMasterAlignPos	62
9.8.12	MotionFS_SetSyncCorrection	64

1 Einleitung

Diese Applikation Note beschreibt die Programmierung eines JM-2xx für die Anwendung "Fliegende Säge". Dabei werden die JM-2xx als Singleachsen betrieben und nicht zusammen mit der Motion Control (MC). Zum besseren Verständnis dieser Applikation Note ist die Applikation Note Nr. 37 "Fliegende Säge – Allgemein" zuvor zu lesen.

Beschrieben werden in dieser Applikation Note die konkreten Schritte zur Umsetzung der verschiedenen Varianten einer fliegenden Säge, wie sie in der Applikation Note Nr. 37 beschrieben sind. Die Umsetzung ist in den folgenden Kapiteln zu finden:

- Klassische fliegende Säge, ab Kapitel 4 „Klassische fliegende Säge“
- Fliegende Säge mit Hardwaresignal, ab Kapitel 5 „Fliegende Säge mit Hardwaresignal“

Für zusätzliche Anforderungen an die Funktionalität gibt es in Kapitel 6 „Zusätzliche Anforderungen“ noch weitere Hinweise.

Die Programmierung der einzelnen Varianten wird jeweils mit Programmbeispielen verdeutlicht. Die verwendeten Programmbeispiele sind in dem JetSym-ST-Projekt „JM2xx_FS_ST_Example“ zusammengefasst. Das Projekt kann vom Downloadbereich der Homepage der Jetter AG heruntergeladen werden.

Zur Programmierung wird die JetSym-ST-Funktionsbibliothek für die fliegende Säge benötigt. Sie besteht aus den ST-Programmdateien LibJM2xxFS_h.stp und LibJM2xxFS.stp. Diese Dateien beinhalten Strukturdefinitionen und notwendige Funktionen zur Umsetzung der Anwendungen. Die Dateien können ebenfalls vom Downloadbereich der Homepage der Jetter AG heruntergeladen werden. Die Funktionsbibliothek ist ab Kapitel 9 „Funktionsbibliothek „Fliegende Säge““ im Dateil beschrieben.

Für alle Beispielprogrammteile, die in dieser Dokumentation aufgeführt sind, ist die Variablendeklaration im Kapitel 9.7 „Standard-Variablendeklaration für alle Beispiele“ auf Seite 39 maßgebend.

1.1 Voraussetzungen

Die fliegende Säge wird mit der Tabellenfunktion des JM-2xx realisiert. Die Funktionen der Bibliothek übernehmen das Handling der Tabellenfunktion, so dass der Anwender die Tabellenfunktion nur grob kennen muss. Dem Anwender muss lediglich bekannt sein, wie eine Tabelle im JM-2xx angelegt wird. Diese Informationen sind aus dem Dokument "jm2xx_an_jetcontrol_bi_xxxx_benutzerinformation.pdf" zu entnehmen.

Damit die Tabellenfunktion verwendet werden kann muss außerdem ein Achsverbund mit Motion-Master und Motion-Slave konfiguriert werden. Informationen zur Konfiguration eines Achsverbunds sind ebenfalls aus dem Dokument "jm2xx_an_jetcontrol_bi_xxxx_benutzerinformation.pdf" im Kapitel "Master-Slave-Betrieb" zu entnehmen. Zur Konfiguration eines Achsverbundes stehen im JetSym Motionbefehle zur Verfügung.

1.1.1 Systemvoraussetzungen

Folgende Software-Versionen werden mindestens benötigt:

Software-Versionen	
Steuerung / Module / Entwicklungsumgebung	SW-Version
JetSym	V3.0
JetMove 2xx	V2.07
JX2-CNT1	V3.0

1.2 Einschränkungen

Die Funktionen der Bibliothek sind für folgende Bedingungen ausgelegt:

- Der Motion-Master und die Sägeachse bewegen sich in positiver Richtung
- Die Sägeachse startet den Sägezyklus vom Stillstand aus
- Die Sägeachse führt eine optimale Synchronisierungsphase durch, d.h. die Sägeachse holt während der Synchronisierung keinen Weg auf, um den Schnittpunkt zu erreichen, sondern startet entsprechend frühzeitig. Bezüglich der zeitlich optimalen Synchronisierungsphase siehe Applikation Note Nr. 37.

2 Programminitialisierung

2.1 Initialisierungsschritte und -Reihenfolge

Zur Verwendung der Funktionsbibliothek müssen im Initialisierungsbereich des Steuerungsprogramms folgende Funktionen in der angegebenen Reihenfolge ausgeführt werden:

- Funktion MotionFS_Ini ausführen, zur Initialisierung der fliegenden Säge Struktur für alle vorhandenen Sägeachsen
- Funktionen MotionFS_LoadTab_Linear bzw. MotionFS_LoadTab_Sin2 ausführen, zum Berechnen und Laden der erforderlichen Tabellen
- Initialisieren der Achsverbunde im Master-Slave-Betrieb

Siehe Beispielprojekt im Task t_Init.

2.2 Festlegen des Masterpositionsbereichs im Slave

Bei der Initialisierung eines Achsverbunds muss auch der Masterpositionsbereich in der Sägeachse festgelegt werden. Der Masterpositionsbereich wird in der Sägeachse über die Achsregister R158 "Max. Master-Position" und R159 "Min. Master-Position" festgelegt. Die richtige Festlegung des Masterpositionsbereichs in der Sägeachse ist sehr wichtig für das korrekte Verhalten der Anwendung. Dabei muss folgendes beachtet werden:

Bei JX2-CNT1 als Motion-Master: Das Produkt aus Masterpositionsbereich (R158 – R159) x Normierungsfaktor Leitposition (R157) soll nicht größer als 500.000 sein.

Bei JetMove als Motion-Master: Das Ergebnis aus Verfahrbereich (R184 – R183) x Übersetzung, Motor (R194) x Übersetzung, Mechanik (195) des Motion-Masters soll nicht größer als 500.000 sein.

Es ist zu empfehlen, den Masterpositionsbereich automatisch über die Funktion *MotionFS_Ini* festlegen zu lassen. Dabei wird auch automatisch der Verfahrbereich im Motion-Master (falls der Motion-Master kein JX2-CNT1 ist) auf denselben Positionsbereich eingestellt. Der Positionsbereich wird dann wie in Abb. 1 dargestellt festgelegt.

ACHTUNG: Bei Verwendung eines Materialsensors für den Start des Sägezyklus muss beachtet werden, dass der Masterpositionsbereich mindestens so groß sein muss, wie die Distanz zwischen Ansprechposition des Materialsensors und Startposition der Sägeachse, Distanz = PosOffset.

D.h. der Parameter *AutoCalcMasterPosLimit* beim Aufruf der Funktion *MotionFS_Ini* muss gegebenenfalls auf 0 gesetzt sein, damit der Anwender selbst den Masterpositionsbereich, wie im vorhergehenden Satz beschrieben, manuell festlegen kann.

Die manuelle Festlegung des Masterpositionsbereichs ist bei einer fliegenden Säge mit mehreren auflaufenden Hardwaresignalen, siehe Kapitel 5.3 *Variante 2: Auflaufen von mehreren Hardwaresignalen*, zwingend notwendig. In diesem Fall ergibt die automatische Festlegung durch die Funktion *MotionFS_Ini* immer einen falschen Masterpositionsbereich.

Bei der automatischen Festlegung wird die angegebene größtmögliche Schnittlänge *MaxCuttingLength* als Grundlage verwendet und nicht die mechanischen Gegebenheiten des Motion-Master.

Beispiel:

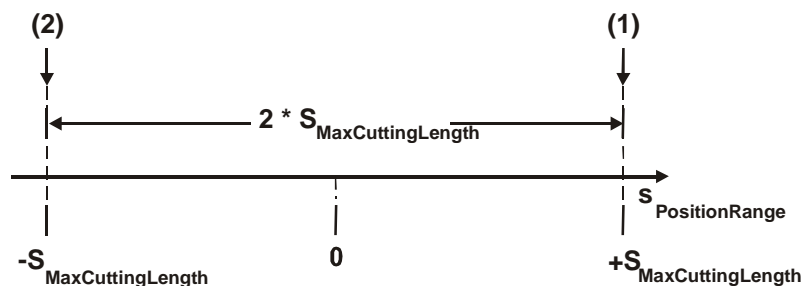
Die Motion-Masterachse ist ein Förderband mit einer Länge von 1000 mm. Die Schnittlänge beträgt 533 mm.

Wenn der Positionsbereich des Motion-Masters nach der Mechanik festgelegt werden soll, dann müssten folgende Positionsgrenzen im Motion-Master angegeben werden:

- 0 ... 1000 mm
- oder
- -500 ... 500 mm

Wenn der Positionsbereich des Motion-Masters automatisch über die Funktion *MotionFS_Ini* festgelegt wird, sind nach der Ausführung die Positionsgrenzen im Motion-Master und in der Sägeachse folgendermaßen eingestellt sein:

- -533 ... 533 mm



(*S_{PositionRange}*) Positionsbereich

(*S_{MaxCuttingLength}*) größtmögliche Schnittlänge *MaxCuttingLength*

(1) Verfahrgränze Positive (Motion-Master) / Maximale Masterposition (Sägeachse)

(2) Verfahrgränze Negativ (Motion-Master) / Minimale Masterposition (Sägeachse)

Abb. 1 : Automatische Festlegung des Masterpositionsbereichs

Durch die automatische Festlegung verliert der Masterpositionsbereich den absoluten Bezug zur Mechanik des Motion-Masters. Bei einer fliegenden Säge ist der absolute Bezug auch nicht zwingend erforderlich, weil die Schnitte relativ zueinander ausgeführt werden oder über ein Hardwaresignal angestoßen werden.

Wenn aus bestimmten Gründen doch der absolute Bezug zur Mechanik beibehalten werden soll, z.B. wenn zwischen den Sägezyklen auch absolute Positionierungen mit der Motion-Masterachse durchgeführt werden sollen, dann muss der Anwender den Positionsbereich vor dem Ausführen der Funktion MotionFS_Ini selbst über das Steuerungsprogramm festlegen. Dazu muss folgendes beachtet werden:

Es werden zwei Fälle unterschieden:

1. Fall: Motion-Master ist ein JM-2xx

In diesem Fall muss der Masterpositionsbereich in der Sägeachse genauso eingestellt werden, wie die negative und positive Verfahrengrenze im Motion-Master selbst, siehe Kapitel "Master-Slave-Betrieb" im Dokument "jm2xx_an_jetcontrol_bi_xxxx_benutzerinformation.pdf".

Außerdem muss die größtmögliche Schnittlänge mindestens zwei Mal in diesen Positionsbereich passen. Ist dies nicht der Fall, wird der Positionsbereich im Motion-Master und in der Sägeachse auf ein ganzzahliges Vielfaches des mechanischen Positionsbereichs festgelegt, so dass die Forderung wieder zutrifft. Überläufe des mechanischen Positionsbereichs müssen dann aber vom Anwender selbst im Steuerungsprogramm mitgezählt werden, um den absoluten Bezug nicht zu verlieren.

Beispiel: Die Masterachse ist ein Förderband mit einer Länge von 1000 mm. Die größtmögliche Schnittlänge beträgt 1200 mm.

Der Anwender muss den Positionsbereich im Motion-Master und in der Sägeachse dann folgendermaßen festlegen:

1. Minimalen Positionsbereich berechnen: $2 * \text{größtmögliche Schnittlänge} = 2400 \text{ mm}$
2. Festlegen, wie oft der mechanische Positionsbereich den minimalen Positionsbereich komplett abdeckt, ohne den mechanischen Positionsbereich zu teilen: $3 * 1000 \text{ mm} = 3000 \text{ mm}$

Positionsbereich:

- 0 ... 3000 mm
- oder
- -1500 ... 1500 mm



Wenn der Motion-Master ein JM-2xx ist, dann muss für den Masterpositionsbereich der Sägeachse exakt der gleiche Positionsbereich definiert werden, der im Motion-Master für den Verfahrensbereich definiert ist.

Wichtig

2. Fall: Master ist ein JX2-CNT1

In diesem Fall gibt es keine Angabe der negativen und positiven Verfahrengrenze im Motion-Master. Der Masterpositionsbereich kann nur in der Sägeachse festgelegt werden, siehe Kapitel "Master-Slave-Betrieb" im Dokument "jm2xx_an_jetcontrol_bi_xxxx_benutzerinformation.pdf".

Hier gilt aber auch wie im ersten Fall die Forderung, dass die größtmögliche Schnittlänge mindestens zwei Mal in diesen Positionsbereich passt. Ist dies nicht der Fall, muss auch hier der festzulegende Positionsbereich in der Sägeachse ein ganzzahliges Vielfaches des mechanischen Positionsbereichs des Motion-Masters sein, so dass die Forderung wieder zutrifft, siehe Beispiel oben vom 1. Fall.

2.3 Tabellendefinition

Für die Beschleunigungs- und Synchronbewegung der Sägeachse muss mindestens eine Tabelle erstellt werden. Die Tabelle wird mit den Funktionen MotionFS_LoadTab_Linear und MotionFS_LoadTab_Sin2 der Funktionsbibliothek erstellt, siehe 9.8.2 „MotionFS_LoadTab_Linear / MotionFS_LoadTab_Sin2“ auf Seite 43.

Es können mehrere Tabellen mit unterschiedlichen Beschleunigungen angelegt werden. Diese können während dem Betrieb gewechselt werden, siehe Kapitel 6.4 „Tabellenwechsel“ auf Seite 26.

3 Produktionsperiode und Sägezyklus

3.1 Produktionsperiode

Eine Produktionsperiode wird wie folgt definiert:

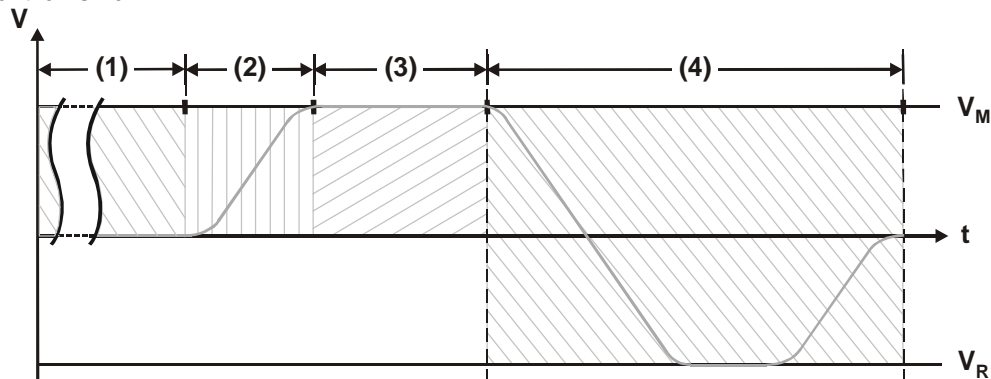
Produktionsperiode enthält mehrere bzw. viele Sägezyklen. Sie geht so lange, bis das aktuelle Produktformat beendet oder geändert wird oder z.B. eine Unterbrechung durch einen Fehler auftritt.

Zum Beginn einer Produktionsperiode muss eine Initialisierung mit der Funktion MotionFS_Reset stattfinden, siehe Kapitel 9.8.3 „MotionFS_Reset“ auf Seite 47.

3.2 Sägezyklus

Ein Sägezyklus wird wie folgt definiert:

Ein Sägezyklus führt einen Schnitt aus. Er besteht beim JM-2xx aus den Phasen, die in Abb. 2 ersichtlich sind:



- (1) 1. Phase: Wartephase: Warten bis Masterposition die automatisch ermittelte Startposition erreicht hat
- (2) 2. Phase: Synchronisierungsphase / Beschleunigungsphase
- (3) 3. Phase: Synchronphase / Sägephase
- (4) 4. Phase: Rückfahrt

Abb. 2 Sägezyklus

Die Sägeachse befindet sich von der 1. bis zur 3. Phase in der Tabellenfunktion, die über eine Startfunktionen (z.B. MotionFS_Start) automatisch gestartet wird. Für die 4. Phase muss der Anwender über das Steuerungsprogramm die Sägeachse aus der Tabelle ausklinken, siehe Kapitel 3.6 „Ausklinken“ auf Seite 11.



Wichtig

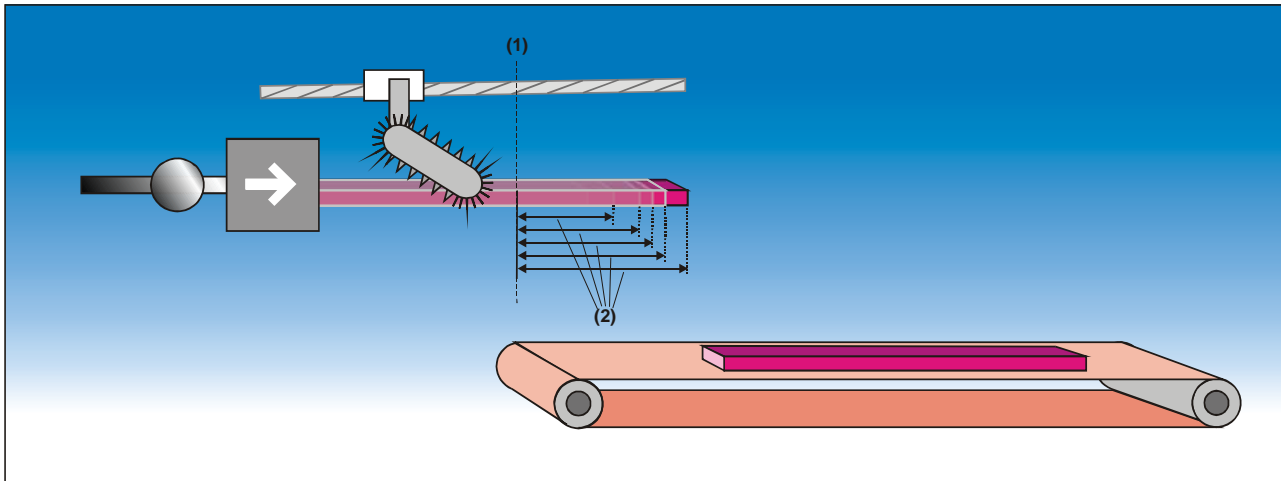
Die Sägeachse darf unter keinen Umständen zum Ende der Tabelle gelangen, da sie sonst abrupt anhält. Sie muss vor dem Ende der Tabelle entsprechend aus der Tabellenfunktion ausgeklinkt werden. Die Länge der Tabelle (Länge des Masterpositionsbereichs der Tabelle) ist identisch mit dem Wert des Parameters MaxCuttingLength, der beim Aufruf der Funktion MotionFS_Ini übergeben wird.

3.3 Status des Sägezyklus

In welcher Startphase sich die Sägeachse befindet (Phase 1-3), wird über die Funktion MotionFS_GetStartStatus abgefragt, siehe Kapitel 9.8.8 „MotionFS_GetStartStatus“ auf Seite 55.

3.4 Anzeige der Materiallänge

Bei einer fliegenden Säge kann es wünschenswert sein, die bereits aufgelaufene Materiallänge vor dem Schnitt anzuzeigen.



- (1) Synchronpunkt, bezogen auf Sägeachse
(2) Materiallänge an unterschiedlichen Zeitpunkten

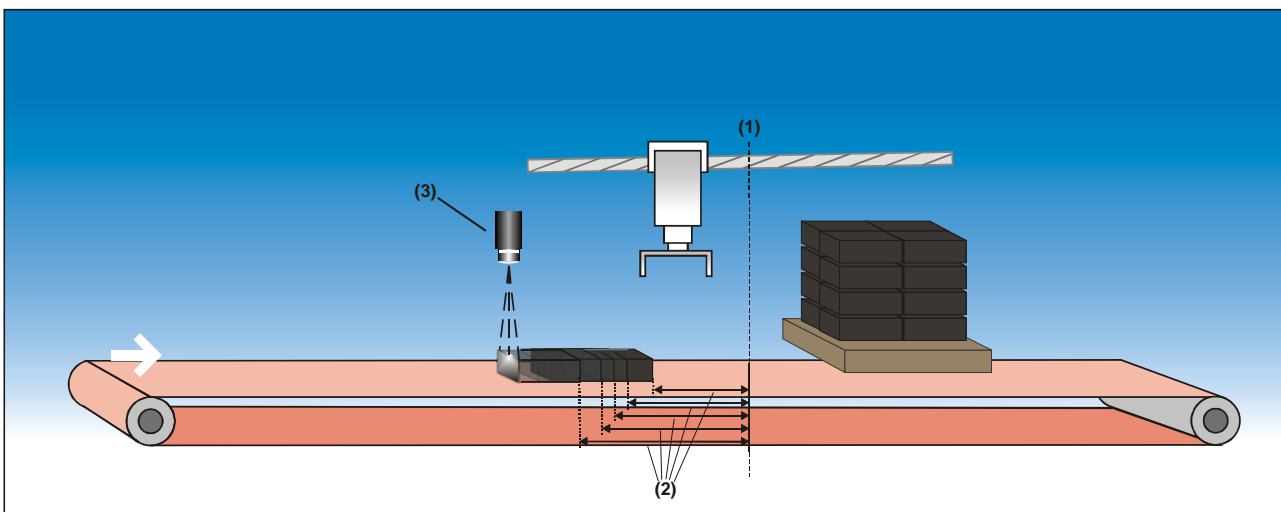
Abb. 3 Ermittlung der Materiallänge

Die Funktion MotionFS_GetActCutLength gibt die Materiallänge während dem Zyklusstart zurück, siehe Kapitel 9.8.9 „MotionFS_GetActCutLength“ auf Seite 57.

Die Materiallänge kann nur bei klassischen fliegenden Sägen ermittelt werden.

3.5 Anzeige der Distanz zum Schnittpunkt

Wenn die Anwendung eine fliegende Säge mit Hardwaresignal ist, dann kann die Distanz zum Schnittpunkt ermittelt werden. Die Funktion MotionFS_GetDistToCutPoint gibt die Distanz zurück, siehe Kapitel 9.8.10 „MotionFS_GetDistToCutPoint“ auf Seite 60.



- (1) Synchronpunkt, bezogen auf Sägeachse

- (2) Distanz zum Schnittpunkt an unterschiedlichen Zeitpunkten
- (3) Materialsensor

Abb. 4 Ermittlung der Distanz zum Schnittpunkt

3.6 Ausklinken

Nach dem Schnitt muss die Sägeachse vom Anwender aus der Tabellenfunktion ausgeklinkt werden. Dieses Unterkapitel gibt dazu Hilfestellung.

3.6.1 Ausklinken mit sofortiger Stopprampe

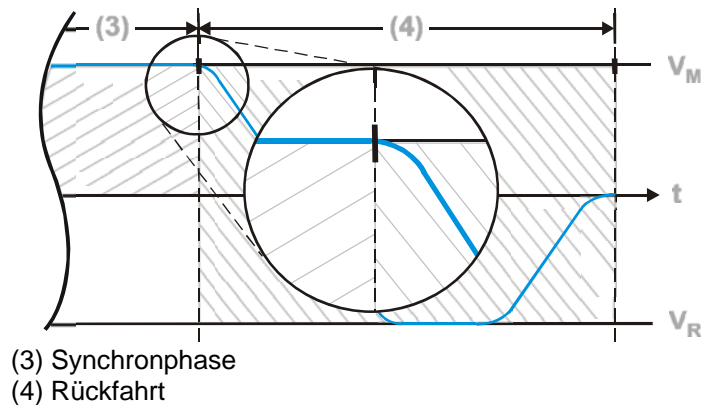


Abb. 5 Ausklinken mit sofortiger Stopprampe

Hier gibt es folgende Möglichkeiten:

1. Ausklinken direkt mit einer Rückfahrtpositionierung
2. Ausklinken mit Stopprampe und anschließender Rückfahrtpositionierung

Beispiel 1: Ausklinken direkt mit Rückfahrtpositionierung

```
...
//--- Rückfahrt: Positionierung starten während
// sich Sägeachse in Tabellenfunktion befindet
// ACHTUNG: Beim Umdrehen wird zum Abbremsen der Beschleunigungswert benutzt
MotionMovePtp(Ax, Alle, Absolute, 5000, 1500, <Beschleunigung>, <Verzögerung>,
               <Zielfenster>);
WHEN MotionReadStatus(Ax, Im Zielfenster) CONTINUE;
//--- Ab hier ist die Sägeachse bereits wieder an der Startposition
...
```

Beispiel 2: Ausklinken mit Stopprampe und anschließender Rückfahrtpositionierung

```
...
//--- Stopprampe mit benutzerdefinierter Rampe während
// sich Sägeachse in Tabellenfunktion befindet
MotionStop(Ax, Benutzerdefinierte Rampe, <Verzögerung>);
//--- Die Bits „Im Zielfenster“ und „Angehalten“ werden in diesem Fall
// von JM-2xx nicht bedient. Stattdessen muss das Bit „Stopprampe“
// abgefragt werden.
WHEN NOT MotionReadStatus(Ax, Stopprampe) CONTINUE;
//--- Rückfahrtpositionierung (siehe Beispiel 1)
...
```

3.6.2 Ausklinken mit gleichbleibender Geschwindigkeit

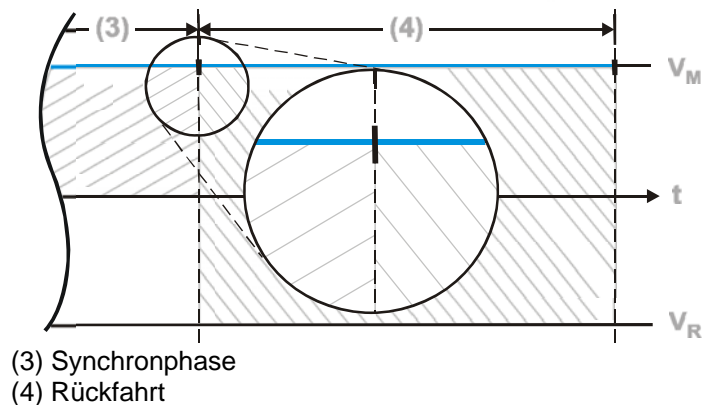


Abb. 6 Ausklinken mit gleichbleibender Geschwindigkeit

Beispiel

```
...
//--- Ausklinken und Geschwindigkeit beibehalten, während
// sich Sägeachse in Tabellenfunktion befindet
//--- Aktuelle Geschwindigkeit der Sägeachse ermitteln
f_IstGeschwindigkeit:= MotionReadValue(Ax, Istgeschwindigkeit, Mechanik);
//--- Aktuelle Position der Sägeachse ermitteln
f_IstPosition := MotionReadValue(Ax, Ist-Position);
//--- Mit der aktuellen Position eine entferntere Position berechnen
f_IstPosition := f_IstPosition + 5000;
//--- Positionierung ausführen
MotionMovePtp(Ax,Alle, Absolute, f_IstPosition, f_IstGeschwindigkeit, *, *, *);
WHEN MotionReadStatus(Ax, Im Zielfenster) CONTINUE;
//--- Rückfahrt
...
```

3.6.3 Ausklinken mit Beschleunigung

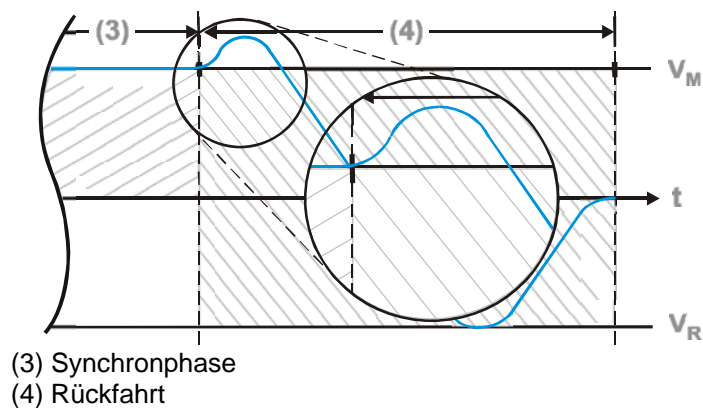
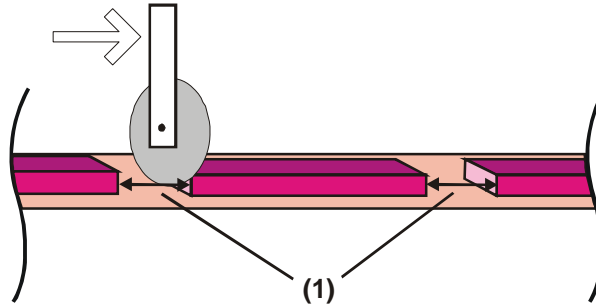


Abb. 7 Ausklinken mit Beschleunigung

Mit Beschleunigung auszuklinken kann z.B. sinnvoll sein, wenn die Werkzeuginheit der Sägeachse eine Lücke zwischen den geschnittenen Teilen ziehen soll.



(1) Lücke zwischen Produkten

Abb. 8 Lücke ziehen

Beispiel:

```
...
//--- Ausklinken mit Beschleunigung, während
// sich Sägeachse in Tabellenfunktion befindet
//--- Aktuelle Geschwindigkeit der Sägeachse ermitteln
    f_IstGeschwindigkeit:= MotionReadValue(Ax, Istgeschwindigkeit, Mechanik);
//--- Geschwindigkeit erhöhen
    f_IstGeschwindigkeit:= f_IstGeschwindigkeit + 500;
//--- Aktuelle Position der Sägeachse ermitteln
    f_IstPosition := MotionReadValue(Ax, Ist-Position);
//--- Auf die aktuelle Position eine entferntere Position berechnen
    f_IstPosition := f_IstPosition + 5000;
//--- Positionierung ausführen
    MotionMovePtp(Ax,Alle, Absolute, f_IstPosition, f_IstGeschwindigkeit,
        <Beschleunigung>, *, *);
    WHEN MotionReadStatus(Ax, Im Zielfenster) CONTINUE;
//--- Rückfahrt
...
```

3.7 Abbruch

Die Sägeachse kann während dem Sägezyklus jederzeit mit dem Motionbefehl MotionStop angehalten werden. Dabei macht es keinen Unterschied, ob sich die Sägeachse gerade in der Tabellenfunktion befindet oder nicht.

Beispiel 1: mit „Maximale Verzögerung“

```
MotionStop(Ax, Maximale Verzögerung);
//--- Die Bits „Im Zielfenster“ und „Angehalten“ werden
// von JM-2xx nicht bedient, wenn er sich in der Tabellenfunktion
// befindet. Stattdessen muss das Bit „Stopprampe“
// abgefragt werden.
WHEN NOT MotionReadStatus(Ax, Stopprampe) CONTINUE;
```

Beispiel 2: mit „Benutzerdefinierte Rampe“

```
MotionStop(Ax, Benutzerdefinierte Rampe, <Verzögerung>);
//--- Die Bits „Im Zielfenster“ und „Angehalten“ werden
// von JM-2xx nicht bedient, wenn er sich in der Tabellenfunktion
// befindet. Stattdessen muss das Bit „Stopprampe“
// abgefragt werden.
WHEN NOT MotionReadStatus(Ax, Stopprampe) CONTINUE;
```



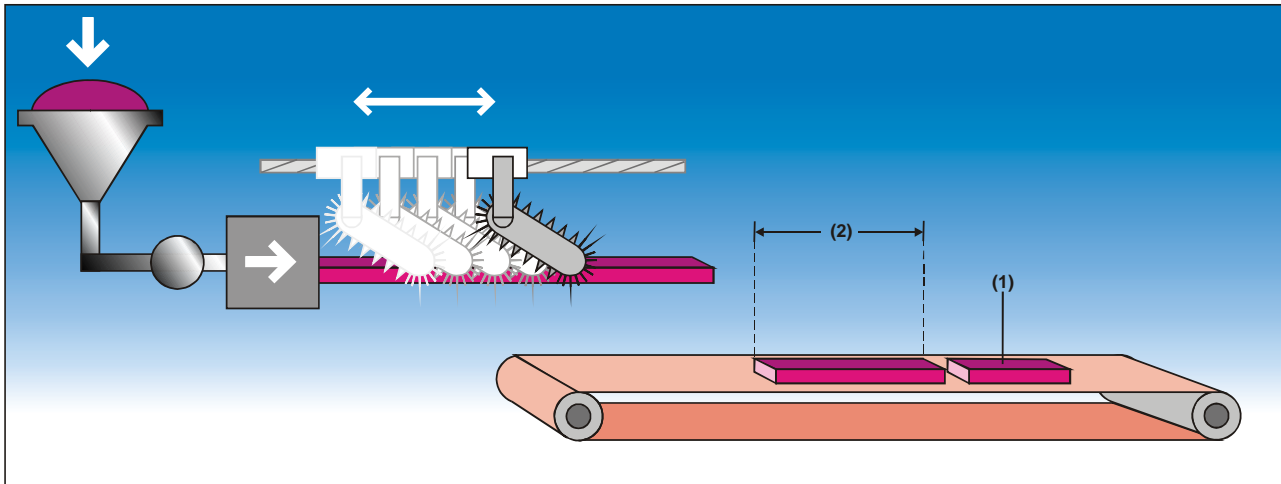
Wichtig

Wenn abgebrochen wird, während ein Task noch in einer der Startfunktionen (MotionFS_Start etc.) auf Synchronität wartet, dann muss dieser Task neu gestartet werden. Da beim Abbruch die Synchronität nie erreicht wird, kann sich die Startfunktion nicht beenden, siehe Beispielprojekt im Task t_Emerg.

4 Klassische fliegende Säge

4.1 Variante 1: Start durch Kopfschnitt

Der erste Schnitt ist ein sogenannter Kopfschnitt. Das ist ein Schnitt, der ohne jegliche Ausrichtung der Werkzeugeinheit zum Material sofort ausgeführt wird und das Material irgendwo schneidet. Das geschnittene Anfangsstück (Kopfschnitt) ist unbrauchbar. Durch diesen Kopfschnitt sind dann die nachfolgenden Schnitte zur Werkzeugeinheit ausgerichtet.



- (1) Kopfschnitt
- (2) feste Schnittlänge

Abb. 8: Klassische fliegende Säge mit Kopfschnitt

4.1.1 Hardwarkonfiguration

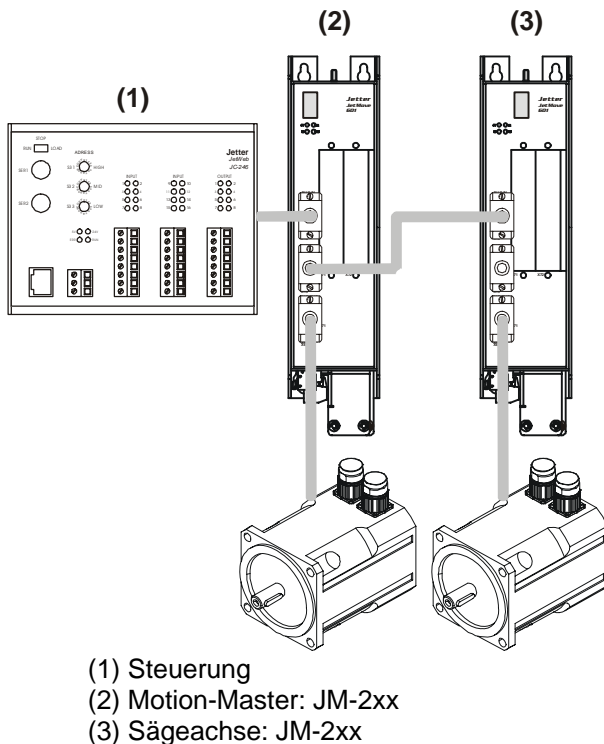


Abb. 9 Hardwarekonfiguration mit JM-2xx

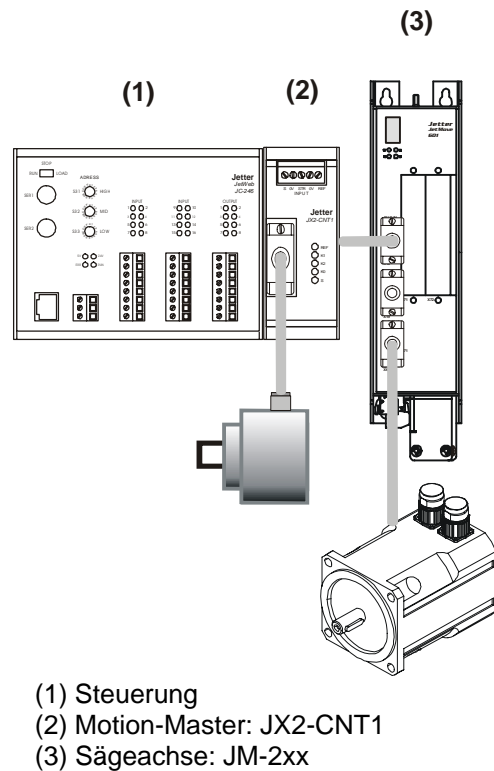


Abb. 10 Hardwarekonfiguration mit JX2-CNT1

4.1.2 Beispielprogrammierung

Siehe auch Prozedur „su_ClassicalHeadCut“ im Beispielprojekt.

```
//--- Produktionsperiode
...
//--- Fahre Sägeachse zur Startposition
MotionMovePtp (Ax, Alle, Absolut, <SägeachseStartposition>,...)
WHEN MotionReadStatus (Ax, Im Zielfenster) CONTINUE;

//--- Kopfschnitt
MotionFS_StartInstant(&st_MotionFS_Slave0Data, &st_MotionFS_Slave0TabData0, 0,
&st_MotionFS_TmpFloatVars0);

//--- Schnitt ausführen
...

//--- Sägezyklus
WHILE <Zyklusmerker> DO //--- Solange Zyklusmerker gesetzt
//--- Fahre Sägeachse zur Startposition
MotionMovePtp (Ax, Alle, Absolut, <SägeachseStartposition>,...)
WHEN MotionReadStatus (Ax, Im Zielfenster) CONTINUE;

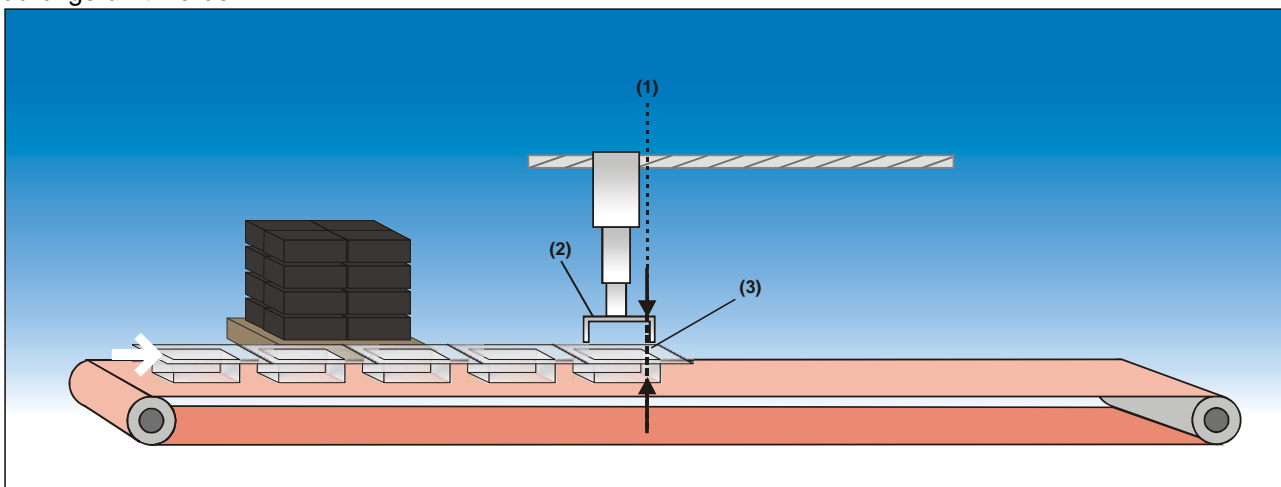
//--- Starte Sägeachse
n_Return:= MotionFS_Start(&st_MotionFS_Slave0Data,
&st_MotionFS_Slave0TabData0, &f_CuttingLength, 0,
&st_MotionFS_TmpFloatVars0);

CASE n_Return OF //--- Auswertung des Rückgabewerts
c_FS_START_STAT_ERR_MASTER_POS_TOO_FAR:
```

```
DISPLAY_TEXT (<Zeile>, <Spalte>, 'FEHLER: Master zu weit$');  
b_AutoCycle:= FALSE;  
RETURN;  
BREAK;  
END_CASE;  
//--- Schnitt ausführen  
...  
END_WHILE  
...
```

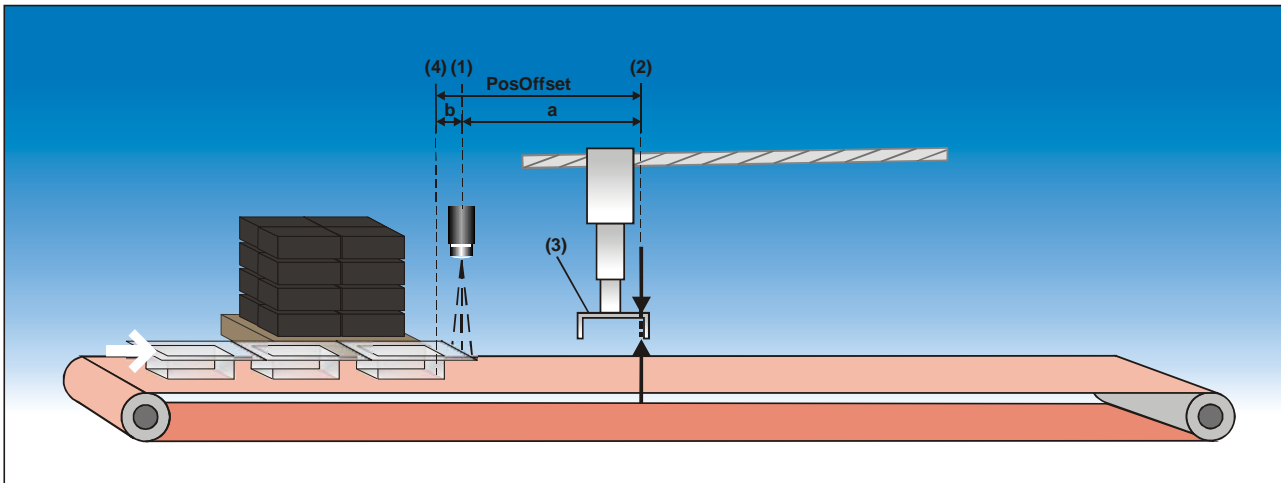
4.2 Variante 2: Start mit Ausrichtung von Material und Werkzeugeinheit

Vor dem ersten Schnitt werden Material und Werkzeugeinheit zueinander ausgerichtet. Somit wird der Schnittpunkt vom ersten Schnitt an auf dem Material genau festgelegt. Die Ausrichtung kann manuell, z.B. über einen Tippbetrieb, durchgeführt werden. Mit Hilfe eines Materialsensors kann sie auch automatisch durchgeführt werden.



- (1) Startposition der Sägeachse
- (2) Werkzeugeinheit
- (3) Ausrichtungspunkt des Materials befindet sich an der Startposition der Sägeachse

Abb. 11 Manuelle Ausrichtung von Material und Werkzeugeinheit



- (1) Ansprechposition des Materialsensors
 (2) Startposition der Sägeachse
 (3) Werkzeugeneinheit
 (4) Ausrichtungspunkt am Material
 (a) Distanz zwischen Materialsensor und Startposition der Sägeachse
 (b) Distanz zwischen Materialerkennungskante und Ausrichtungspunkt am Material
 (PosOffset) Übergabeparameter PosOffset für Funktion MotionFS_StartPosCap

Abb. 12 Automatische Ausrichtung von Material und Werkzeugeneinheit

4.2.1 Rahmenbedingungen

Automatische Ausrichtung:

- Der Materialsensor muss mit entsprechendem Abstand VOR der Werkzeugeneinheit positioniert werden, siehe Kapitel 9.8.7 „MotionFS_StartCapPos“ auf Seite 53.
- Die Distanz zwischen Ansprechposition des Materialsensors und der Startposition der Sägeachse, PosOffset, darf nicht größer sein, als der maximale Positionsbereich des Motion-Masters.

4.2.2 Hardwarekonfiguration

Manuelle Ausrichtung:

Wie oben, bei Variante 1.

Automatische Ausrichtung:

Wie unten, bei Variante 1 der fliegenden Säge mit Hardwaresignal.

4.2.3 Beispielprogrammierung

Manuelle Ausrichtung:

```
VAR
    ...
    f_AlignPos:                FLOAT    AT %VL  65080;  //--- Zum Speichern der
                                                //      Ausrichtposition
    ...
VAR_END;

//--- Produktionsperiode
...
//--- Position des Motion-Master auf Ausrichtungspunkt setzen.
//      Zuvor ist das Material und die Werkzeugeneinheit an der Startposition der
//      Sägeachse manuell ausgerichtet worden.
```

```
MotionFS_GetMasterAlignPos(&st_MotionFS_Slave0Data,&st_MotionFS_Slave0TabData0,
                           0,&f_AlignPos,&st_MotionFS_TmpFloatVars0);
MotionHome ( Ax, Grundstellung setzen, f_AlignPos);

//--- Sägezyklus
siehe oben, bei Variante 1...
```

Automatische Ausrichtung:

Die Beispielapplikation ist in Abb. 12 dargestellt. Die Distanz zwischen Ausrichtungspunkt am Material und Startposition der Sägeachse (a + b) beträgt 90 mm + 500 mm = 590 mm. Siehe auch Prozedur „su_ClassicalAutoAlign“ im Beispielprojekt.

```
VAR
...
f_CapturePos:          FLOAT   AT %VL  65081;  //--- Captureposition
...
VAR_END;

//--- Produktionsperiode
...
//--- Fahre Sägeachse zur Startposition
MotionMovePtp (Ax, Alle, Absolut, <SägeachseStartposition>,...)
WHEN MotionReadStatus (Ax, Im Zielfenster) CONTINUE;

//--- Einschalten der Capturfunktion für die Motion-Master-Position und warten
//   bis Captureereignis eingetreten ist,
//   und speichere Captureposition in f_CapturePos
...(siehe im Beispielprojekt)
f_CapturePos:= ...

//--- Starte ersten Sägezyklus nach dem Captureereignis
MotionFS_StartCapPos(&st_MotionFS_Slave0Data,&st_MotionFS_Slave0TabData0,
                    &f_CapturePos,590,0,&st_MotionFS_TmpFloatVars0);

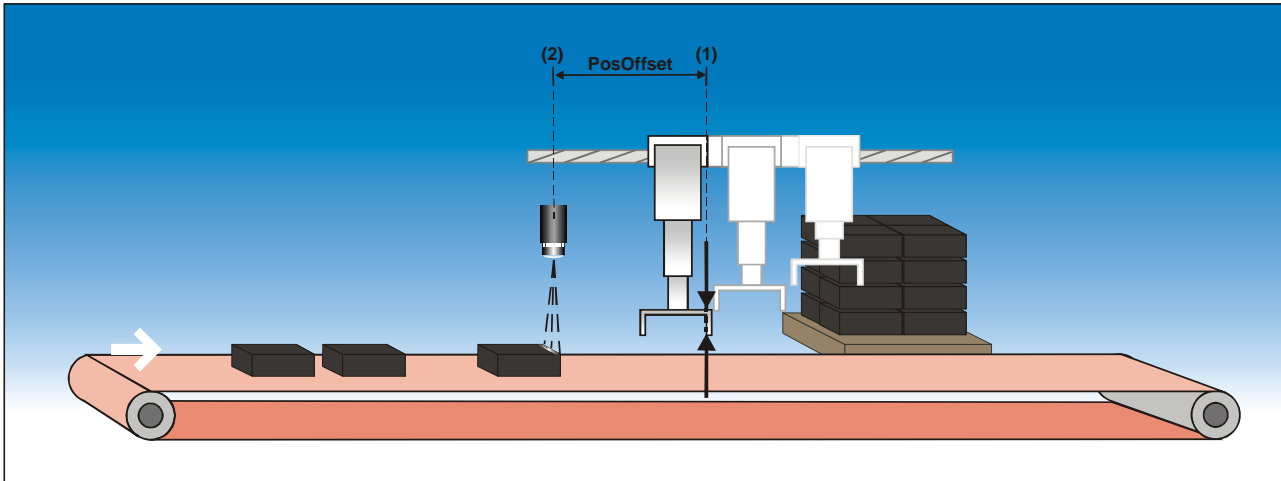
//--- Führe Schnitt aus
...

//--- Sägezyklus
siehe oben, bei Variante 1...
```

5 Fliegende Säge mit Hardwaresignal

5.1 Variante 1: Direktes Starten nach Hardwaresignal, Fall 1

Der Sägezyklus wird direkt nach Erhalt des zugehörigen Hardwaresignals gestartet. Das Hardwaresignal erfasst das Material so, dass während dem Hardwaresignal und dem Start des zugehörigen Sägezyklus keine weiteren Hardwaresignale erzeugt werden, die wieder einen Sägezyklus starten.



(1) Startposition der Sägeachse

(2) Ansprechposition des Materialsensors

(PosOffset) Übergabeparameter PosOffset für Funktion MotionFS_StartPosCap

Abb. 13 Fliegende Säge mit Hardwaresignal – Fall 1

5.1.1 Rahmenbedingungen

- Der Materialsensor muss mit entsprechendem Abstand *vor* der Werkzeugeinheit platziert werden, siehe Kapitel 9.8.7 „MotionFS_StartCapPos“ auf Seite 53.
- Der Positionsbereich des Motion-Masters muss mindestens so groß sein, wie die Distanz zwischen Ansprechposition des Materialsensors und Startposition der Sägeachse, Distanz = PosOffset.

ACHTUNG: D.h. der Parameter *AutoCalcMasterPosLimit* beim Aufruf der Funktion *MotionFS_Ini* muss eventuell auf 0 gesetzt sein. Der Anwender legt dann den Positionsbereich des Motion-Master, wie im letzten Absatz beschrieben, manuell fest.

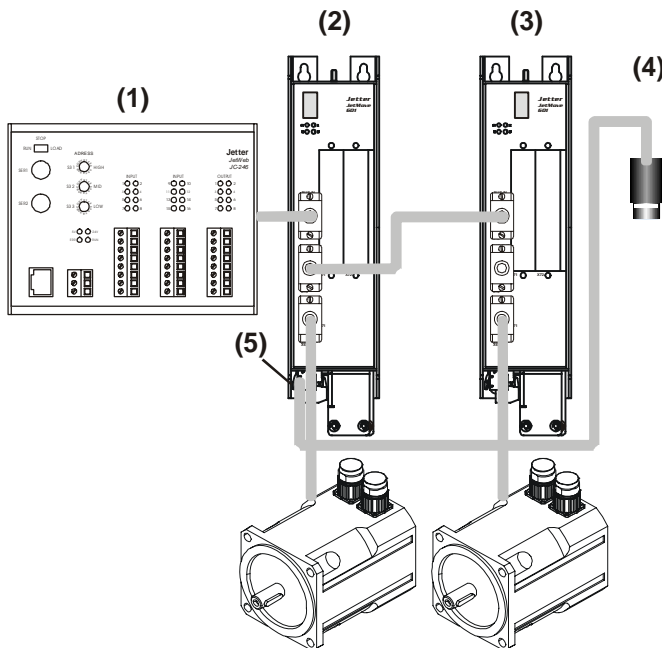
- Die Dauer von einem Hardwaresignal zum nächsten darf nicht kürzer sein, als die Dauer des Sägezyklus.

5.1.2 Hardwarekonfiguration



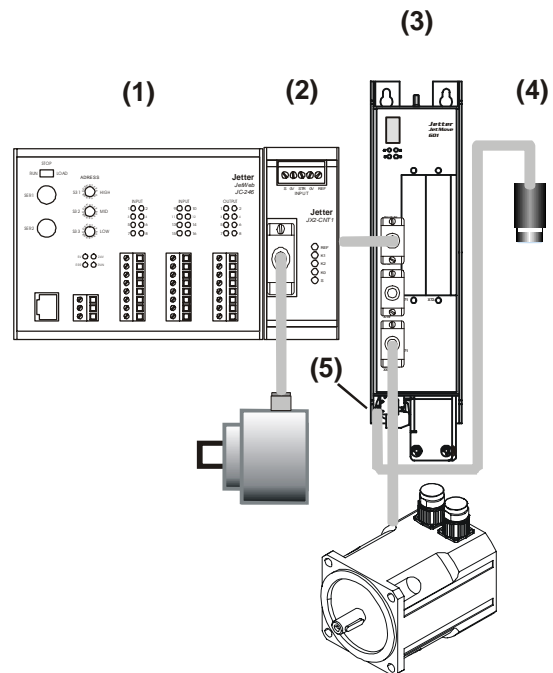
Beim Einsatz des JX2-CNT1 als Motion-Master wird zum Capturen der Masterposition die Funktion „Fliegende Referenzierung“ auf der Sägeachse verwendet.

Wichtig



- (1) Steuerung
- (2) Motion-Master: JM-2xx
- (3) Sägeachse: JM-2xx
- (4) Materialsensor
- (5) Eingang „INPUT“ am Motion-Master

Abb. 14 Hardwarekonfiguration mit JM-2xx



- (1) Steuerung
- (2) Motion-Master: JX2-CNT1
- (3) Sägeachse: JM-2xx
- (4) Materialsensor
- (5) Eingang „INPUT“ an Sägeachse

Abb. 15 Hardwarekonfiguration mit JX2-CNT1

5.1.3 Beispielprogrammierung

Die Beispielapplikation ist in Abb. 13 dargestellt. Die Distanz zwischen Ansprechposition des Materialsensors und Startposition der Sägeachse (PosOffset) beträgt 1000 mm. Siehe auch Prozedur „su_DirectTriggerEndlessCutCase1“ im Beispielprogramm.

```
//--- Produktionsperiode
...
//--- Sägezyklus
  WHILE <Zyklusmerker> DO      //--- Solange Zyklusmerker gesetzt...
    //--- Fahre Sägeachse zur Startposition
    MotionMovePtp (Ax, Alle, Absolut, <SägeachseStartposition>,...)
    WHEN MotionReadStatus (Ax, Im Zielfenster) CONTINUE;

    //--- Einschalten der Capturfunktion für die Motion-Master-Position
    //   und warten bis Captureereignis eingetreten ist,
    //   und speichere Captureposition in f_CapturePos
    ...(siehe im Beispielprogramm)
    f_CapturePos:= ...
```

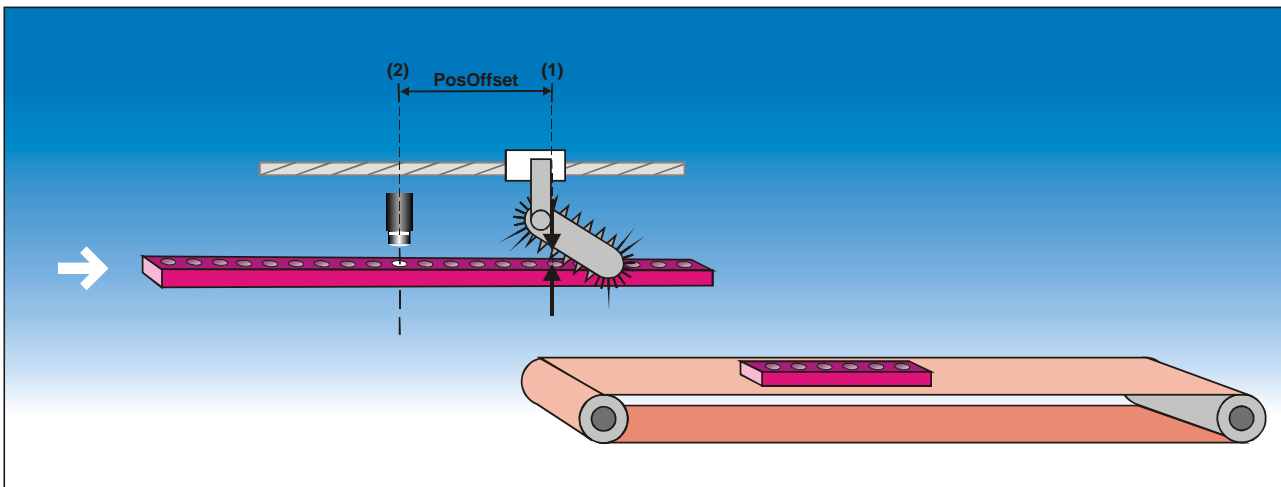
```
//--- Starte nach dem Captureereignis
MotionFS_StartCapPos(&st_MotionFS_Slave0Data,
                    &st_MotionFS_Slave0TabData0,&f_CapturePos,1000,0,
                    &st_MotionFS_TmpFloatVars0);

//--- Führe Schnitt aus
...

END_WHILE
...
```

5.2 Variante 1: Direktes Starten nach Hardware signal, Fall 2

Im Fall 2 wird das Hardware signal zusätzlich gezählt und führt erst nach einer bestimmten Anzahl von Signalen zum Start eines Sägezyklus.



- (1) Startposition der Sägeachse
(2) Ansprechposition des Materialsenors
(PosOffset) Übergabeparameter PosOffset für Funktion MotionFS_StartPosCap

Abb. 16 Fliegende Säge mit Hardware signal – Fall 2

5.2.1 Rahmenbedingungen

Rahmenbedingungen wie oben, bei Fall 1 und zusätzlich:

- Die Zählung der Hardware signale und der Start der Sägeachse muss jeweils in einem separaten Task ablaufen, so dass sie parallel zueinander durchgeführt werden.
- Die Hardware signale müssen in solch einem Abstand zueinander eintreffen, dass der Task, für die Zählung, jedes Signal noch erkennen kann. Hierbei ist die Taskzykluszeit der Steuerung zu beachten.

5.2.2 Hardwarekonfiguration

Siehe oben, bei Fall 1.

5.2.3 Beispielprogrammierung

Wie Beispiel für Fall 1, nur mit zusätzlicher Zählung der Captureereignisse in einem separaten Task. Siehe Prozedur „su_DirectTriggerEndlessCutCase2“ im Beispielprogramm.

```
TASK t_Zählung
...
//--- Zählung der Captureereignisse
IF <ZählungAktivFlag> THEN
    //--- Capturefunktion aktivieren und auswerten
    ...(siehe Beispielprogramm)
    //--- Captureereigniszähler erhöhen
    ...
    //--- Wenn Capturezähler Maximalwert erreicht hat, dann abspeichern der
    //      Position
    f_CapturePos:= ...

END_IF;

...
END_TASK;

TASK Produktionsperiode
...
//--- Produktionsperiode
...
//--- Sägezyklus
WHILE <Zyklusmerker> DO      //--- Solange Zyklusmerker gesetzt
    //--- Fahre Sägeachse zur Startposition
    MotionMovePtp (Ax, Alle, Absolut, <SägeachseStartposition>,...)
    WHEN MotionReadStatus (Ax, Im Zielfenster) CONTINUE;

    //--- Warte bis die notwendige Anzahl von Captureereignissen
    //      eingetroffen sind, im Beispiel = 7
    WHEN <CaptureCnt>= 7 CONTINUE;
    //--- Initialisiere wieder die Captureeinheit für die nächste
    //      Zählung
    ...

    //--- Starte nach dem Captureereignis
    //      Variable f_CapturePos wurde im Task t_Zählung bereits beschrieben
    MotionFS_StartCapPos(&st_MotionFS_Slave0Data,
                        &st_MotionFS_Slave0TabData0,&f_CapturePos,1000,0,
                        &st_MotionFS_TmpFloatVars0);

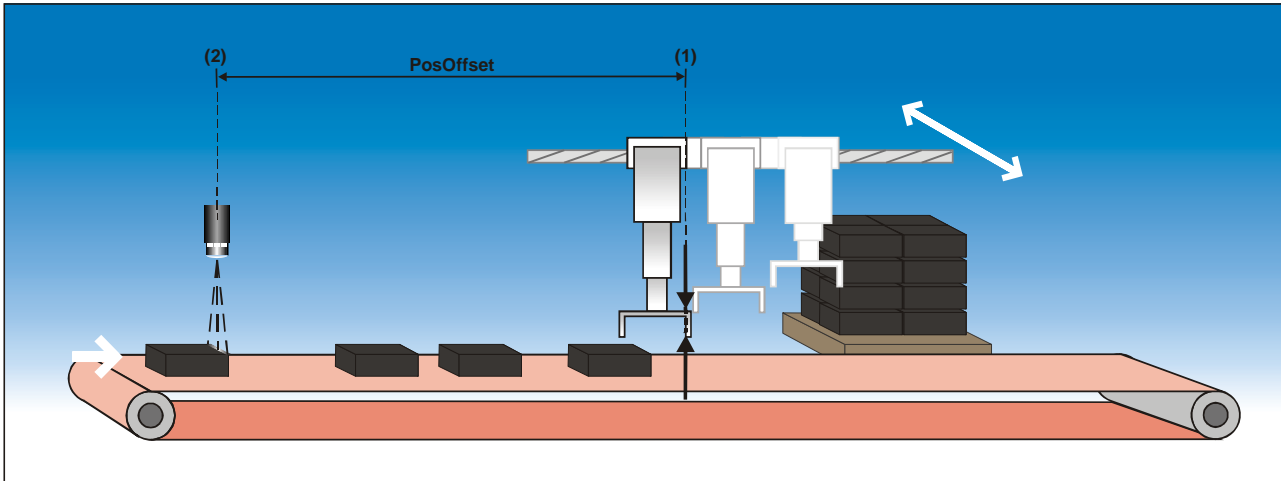
    //--- Führe Schnitt aus
    ...

END_WHILE
...

...
END_TASK;
```

5.3 Variante 2: Auflaufen von mehreren Hardwaresignalen

Bevor der Sägezyklus zum zugehörigen Hardwaresignal gestartet wird, läuft eine bestimmte Anzahl weiterer Hardwaresignale auf, die ebenfalls einen Sägezyklus starten.



- (1) Startposition der Sägeachse
(2) Ansprechposition des Materialsensors
(PosOffset) Übergabeparameter PosOffset für Funktion MotionFS_StartPosCap

Abb. 17 Fliegende Säge mit auflaufenden Hardwaresignalen

5.3.1 Rahmenbedingungen

- Der Materialsensor muss mit entsprechendem Abstand *vor* der Werkzeuginheit platziert werden, siehe Kapitel 9.8.7 „MotionFS_StartCapPos“ auf Seite 53.
- Der Positionsbereich des Motion-Masters muss mindestens so groß sein, wie die Distanz zwischen Ansprechposition des Materialsensors und Startposition der Sägeachse, Distanz = PosOffset.

ACHTUNG: D.h. der Parameter *AutoCalcMasterPosLimit* beim Aufruf der Funktion *MotionFS_Ini* muss auf 0 gesetzt sein. Der Anwender muss den Positionsbereich des Motion-Master, wie im letzten Absatz beschrieben, manuell festlegen.

- Die Dauer von einem Hardwaresignal zum nächsten darf nicht kürzer sein, als die Dauer des Sägezyklus.
- Ein FIFO zur Speicherung der Position des Motion-Master bei Erhalt des Hardwaresignals muss verwendet werden.
- Das Befüllen des FIFOs und der Start der Sägeachse muss jeweils in einem separaten Task ablaufen, so dass sie parallel zueinander durchgeführt werden.

5.3.2 Hardwarekonfiguration

Siehe oben, bei Variante 1.

5.3.3 Beispielprogrammierung

Der Parameter PosOffset ist im Beispiel 10.000 mm. Siehe Prozedur „su_FifoTriggerEndlessCut“ im Beispielprogramm.

```
TASK t_FIFO
...
//--- Befüllen des FIFOs
IF <FIFO_FüllenAktivFlag> THEN
    //--- Capturefunktion aktivieren und auswerten
    ...(siehe Beispielprogramm)
```

```

        //--- Bei Captureereignis, nächstes FIFO-Element mit dem FIFO-Inputzeiger
        //      füllen
        ... (siehe Beispielprogramm)
        //--- FIFO-Inputzeiger erhöhen
        ... (siehe Beispielprogramm)
    END_IF;

    ...
END_TASK;

TASK Produktionsperiode
    ...
    //--- Produktionsperiode
    ...
    //--- Sägezyklus
    WHILE <Zyklusmerker> DO      //--- Solange Zyklusmerker gesetzt
        //--- Fahre Sägeachse zur Startposition
        MotionMovePtp (Ax, Alle, Absolut, <SägeachseStartposition>, ...)
        WHEN MotionReadStatus (Ax, Im Zielfenster) CONTINUE;

        //--- Warte bis das nächste FIFO-Element einen Eintrag enthält
        WHEN <FIFO_AusgangsZeiger> <> <Vergleichswert> CONTINUE;

        //--- Capturepositionsvariable setzen
        f_CapturePos:= @FIFO_AusgangsZeiger;
        ...

        //--- FIFO-Ausgangszeiger erhöhen
        ... (siehe Beispielprogramm)

        //--- Starte nach dem Captureereignis
        MotionFS_StartCapPos(&st_MotionFS_Slave0Data,
                             &st_MotionFS_Slave0TabData0, &f_CapturePos, 10000, 0,
                             &st_MotionFS_TmpFloatVars0);

        //--- Führe Schnitt aus
        ...

    END_WHILE
    ...

    ...
END_TASK;

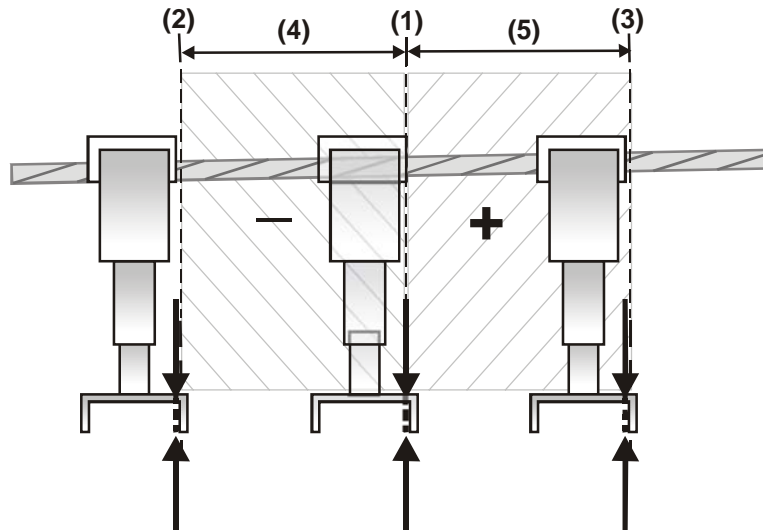
```

6 Zusätzliche Anforderungen

6.1 Startposition der Sägeachse verändern

Wenn die Startposition der Sägeachse nicht die gleiche ist, wie die, die bei der Ausführung der Funktion MotionFS_LoadTabLinear bzw. MotionFS_LoadTabSin2 angegeben wurde, dann ist der Schnittpunkt um die Differenz zwischen angegebener Startposition und aktueller Startposition verschoben. D.h. bei einer klassischen fliegenden Säge muss zur Schnittlänge noch diese Differenz addiert werden und bei einer fliegenden Säge mit Hardware-Signal muss diese Differenz zum Abstand (PosOffset) addiert werden, den der Materialsensor zur Startposition hat.

Eine Veränderung der angegebenen Startposition ist z.B. bei Schnitten ohne Rückfahrt notwendig, siehe nächstes Kapitel.



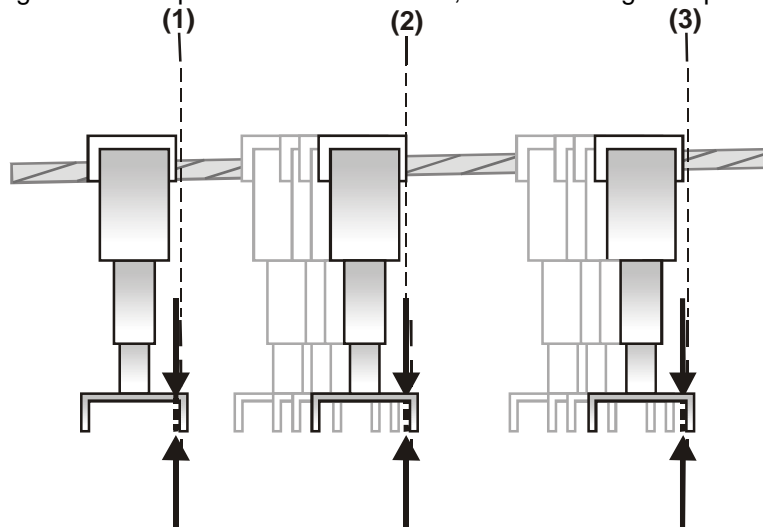
- (1) Bei der Tabellendefinition angegebene Startposition der Sägeachse
- (2) Startposition mit negativer Differenz (4) zur angegebenen Startposition
- (3) Startposition mit positiver Differenz (5) zur angegebenen Startposition

Abb. 18 **Verändern der Startposition**

6.2 Schnitt ohne Rückfahrt

Um z.B. kürzere Schnittlängen zu schneiden, als die Zykluszeit mit Rückfahrt zulässt, können einige Schnitte ohne Rückfahrt durchgeführt werden. Dadurch wird die Rückfahrzeit eingespart. Die Anzahl dieser Schnitte wird durch den Verfahrensweg der Sägeachse bestimmt.

Dabei muss nach jedem Schnitt, bei dem keine Rückfahrt stattgefunden hat, die Schnittlänge für den nächsten Schnitt korrigiert werden: Auf die gewünschte Schnittlänge muss die Differenz zwischen aktueller Startposition und angegebener Startposition addiert werden, siehe vorheriges Kapitel.



- (1) Startposition Zyklus n (angegebene Startposition)
- (2) Startposition Zyklus n+1
- (3) Startposition Zyklus n+2

Abb. 19 **Schnitte ohne Rückfahrt**

6.3 Unterschiedliche Geschwindigkeit zwischen Sägeachse und Motion-Master

Hier sind folgende Anwendungsbeispiele zu nennen:

- Motion-Master und Sägeachsen haben unterschiedliche Positionseinheiten: Der Motion-Master ist z.B. ein Förderband (Position in mm) und die Sägeachse eine Trommel (Position in °).
- Motion-Master und Sägeachsen sind rotatorische Achsen und haben unterschiedliche Umfänge.
- Diagonalschneider: Sägeachse und Motion-Master sind lineare Achsen, bewegen sich aber NICHT parallel zueinander

Bei diesen Anwendungen kommt der Parameter TabGearRatio zum Einsatz. Dieser Parameter wird beim Ausführen der Funktion MotionFS_LoadTab_Linear / MotionFS_LoadTab_Sin2 angegeben. Der Parameter ist normalerweise = 1. Bei diesen Anwendungen ist er $\neq 1$.

Der Parameter gibt das Verhältnis an, mit der sich die Sägeachse während der Synchronphase langsamer oder schneller als der Motion-Master bewegt.

TabGearRatio < 1 = Sägeachse bewegt sich langsamer als mit TabGearRatio = 1
TabGearRatio > 1 = Sägeachse bewegt sich schneller als mit TabGearRatio = 1

6.4 Tabellenwechsel

Ändern sich kurzfristig die Rahmenbedingungen für einen Schnitt, so kann von Sägezyklus zu Sägezyklus die Tabelle gewechselt werden. Eine Änderung der Rahmenbedingungen wäre z.B. eine wesentliche Änderung der Geschwindigkeit des Motion-Masters. Oder eine wesentliche Änderung der Beschleunigung der Sägeachse, um z.B. einen kürzeren Schnitt durchzuführen.

Voraussetzung für den Tabellenwechsel ist eine vorhandene Tabellendefinition der Tabelle, zu der gewechselt werden soll.

Der Tabellenwechsel ist für die fliegende Säge mit Hardwaresignal ganz einfach: Bei jedem Aufruf der Startfunktion MotionFS_StartCapPos wird die entsprechende Tabelle angegeben.

Bei einer klassischen fliegenden Säge ist etwas mehr zu beachten: Mit der Funktion MotionFS_StartChangeTab wird der nächste Sägezyklus mit der neuen Tabelle gestartet. Für den Zyklus nach dem Wechsel kann dann wieder die Funktion MotionFS_Start mit der neuen Tabelle aufgerufen werden. Zum Wechsel auf die alte Tabelle wird der Vorgang wieder mit MotionFS_StartChangeTab eingeleitet. Es ist auch möglich, bei jedem Schnitt zu wechseln, dann muss der Zyklus immer mit der Funktion MotionFS_StartChangeTab gestartet werden.

Beispiel

Siehe auch Kapitel 9.8.5 „MotionFS_StartChangeTab“ auf Seite 50.

```
...
//--- Wechsel von Tabelle 0 auf Tabelle 1
    MotionFS_StartChangeTab(&st_MotionFS_Slave0Data,&st_MotionFS_Slave0TabData1,
                           &st_MotionFS_Slave0TabData0,...);
...

...
//--- Weitere Schnitte mit Tabelle 1
    MotionFS_Start(&st_MotionFS_Slave0Data,&st_MotionFS_Slave0TabData1,
                  &f_CuttingLength,0,&st_MotionFS_TmpFloatVars0);
...

...
//--- Wieder Wechsel auf Tabelle 0 von Tabelle 1
    MotionFS_StartChangeTab(&st_MotionFS_Slave0Data,&st_MotionFS_Slave0TabData0,
```

```

...
    &st_MotionFS_Slave0TabData1,...);
...
//--- Weitere Schnitte mit Tabelle 0
    MotionFS_Start(&st_MotionFS_Slave0Data,&st_MotionFS_Slave0TabData0,
                  &f_CuttingLength,0,&st_MotionFS_TmpFloatVars0);
...

```

7 Schnittgenauigkeit

7.1 Ungenauigkeit der Übersetzungsangaben

Die Schnittlänge bzw. der Abstand zum Materialsensor wird in Millimeter bzw. Grad vorgegeben. Sind die Übersetzungsangaben vom Motorgeber auf die Mechanik (nach Millimeter bzw. Grad) ungenau, dann werden die Schnitte ungenau ausgeführt. Beim Nachmessen der geschnittenen Produkte wird dann festgestellt, dass sie immer zu kurz bzw. zu lang sind.

Durch eine ungenaue Angabe des Übersetzungsverhältnisses ist ein Millimeter bzw. Grad an der Mechanik etwas weniger oder mehr als ein Millimeter bzw. Grad. Es kann nur so genau geschnitten werden, wie die Genauigkeit der Übersetzungsangabe nach Millimeter bzw. Grad ist.

Dies betrifft die Übersetzungsangaben von Motorgeber auf die Mechanik beim Motion-Master und bei der Sägeachse. Wenn ein JX2-CNT1 eingesetzt wird, dann betrifft dies auch das Übersetzungsverhältnis von der inkrementellen Auflösung auf Millimeter bzw. Grad. Die Einstellung dafür wird in der Sägeachse durchgeführt.

7.2 Schnittpunktkorrektur

Manuelles Korrigieren des Schnittpunktes ist während den Sägezyklen durch die Schnittpunktkorrektur möglich. Zur Korrektur wird der Parameter CuttingPointCorr in der Struktur MOTION_FS_SLAVE_DATA verwendet, siehe dazu Kapitel 9.5.1 „MOTION_FS_SLAVE_DATA“ auf Seite 35. Über diesen Parameter wird ein Korrekturoffset vorgegeben, der beim nächsten Zyklusstart wirksam wird. Danach wird der Parameter wieder auf 0 gesetzt.

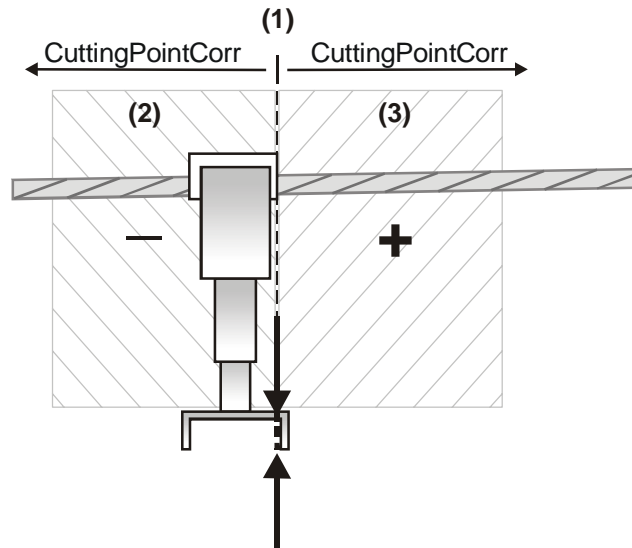
Beispiel

Der Korrekturoffset wird im Beispiel über eine Taste der Visualisierung beim Betätigen der Taste auf 10 mm gesetzt. Er kann aber auch über einen Setup-Screen direkt in den Parameter CuttingPointCorr eingetragen werden.

```

TASK <z.B. Tastenmanagement der Visu>
...
//--- Verschiebung des Schnittpunkts um 10 mm nach hinten (Schnitte werden länger)
    st_MotionFS_Slave0Data.CuttingPointCorr:= 10;
...

```



- (1) Aktueller Schnittpunkt
- (2) Negativer Wertebereich für CuttingPointCorr: Schnittpunktverschiebung nach hinten: Schnittlänge erhöht sich
- (3) Positiver Wertebereich für CuttingPointCorr: Schnittpunktverschiebung nach vorne: Schnittlänge verkürzt sich

Abb. 20 Schnittpunktkorrektur

Die Schnittpunktkorrektur über den Parameter CuttingPointCorr steht nur bei klassischen fliegenden Sägen zur Verfügung, also beim Start des Sägezyklus mit den Funktionen MotionFS_Start und MotionFS_StartChangeTab. Bei fliegenden Sägen mit Hardware-Signal, also beim Start mit Funktion MotionFS_StartCapPos, muss für eine Schnittpunktkorrektur der Übergabeparameter PosOffset korrigiert werden, siehe dazu Kapitel 9.8.7 „MotionFS_StartCapPos“ auf Seite 53.

Auch der Korrekturoffset ist abhängig von den Übersetzungsangaben des Motors zur Mechanik. Sind die Übersetzungsangaben nicht genau, dann ist der Korrekturoffset an der Mechanik etwas mehr oder weniger als der Vorgabewert in mm bzw. °. Wenn sich bei der Korrektur der Schnittpunkt nicht genau um den Korrekturoffset am Material verschiebt, dann stimmt das Übersetzungsverhältnis des Masters nicht.

7.3 Gleichlaufanpassung

Läuft die Sägeachse während der Synchronphase unerwünscht schneller oder langsamer als der Motion-Master, dann bedeutet das eine ungenaue Angabe des Übersetzungsverhältnisses zwischen Motor und Mechanik. Das Übersetzungsverhältnis ist entweder beim Motion-Master oder bei der Sägeachse oder bei beiden nicht genau eingestellt.

Die Ungenauigkeit lässt sich manuell über die Änderung des Synchronfaktors ermitteln. Der Synchronfaktor wird über den Parameter SyncFactor aus der Struktur MOTION_FS_SLAVE_DATA vorgegeben, siehe dazu Kapitel 9.5.1 „MOTION_FS_SLAVE_DATA“ auf Seite 35 und Kapitel 9.8.12 „MotionFS_SetSyncCorrection“ auf Seite 64. Der Synchronfaktor ist im Normalfall = 1. Durch Änderung des Faktors bewegt sich die Sägeachse langsamer oder schneller als der Motion-Master. Nachfolgend einige Beispielwerte:

SyncFactor = 0.5:	Sägeachse bewegt sich halb so schnell wie der Motion-Master
SyncFactor = 1:	Sägeachse bewegt sich genau so schnell wie der Motion-Master
SyncFactor = 2:	Sägeachse bewegt sich doppelt so schnell wie der Motion-Master

Die Änderung des Synchronfaktors wirkt sich erst beim nächsten Sägezyklus aus.



Wichtig

Die Gleichlaufenanpassung dient während der Inbetriebnahmephase zum Ermitteln der Ungenauigkeiten der Übersetzungsverhältnisse. Die Gleichlaufenanpassung soll danach über die Übersetzungsverhältnisse vorgenommen werden, so dass der Synchronfaktor für den Normalbetrieb wieder auf 1 gesetzt werden kann.



Wichtig

Bei jeder Änderung des Synchronfaktors verschiebt sich automatisch auch der Schnittpunkt. Die Verschiebung kann über die Schnittpunktkorrektur manuell ausgeglichen werden.

Beispiel

Der Synchronfaktor wird im Beispiel über eine Taste der Visualisierung beim Betätigen der Taste mit Hilfe der Funktion MotionFS_SetSyncCorrection um 0.01 verändert. Der Synchronfaktor kann aber auch über einen Setup-Screen direkt im Parameter SyncFactor gesetzt werden.

TASK <z.B. Tastenmanagement der Visu>

```
...
//--- Synchronfaktor um 0.01 erhöhen, d.h. Sägeachse läuft nachher schneller
//    als vorher.
f_SyncFactorCorr:= 0.01;
MotionFS_SetSyncCorrection(&st_MotionFS_Slave0Data,&f_SyncFactorCorr,0);
...
```

Der hier beschriebene Synchronfaktor hat die gleich Wirkung, wie der Parameter TabGearRatio, der bei der Tabellendefinition angegeben wird, siehe Kapitel 9.8.2 „MotionFS_LoadTab_Linear / MotionFS_LoadTab_Sin2“ auf Seite 43. Außer dass mit TabGearRatio der Synchronfaktor schon im Tabellenprofil enthalten ist. Dabei wird TabGearRatio nur auf die Synchronbewegung des Tabellenprofils angewandt.

Mit dem Parameter SyncFactor kann der Synchronfaktor dynamisch verändert werden. Allerdings wirkt sich der Synchronfaktor bei einem Wert $\neq 1$ auch auf die angegebene Beschleunigung aus. D.h. die Sägeachse beschleunigt nicht mit der Beschleunigung, die bei der Tabellendefinition angegeben wurde, wenn ein Synchronfaktor von $\neq 1$ eingestellt ist. Die Beschleunigung ist höher oder niedriger, abhängig vom Wert des Synchronfaktors.

7.4 Totzeit-Kompensation bei hoher Mastergeschwindigkeit

Die Sägeachse verarbeitet die Masterposition nicht genau zu dem Zeitpunkt, wenn sie beim Motion-Master erfasst wird. Eine Verzögerung entsteht durch die Kommunikationszeit der Masterposition zur Sägeachse und der Intervallzeit, mit der die Sägeachse die Masterposition verarbeitet. Auf Grund dieser Totzeit eilt die Sägeachse dem Motion-Master immer etwas hinterher. Je höher die Mastergeschwindigkeit, desto größer ist der absolute Versatz zwischen den beiden.

Dieser Versatz kann zu Schnittungenauigkeit führen, wenn mit hoher Mastergeschwindigkeit gefahren wird **und** die Mastergeschwindigkeit nicht konstant ist, sondern sich ständig wesentlich ändert. Mit einer Totzeit-Kompensation wird die aktuelle Position der Sägeachse um die Totzeit korrigiert. Das verbessert die Schnittgenauigkeit.

Die Totzeit-Kompensation ist per Default nicht aktiv. Zur Einstellung und Aktivierung, siehe Dokument "jm2xx_an_jetcontrol_bi_xxxx_benutzerinformation.pdf".

8 Fehlersuche

Sägeachse läuft nicht los

Möglichkeit 1: Verwendete Tabelle nicht definiert

Lösung: Tabelle definieren

Möglichkeit 2: Position des Motion-Master schon zu weit

Lösung bei Start mit Funktion MotionFS_Start und MotionFS_StartChangeTab:
Geschwindigkeit des Motion-Masters reduzieren.

Lösung bei Start mit Funktion MotionFS_StartInstant:
Konstante c_FS_STARTINSTANT_TIME_OF_PROCESS erhöhen.

Lösung bei Start mit Funktion MotionFS_StartCapPos:
Mechanische Distanz vom Materialsensor zur Startposition vergrößern.

Schnittpunkt oder Schnittlänge ist ungenau

Siehe Kapitel 7 „Schnittgenauigkeit“ auf Seite 27.

Schnittpunktkorrektur ist ungenau

Siehe Kapitel 7.2 „Schnittpunktkorrektur“ auf Seite 27.

Sägeachse läuft nicht synchron mit Motion-Master bzw. Werkzeug drückt beim Schnitt seitlich auf das Material

Siehe Kapitel 7.3 „Gleichlaufanpassung“ auf Seite 28.

Sägeachse fährt zu weit

Möglichkeit: Verfahrbereich der Sägeachse ist für den den eingestellten Synchronfaktor zu klein

Lösung: Verfahrbereich vergrößern.

9 Funktionsbibliothek ‚Fliegende Säge‘

9.1 Was ist die Funktionsbibliothek „Fliegende Säge“ ?

Die Funktionsbibliothek ist eine Sammlung von getesteten Funktionen, die den Programmierer von JetControl-Steuerungen der Jetter AG bei der Programmierung seiner Anwendung für fliegende Sägen mit der Programmiersprache JetSym-ST unterstützen sollen.

Die Bibliothek enthält Funktionen und die zugehörigen Definitionen von Datenstrukturen und Konstanten. Mit Hilfe der Bibliotheken ist der Programmierer in der Lage das Anwenderprogramm schneller umzusetzen und zu testen. Die Programmierzeiten werden somit geringer.

9.2 Was ist zu beachten ?

1. Das Einbinden einer Funktionsbibliothek erhöht die Programmgröße des JetSym-ST-Projekts.

9.3 Funktionen in JetSym-ST

Die Programmiersprache JetSym-ST sieht für Funktionen folgendes vor:

1. Übergabeparameter sind immer vom Typ INT
2. Rückgabeparameter ist immer vom Typ INT

Wenn als Übergabeparameter vom Typ FLOAT oder ganze Strukturen an eine Funktion übergeben werden müssen, dann wird als Übergabeparameter die Registeradresse der jeweiligen Float-Variable bzw. die Startadresse der Struktur angegeben. Die Adressen sind vom Typ INT.

Beispiel

Bei der Funktion `xyz(p_FloatPointer, p_StrukturPointer)` wird die Float-Variable `f_IchBinFloat` und die Struktur `st_IchBinStruktur` übergeben.

Funktionsaufruf mit dem &-Operator

```
xyz(&f_IchBinFloat, &st_IchBinStruktur);
```

Funktionsaufruf mit absoluten Adressen

```
xyz(65051, 450);
```

Verarbeitung innerhalb Funktion

Innerhalb der Funktion wird auf die Übergabeparameter indirekt über die Adresse zugegriffen:

```
...
VAR
    pl_HilfsStrukturPointer:    INT;           //-- Deklaration eines
                                           //-- lokalen Hilfspointers,
                                           //-- damit die Variable mit
                                           //-- der Startadresse nicht
                                           //-- verändert werden muss

END_VAR;

...
/-- Verabeitung der Float-Variablen
..... @p_FloatPointer * 1.23;                //-- Float-Variable
                                           //-- innerhalb einer Formel
                                           //-- verwendet

...
/-- Verabeitung der Struktur
pl_HilfsStrukturPointer:=p_StrukturPointer +1;  //-- Hilfspointer auf
                                           //-- 2. Element der
                                           //-- Struktur setzen
```

```
IF @pl_HilfsStrukturPointer > 100 THEN    //--- 2. Element der Struktur
    ...                                //    innerhalb einer IF-
                                      //    Bedingung
...

```

9.4 Einbinden der Bibliotheksdateien

Die Funktionsbibliothek besteht aus zwei Dateien:

- Header-Datei
- Funktionsdatei

Damit die Bibliothek genutzt werden kann, müssen beide Dateien in das JetSym-ST-Projekt eingebunden werden.

Das Einbinden geschieht über die Angabe der Dateinamen zusammen mit der Compilerdirektive `#INCLUDE` in der Hauptprogrammdatei. Dabei muss die Angabe vor dem ersten TASK-Statement gemacht werden:

Hauptprogrammfile:

```
//--- Definition der Typen und Konstanten
...
#include "LibJM2xxFS.h.stp"           // Einfügen der Headerdatei
...
//--- Definition der Funktionen
...
#include "LibJM2xxFS.stp"             // Einfügen der Funktionsdatei
...
//--- Hauptprogramm
...
TASK 0                               // Hier fängt der 1.Task an

```

Wenn sich die Dateien nicht im gleichen Verzeichnis befinden, wie das JetSym-ST-Projekt, in das sie eingefügt werden sollen, dann muss der Pfad mit dem kompletten Dateinamen in der INCLUDE-Anweisung angegeben werden. Oder die Dateien müssen im Projektbaum unter "Programm" und "Include" eingefügt werden, siehe Abb. 1. Dann ist der komplette Pfadangabe über das Einfügen bekannt.

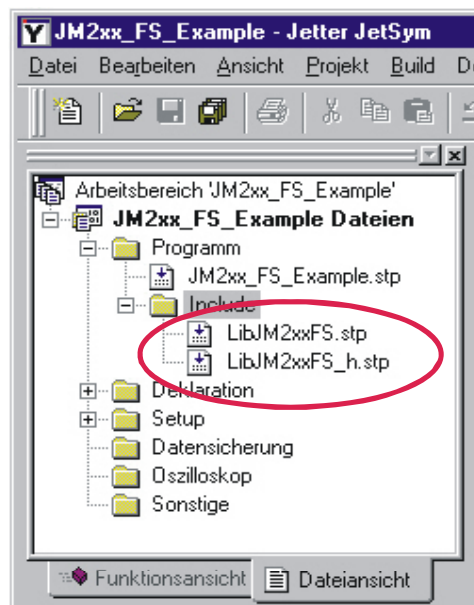


Abb. 1: Include-Dateien im Projektbaum

9.5 Typen

Typenübersicht
Strukturen
MOTION_FS_SLAVE_DATA Definition der Daten für eine fliegende Sägeachse.
MOTION_FS_TAB_DATA Definition der Daten für eine Tabelle, die die Beschleunigungs- und Synchronbewegung einer fliegenden Sägeachse festlegt.
MOTION_FS_TMP_FLOATVARS Definition der temporären FLOAT-Variablen, die als FLOAT-Zwischenspeicher für Berechnungen innerhalb der Funktionen für die fliegende Säge benutzt werden.

Die in der obigen Tabelle aufgeführten Struktur-Datentypen werden von den Funktionen der Funktionsbibliothek benutzt. Damit die Funktionen die Strukturen benutzen können, muss für jede Struktur, im Programmbereich VAR und END_VAR, mindestens eine Variable für jede Struktur deklariert werden. Für jede Variable muss dann eine Registeradresse festgelegt werden, an der die Struktur beginnt. Alle Variablen der Strukturen **müssen** komplett im FLOAT-Registerbereich der Steuerung liegen.

Für jede vorhandene fliegende Sägeachse (es können auch mehrere Sägeachsen an einem Systembusstrang vorhanden sein) muss eine Variable der Struktur MOTION_FS_SLAVE_DATA angelegt werden.

Für jede vorhandene Sägeachse muss mindestens eine Tabelle definiert werden, die die Beschleunigungs- und Synchronbewegung für den Sägezyklus festlegt. Es können auch mehrere Tabellen für eine fliegende Sägeachse definiert werden, zwischen denen gewechselt werden kann. Für jede Tabelle einer Sägeachse muss also mindestens eine Variable der Struktur MOTION_FS_TAB_DATA angelegt werden.

Beim Aufruf von einigen Funktionen für die fliegende Säge muss eine Variable der Struktur MOTION_FS_TMP_FLOATVARS als Parameter mitgegeben werden. Die Struktur wird intern zum Speichern von Zwischenwerten für Berechnungen etc. verwendet. Solange die Funktionen für die fliegende Säge immer nacheinander aufgerufen werden, reicht die Deklaration einer Variable dieser Struktur. Dabei ist es egal, für welche fliegende Sägeachse oder für welche Tabelle die Funktion ausgeführt wird. Sobald allerdings die Funktionen parallel, also zur gleichen Zeit in unterschiedlichen Tasks ausgeführt werden, müssen entsprechend viele Variablen dieser Struktur vorhanden sein. Dann wird für jede Funktion, die parallel zu einer anderen ausgeführt wird, eine eigene Variable dieser Struktur übergeben.

Die Strukturen werden im Normalfall von den Funktionen für die fliegende Säge initialisiert, verändert und gelesen. Das Steuerungsprogramm muss nur dafür sorgen, dass die Struktur-Variablen deklariert und vorhanden sind. Das Steuerungsprogramm soll im Normalfall außerhalb der Funktionen für die fliegende Säge die Parameter innerhalb der Strukturen nicht verändern. Hiervon ausgenommen sind die Parameter CuttingPointCorr und SyncFactor der Struktur MOTION_FS_SLAVE_DATA. Diese Parameter müssen vom Steuerungsprogrammteil des Anwenders beschrieben werden, wenn eine Schnittpunktkorrektur oder eine Gleichlaufanpassung durchgeführt werden soll, siehe dazu Kapitel 7 „Schnittgenauigkeit“ auf Seite 27. Die Strukturdaten können aber jederzeit über das Steuerungsprogramm gelesen oder über den JetSym-Setup-Seiten beobachtet werden.

Einige der Daten, die in den Strukturen gespeichert werden, z.B. in der Struktur MOTION_FS_TAB_DATA sind auch als Parameter auf der entsprechenden Sägeachse vorhanden. Aus Optimierungsgründen, und um Buszugriffe zu minimieren und dadurch Laufzeit einzusparen, sind diese Daten in den Strukturen zusätzlich im Speicher der Steuerung abgelegt.

Beispiel

Im folgenden Beispiel werden alle notwendigen Variablen für eine fliegende Sägeachse mit einer Tabelle in einer JC-24x im FLOAT-Registerbereich deklariert.

```
VAR
...
st_MotionFS_Slave0Data:    MOTION_FS_SLAVE_DATA AT %VL 65024;

st_MotionFS_Slave0TabData0: MOTION_FS_TAB_DATA AT %VL 65024 +
                           SIZEOF(st_MotionFS_Slave0Data);

st_MotionFS_TmpFloatVars:  MOTION_FS_TMP_FLOATVARS AT %VL 65024 +
                           SIZEOF(st_MotionFS_Slave0Data) +
                           SIZEOF(st_MotionFS_Slave0TabData0);

...
VAR_END;
```

9.5.1 MOTION_FS_SLAVE_DATA

Definition

MOTION_FS_SLAVE_DATA:

STRUCT

SlaveBase, MasterPosMax, MasterPosMin, MasterPosRange, LastTabOffsM, ActTabOffsM,
ActMasterSyncPos, ActSyncPosTabIdx, MaxCuttingLength, CuttingPointCorr, SyncFactor,
Reserved0: FLOAT;

END_STRUCT;

Parameter

SlaveBase:	Startadresse des JM-2xx am Systembus, der fliegende Sägeachse ist, z.B. 12000 = erster JM-2xx direkt nach der Steuerung JC-24x.
MasterPosMax:	Die maximale Masterposition in mm bzw. °.
MasterPosMin:	Die minimale Masterposition in mm bzw. °.
MasterPosRange:	Der Masterpositionsbereich in mm bzw. °, abhängig von MasterPosMax und MasterPosMin.
LastTabOffsM:	Tabellen-Masteroffset in mm bzw. ° des letzten Sägezyklus
ActTabOffsM:	Tabellen-Masteroffset in mm bzw. ° des aktuellen Sägezyklus
ActMasterSyncPos:	Synchronposition des Masters in mm bzw. ° für den aktuellen Sägezyklus. Die Synchronposition, ist die Position, ab der die fliegend Sägeachse mit der Masterachse synchron läuft.
ActSyncPosTabIdx:	Index des Tabellenelements mit den Synchronpositionen des aktuellen Sägezyklus.
MaxCuttingLength:	Größtmögliche Schnittlänge in mm bzw. °. Hier wird der Wert der größtmöglichen Schnittlänge MaxCuttingLength gespeichert, die beim Aufruf der Funktion MotionFS_Ini an diese übergeben wird.
CuttingPointCorr:	Korrekturwert des Schnittpunktes. Hier muss ein relativer Korrekturwert in mm bzw. ° eingetragen werden, wenn eine Schnittpunktkorrektur durchgeführt werden soll. Die Korrektur wird dann beim nächsten Sägezyklus ausgeführt. Nach der Korrektur wird der Parameter automatisch wieder auf 0 gesetzt. Siehe dazu Kapitel 7.2 „Schnittpunktkorrektur“ auf Seite 27. Defaultwert: 0 Wertebereich: FLOAT-Grenzen, negative Werte = Schnittlänge erhöhen positive Werte = Schnittlänge verkürzen
SyncFactor:	Gleichlauffaktor. Hier muss ein Gleichlauffaktor $\neq 1$ angegeben werden, wenn der Gleichlauf zwischen Motion-Master und Motion-Slave (=Sägeachse) korrigiert werden muss.

Eine Änderung wird dann beim nächsten Sägezyklus übernommen. (Der Gleichlauffaktor wird so wie er hier eingetragen ist beim nächsten Sägezyklus in Achsregister R446 "Skalierungsfaktor Slave" der entsprechenden Tabelle in der Sägeachse eingetragen.) Siehe dazu Kapitel 7.3 „Gleichlaufenanpassung“ auf Seite 28 und Kapitel 9.8.12 „MotionFS_SetSyncCorrection“ auf Seite 64.

Defaultwert: 1 = Gleichlauffaktor 1:1
Wertebereich: FLOAT-Grenzen,
 < 1 = Sägeachse läuft langsamer als in
 Tabelle definiert
 > 1 = Sägeachse läuft schneller als in Tabelle
 definiert

Reserved0: Reserviert für spätere Erweiterungen

Beschreibung

Definition der Daten für eine fliegende Sägeachse. Die Daten werden von den Funktionen für die fliegenden Säge eingetragen, verwaltet und verwendet. Für jede vorhandene fliegende Sägeachse muss eine Variablen dieser Struktur deklariert werden.

Der Steuerungsprogrammteil des Anwenders greift nur auf die Parameter CuttingPointCorr und SyncFactor zu, wenn eine Schnittpunktkorrektur oder eine Gleichlaufenanpassung durchgeführt werden soll.

Einige Parameter beinhalten einen Wert, der nur im aktuellen Sägezyklus gültig ist, z.B. ActMasterSyncPos. Diese Parameter werden aktualisiert, nach dem eine der Funktionen für den Zyklusstart MotionFS_Start, MotionFS_StartCapPos, MotionFS_StartChangeTab oder MotionFS_StartInstant aufgerufen wird. Der Parameter LastTabOffsM wird ebenfalls zum gleichen Zeitpunkt aktualisiert.

9.5.2 MOTION_FS_TAB_DATA

Definition

```
MOTION_FS_TAB_DATA:
    STRUCT
        TabNum, SyncPosTabIdx, MasterSyncPos, MasterPosTabMin, SlaveSyncPos,
        SlavePosTabMin, TabStartIdx, TabGearRatio, Reserved0: FLOAT;
    END_STRUCT;
```

Parameter

TabNum:	Tabellenkonfigurationsnummer
SyncPosTabIdx:	Index des Tabellenelements mit den Synchronpositionen. Die Synchronposition, ist die Position, ab der die fliegend Sägeachse mit der Masterachse synchron läuft.
MasterSyncPos:	Synchronposition des Masters in mm bzw. °
MasterPosTabMin:	Erste Tabellenposition des Masters in mm bzw. °
SlaveSyncPos:	Synchronposition des Slaves in mm bzw. °
SlavePosTabMin:	Erste Tabellenposition des Slaves in mm bzw. °
TabStartIdx:	Index auf erstes Tabellenelement

TabGearRatio: Tabellen-Übersetzungsverhältnis zwischen Motion-Master und Motion-Slave. Hier wird der Wert des Tabellen-Übersetzungsverhältnisses TabGearRatio gespeichert, das beim Aufruf der Funktion MotionFS_LoadTab_Linear bzw. MotionFS_LoadTab_Sin2 an diese übergeben wird.

Reserved0: Reserviert für spätere Erweiterungen.

Beschreibung

Definition der Daten für eine Tabelle, die die Beschleunigungs- und Synchronbewegung einer fliegenden Sägeachse festlegt. Die Daten werden von den Funktionen für die fliegenden Säge eingetragen, verwaltet und verwendet. Für jede Tabellendefinition einer fliegenden Sägeachse muss eine Variable dieser Struktur deklariert werden.

9.5.3 MOTION_FS_TMP_FLOATVARS

Definition

```
MOTION_FS_TMP_FLOATVARS:
    STRUCT
        Tmp0, Tmp1, Tmp2, Tmp3, Tmp4, Tmp5, Tmp6, Tmp7: FLOAT;
    END_STRUCT;
```

Parameter

Tmp0...Tmp7: Temporäre FLOAT-Variablen

Beschreibung

Definition der temporären FLOAT-Variablen, die als FLOAT-Zwischenspeicher für Berechnungen innerhalb der Funktionen für die fliegende Säge benutzt werden. Sobald Funktionen für die fliegende Säge, die eine Variable dieser Struktur benötigen, parallel in unterschiedlichen Tasks aufgerufen werden, werden ebenso viele Variablen dieses Struktur benötigt, wie parallele Funktionsaufrufe vorhanden sind.

9.6 Konstanten

Konstantenübersicht	
Für Funktion MotionFS_StartInstant:	
c_FS_STARTINSTANT_TIME_OF_PROCESS = 100	Totzeit des Startprozesses für den Sofortschnitt in ms.
Für Funktion MotionFS_Ini:	
c_FS_INI_STAT_OK = 0	Ausführung O.K.
c_FS_INI_STAT_ERR_MASTER_ENABLED = -1	Fehler: Der Verfahrbereich des Motion-Master kann nicht gesetzt werden, da die Endstufe der Achse freigegeben ist. Endstufe muss gesperrt sein. Der Fehler kann nur auftreten, wenn AutoCalcMasterPosLimit = 1 gesetzt ist.
c_FS_INI_STAT_ERR_SLAVE_ENABLED = -2	Fehler: Der Masterpositionsbereich der fliegenden Sägeachse kann nicht gesetzt werden, da die Endstufe der Achse freigegeben ist. Endstufe muss gesperrt sein. Der Fehler kann nur auftreten, wenn AutoCalcMasterPosLimit = 1 gesetzt ist.

<p>c_FS_INI_STAT_ERR_MASTER_POSRANGE_TOO_SMALL = -3 Fehler: Verfahrbereich des Motion-Masters ist zu klein. Der Verfahrbereich muss 2 mal so groß sein, wie die angegebene größtmögliche Schnittlänge MaxCuttingLength.</p>
<p>Für Funktion MotionFS_LoadTab_Linear / MotionFS_LoadTab_Sin2:</p>
<p>c_FS_LOADTAB_STAT_OK = 0 Ausführung O.K.</p>
<p>c_FS_LOADTAB_STAT_ERR_ACC_PHASE_TOO_LONG = -1 Fehler: Beschleunigungsstrecke ist länger als die zu überprüfende Schnittlänge. pTmp7 = 0: Beschleunigungsstrecke > MOTION_FS_SLAVE_DATA.MaxCuttingLength pTmp7 > 0: Beschleunigungsstrecke > pTmp7</p>
<p>c_FS_LOADTAB_STAT_WARN_REMAINTIME_TOO_SMALL = 10 Warnung: Die verbleibende Zykluszeit, vom Ende der Synchronisierungsphase bis zum Ende der Tabelle, ist zu klein. pTmp6 = 0: verbleibende Zykluszeit < 1ms pTmp6 > 0: verbleibende Zykluszeit < pTmp6</p>
<p>Für Funktion MotionFS_Start / MotionFS_StartChangeTab:</p>
<p>c_FS_START_STAT_OK = 0 Ausführung O.K.</p>
<p>c_FS_START_STAT_ERR_MASTER_POS_TOO_FAR = -1 Fehler: Start des Zyklus nicht möglich, weil die Masterposition bereits größer ist, als die intern ermittelte Startposition.</p>
<p>Für Funktion MotionFS_StartInstant / MotionFS_StartCapPos:</p>
<p>c_FS_STARTINSTANT_STAT_OK = 0 Ausführung O.K.</p>
<p>Für Funktion MotionFS_StartStatus:</p>
<p>c_FS_STARTSTATUS_STAT_NO_TABLE_ACTIVE = 0 Tabellenfunktion ist nicht aktiv.</p>
<p>c_FS_STARTSTATUS_STAT_WAITING_FOR_START = 1 Fliegende Sägeachse wartet, bis Masterposition die intern ermittelte Startposition erreicht hat.</p>
<p>c_FS_STARTSTATUS_STAT_IN_RAMP = 2 Sägeachse befindet sich in der Synchronisierungsphase.</p>
<p>c_FS_STARTSTATUS_STAT_SYNCHRON = 3 Sägeachse ist mit der Masterachse synchron.</p>
<p>c_FS_STARTSTATUS_STAT_ERR_INCORRECT_STATUS = -1 Kein ordnungsgemäßer Status erkennbar: Tabelle ist zwar gestartet, aber kein sinnvoller Status für eine Fliegende-Säge-Anwendung konnte erkannt werden.</p>

9.6.1 c_FS_STARTINSTANT_TIME_OF_PROCESS

Diese Konstante wird ausschließlich von der Funktion MotionFS_StartInstant verwendet. Bei der Ausführung der Funktion MotionFS_StartInstant entsteht eine Totzeit vom Aufrufen der Funktion bis zum Zeitpunkt, an dem die Funktion das Kommando 46 zum Start der Tabellenfunktion für die Sägeachse gibt.

Während dieser Totzeit legt der Motion-Master eine bestimmte Strecke zurück. Die Funktion MotionFS_StartInstant muss diese Strecke rechnerisch ermitteln. Die Totzeit ist abhängig von der Zykluszeit der Steuerung und der Auslastung des Systembusses. Ebenso haben kleinere Mastergeschwindigkeiten (< 10 Umdrehungen/min) einen größeren Einfluss auf die rechnerisch ermittelte Motion-Master-Strecke als höhere Mastergeschwindigkeiten.

Die Konstante c_FS_STARTINSTANT_TIME_OF_PROCESS gibt dabei einen ungefähren Wert der Totzeit in ms an.



Wichtig

Wenn der Wert kleiner gewählt wird, als die tatsächliche Totzeit ist, wird der Sofortschnitt nicht sofort ausgeführt. Die Sägeachse wartet mit aktiver Tabellenfunktion solange, bis die Masterposition wieder die Startposition für die Tabelle erreicht hat. Dies passiert erst nachdem die Masterposition einen kompletten Modulodurchlauf vollzogen hat.

Wenn der Wert zu groß ist, dann entsteht unter Umständen eine unerwünschte Verzögerungszeit bis der Sofortschnitt ausgeführt wird.

Der Defaultwert von 100 ms ist ein Erfahrungswert, der genügend Puffer zur tatsächlich auftretenden Totzeit lässt und deshalb auch für kleinere Geschwindigkeiten geeignet ist. Wenn die entstehende Verzögerungszeit unerwünscht ist, vor allem bei größeren Geschwindigkeiten, kann der Wert reduziert werden. Allerdings sollte der Wert nicht unter 10 ms eingestellt sein.

9.7 Standard-Variablendeklaration für alle Beispiele

Die nachfolgende Variablendeklaration gilt für alle in diesem Dokument vorhandenen Beispielprogramme bzw. -programmteile.

```
VAR
...
st_MotionFS_Slave0Data:      MOTION_FS_SLAVE_DATA AT %VL 65024;
st_MotionFS_Slave0TabData0:  MOTION_FS_TAB_DATA AT %VL 65036;
st_MotionFS_Slave0TabData1:  MOTION_FS_TAB_DATA AT %VL 65045;
st_MotionFS_TmpFloatVars:    MOTION_FS_TMP_FLOATVARS AT %VL 65054;
...
f_MasterSpeed:               FLOAT AT %VL 65073;
f_SlaveAcc:                   FLOAT AT %VL 65074;
f_SlaveStartPos:              FLOAT AT %VL 65076;
f_TabGearRatio:               FLOAT AT %VL 65077;
f_CuttingLength:              FLOAT AT %VL 65070;
...
n_Return:                     INT AT %VL 200;
...
VAR_END;
```

9.8 Funktionen

Funktionsübersicht
Funktionen, die während der Initialisierungsphase des Steuerungsprogramms ausgeführt werden:
MotionFS_Ini (pMasterBase, pSlaveBase, pSlaveData, AutoCalcMasterPosLimit, pMaxCuttingLength, Option, pTmp0) Hauptfunktion zur Initialisierung der Fliegende-Säge-Anwendung.

<p>MotionFS_LoadTab_Linear (pSlaveData, pTabData, pSlaveStartPos, pMinCuttingLength, pMaxMasterSpeed, pMaxAcc, TabPoints, TabNum, TabStartIdx, pTmp0) Berechnet und lädt eine Fliegende-Säge-Tabelle mit linearer Geschwindigkeitsrampe in die Sägeachse.</p>
<p>MotionFS_LoadTab_Sin2 (pSlaveData, pTabData, pSlaveStartPos, pMinCuttingLength, pMaxMasterSpeed, pMaxAcc, TabPoints, TabNum, TabStartIdx, pTmp0) Berechnet und lädt eine Fliegende-Säge-Tabelle mit Sinus²-Geschwindigkeitsrampe in die Sägeachse.</p>
<p>Funktionen, zum Starten und Durchführen eines Sägezyklus:</p>
<p>MotionFS_Reset (pSlaveData, pTabData, Option) Initialisiert den Start der nächsten Produktionsperiode.</p>
<p>MotionFS_Start (pSlaveData, pTabData, pCuttingLength, Option, pTmp0) Startet einen neuen Sägezyklus für eine klassische fliegende Säge.</p>
<p>MotionFS_StartChangeTab (pSlaveData, pTabDataNext, pTabDataLast, pCuttingLength, Option, pTmp0) Startet einen neuen Sägezyklus für eine klassische fliegende Säge mit Tabellenwechsel.</p>
<p>MotionFS_StartInstant (pSlaveData, pTabData, Option, pTmp0) Startet einen Sofortschnitt.</p>
<p>MotionFS_StartCapPos (pSlaveData, pTabData, Option, pTmp0) Startet einen neuen Sägezyklus für eine fliegende Säge mit Hardwaresignal.</p>
<p>MotionFS_SetSyncCorrection (pSlaveData, pCorrection, Option) Setzt einen neue Synchronkorrekturwert für den nächsten Sägezyklus.</p>
<p>Funktionen, zur Ermittlung wichtiger Parameter:</p>
<p>MotionFS_GetStartStatus (pSlaveData, Option) Gibt den Status beim Starten eines neuen Sägezyklus zurück.</p>
<p>MotionFS_GetActCutLength (pSlaveData, Direction, Option, pTmp0, pActCutLength) Gibt die Länge in mm bzw. ° zurück, die das Material noch bis zum Synchronpunkt zurücklegt. Für Sägezyklen, die mit den Funktion MotionFS_Start oder MotionFS_StartChangeTab gestartet werden.</p>
<p>MotionFS_GetDistToCutPoint (pSlaveData, Option, pTmp0, pDistCutPoint) Gibt die Länge in mm bzw. ° zurück, die das Material noch bis zum Synchronpunkt zurücklegt. Für Sägezyklen, die mit den Funktion MotionFS_StartInstant oder MotionFS_StartCapPos gestartet werden.</p>
<p>MotionFS_GetMasterAlignPos (pTabData, Option, pTmp0, pMasterAlignPos) Gibt die Position in mm bzw. ° zurück, auf den der Master gesetzt werden muss, wenn die nächste Produktionsperiode ohne Kopfschnitt aber mit Ausrichtung von Material und Werkzeugeinheit gestartet werden soll.</p>

9.8.1 MotionFS_Ini

MotionFS_Ini (pMasterBase, pSlaveBase, pSlaveData, AutoCalcMasterPosLimit, pMaxCuttingLength, Option, pTmp0)

Parameter

pMasterBase:	Startadresse des JM-2xx am Systembus, der Motion-Master sein soll, z.B. 12000 = erster JM-2xx direkt nach der Steuerung. Wenn der Master ein JX2-CNT1 ist, dann muss pMasterBase = 0 sein.
pSlaveBase:	Startadresse des JM-2xx am Systembus, der Sägeachse sein soll, z.B. 13000 = zweiter JM-2xx direkt nach der Steuerung.
pSlaveData:	Startadresse der Struktur vom Typ MOTION_FS_SLAVE_DATA, die mit den Daten der angegebenen Sägeachse gefüllt werden soll.
AutoCalcMasterPosLimit:	Schalter, der festlegt, ob der Masterpositionsbereich im Motion-Master und in der Sägeachse automatisch berechnet und gesetzt werden soll oder nicht, siehe Funktionsbeschreibung. Wertebereich: 0;1 0 = Masterpositionsbereich wird vom Anwender festgelegt 1 = Masterpositionsbereich wird automatisch festgelegt
pMaxCuttingLength:	Adresse der FLOAT-Variablen, die die größtmögliche Schnittlänge in mm bzw. ° enthält, siehe Funktionsbeschreibung. Wertebereich der Schnittlänge: 0...250000 mm
Option:	Reserviert für Erweiterungen: Immer = 0.
pTmp0:	Startadresse der Struktur der temporären FLOAT-Variablen vom Typ MOTION_FS_TMP_FLOATVARS.

Rückgabewert Funktionsstatus. Die möglichen Werte sind:
c_FS_INI_STAT_OK
c_FS_INI_STAT_ERR_MASTER_ENABLED
c_FS_INI_STAT_ERR_SLAVE_ENABLED
c_FS_INI_STAT_ERR_MASTER_POSRANGE_TOO_SMALL
Siehe auch Kapitel 9.6 "Konstanten" auf Seite 37

Beschreibung

Die Funktion beschreibt die MOTION_FS_SLAVE_DATA-Struktur der jeweiligen Sägeachse mit Initialwerten und legt den Masterpositionsbereich, abhängig vom Schalter AutoCalcMasterPosLimit, automatisch festgelegt.

AutoCalcMasterPosLimit

Siehe dazu Kapitel 2.2 „Festlegen des Masterpositionsbereichs im Slave“ auf Seite 6.

0 = Verfahrbereich (Achsregister 182, R183) im Motion-Master und Masterpositionsbereich (Achsregister R158, R159) in der Sägeachse wird nicht verändert. Es wird davon ausgegangen, dass der Verfahrbereich und die Masterpositionsbereich bereits korrekt vom Anwender eingestellt wurde. Der Masterpositionsbereich muss mindestens doppelt so groß sein, wie die angegebene größtmögliche Schnittlänge MaxCuttingLength.

Ist dies nicht der Fall wird die Funktion mit der Fehlermeldung `c_FS_INI_STAT_ERR_MASTER_POSRANGE_TOO_SMALL` abgebrochen.

1 = Verfahrbereich (Achsregister R182, R183) im Motion-Master (falls Motion-Master kein JX2-CNT1) und Masterpositionsbereich (Achsregister R158, R159) in der Sägeachse werden abhängig von der Angabe der größtmöglichen Schnittlänge `MaxCuttingLength` automatisch festgelegt. Welcher Positionsbereich dann automatisch festgelegt wird, ist in Abb. 1 in Kapitel 2.2 „Festlegen des Masterpositionsbereichs im Slave“ auf Seite 6 beschrieben.

Zur automatischen Festlegung des Verfahrbereichs und des Masterpositionsbereichs muss die Endstufe des Motion-Master und der Sägeachse gesperrt sein. Ist dies nicht der Fall wird die Funktion mit einer der Fehlermeldungen `c_FS_INI_STAT_ERR_MASTER_ENABLED` oder `c_FS_INI_STAT_ERR_SLAVE_ENABLED` abgebrochen.

pMaxCuttingLength

Bei klassischen fliegenden Sägen wird hier die größtmögliche Schnittlänge eingetragen, die bei der Applikation vorkommen könnte. Bei fliegenden Sägen mit Hardwaresignal ist hier der maximale Abstand zwischen den Hardwaresignalen anzugeben.

Während dem Betrieb sind andere Schnittlängen / Abstände möglich. Sie können wesentlich kleiner sein, dürfen aber nie größer sein, als die hier angegebene `MaxCuttingLength`. Die Schnittlänge / Abstand ist bezogen auf die Positionsskalierung des Motion-Masters.

Funktionsausführung

Die Funktion wird während der Initialisierungsphase des Steuerungsprogramms ausgeführt, bevor irgendeine andere Funktion für die fliegende Säge ausgeführt wird. Sie muss für jede am System befindliche Sägeachse mit der jeweiligen `MOTION_FS_SLAVE_DATA`-Struktur ausgeführt werden.



Die Funktion muss vor der Konfiguration der Achsverbunde für den Master-Slave-Betrieb aufgerufen werden, siehe dazu Kapitel 2.1 Initialisierungsschritte und -Reihenfolge Seite 6.

Wichtig

Achsregistermanipulation

Folgende Achsregister werden im Motion-Master beschrieben, wenn `AutoCalcMasterPosLimit = 1`: R182, R183. Folgende Achsregister werden in der Sägeachse beschrieben: R435, R436 und wenn `AutoCalcMasterPosLimit = 1`: R158, R159.

Beispiel

Im folgenden Beispiel ist der Motion-Master im JetSym als 1. Achse definiert und der Motion-Slave (=Sägeachse) als 2. Achse. Die Achsen sind an einer JC-24x angeschlossen. Der Positionsbereich soll im Motion-Master und in der Sägeachse automatisch gesetzt werden. Die Schnittlänge, die nach der Initialisierung des Steuerungsprogramms für diese Sägeachse maximal vorkommen kann beträgt 5000 mm.

```
TASK t_Init
...
//--- Ausführen der FS-Initialisierung für Slave 0
f_CuttingLength:= 5000;      //--- Maximal Cutting Length
n_Return:=MotionFS_Ini(12000,13000,&st_MotionFS_Slave0Data,1,
                        &f_CuttingLength,0,&st_MotionFS_TmpFloatVars);
//--- Auswerten des Rückgabewerts
CASE n_Return OF
    c_FS_INI_STAT_ERR_MASTER_ENABLED:
        //--- Fehler: Motion-Master ist freigegeben
    ...
```

```
        BREAK;
    c_FS_INI_STAT_ERR_SLAVE_ENABLED:
        //--- Fehler: Sägeachse ist freigegeben
        ...
        BREAK;
    c_FS_INI_STAT_ERR_MASTER_POSRANGE_TOO_SMALL:
        //--- Fehler: Masterpositionsbereich ist zu klein gewählt
        ...
        BREAK;
    END_CASE;

    //--- Konfiguration des Achsverbunds
    ...
END_TASK;
```

9.8.2 MotionFS_LoadTab_Linear / MotionFS_LoadTab_Sin2

MotionFS_LoadTab_Linear (pSlaveData, pTabData, pSlaveStartPos, pMaxMasterSpeed, pMaxAcc, pTabGearRatio, TabPoints, TabNum, TabStartIdx, pTmp0)

MotionFS_LoadTab_Sin2 (pSlaveData, pTabData, pSlaveStartPos, pMaxMasterSpeed, pMaxAcc, pTabGearRatio, TabPoints, TabNum, TabStartIdx, pTmp0)

Parameter

pSlaveData:	Startadresse der Struktur vom Typ MOTION_FS_SLAVE_DATA für die Sägeachse, in die die Tabelle geladen werden soll.
pTabData:	Startadresse der Struktur vom Typ MOTION_FS_TAB_DATA, die mit den Daten der anzulegenden Tabelle gefüllt werden soll.
pSlaveStartPos:	Adresse der FLOAT-Variablen, in der die Startposition der Sägeachse in mm bzw. ° eingetragen ist, siehe Funktionsbeschreibung. Wertebereich der Startposition: Verfahrbereich der Sägeachse
pMaxMasterSpeed:	Adresse der FLOAT-Variablen, in der die Mastergeschwindigkeit in mm/s bzw. °/s eingetragen ist, mit der die anzulegende Tabelle maximal gefahren wird, siehe Funktionsbeschreibung. Wertebereich der Geschwindigkeit: 0...positive FLOAT-Grenze
pMaxAcc:	Adresse der FLOAT-Variablen, in der die Beschleunigung in mm/s ² bzw. °/s ² eingetragen ist, mit der die Sägeachse während der Synchronphase maximal beschleunigt werden soll, siehe Funktionsbeschreibung. Wertebereich der Beschleunigung: 0...positive FLOAT-Grenze

pTabGearRatio:	Adresse der FLOAT-Variablen, in der das Tabellenübersetzungsverhältnis für die Synchronphase zwischen der Masterposition und Slaveposition festgelegt wird, siehe Funktionsbeschreibung. Wertebereich des Übersetzungsverhältnis: FLOAT-Bereich Im Normalfall: 1 (=Übersetzung Master zu Slave 1:1)
TabPoints:	Anzahl der Punkte, die die anzulegende Tabelle haben soll, siehe Funktionsbeschreibung. Wertebereich: 10...4096 Empfohlener Wertebereich: 20 – 50 Mindestwert: 10
TabNum:	Tabellenkonfigurationsnummer, die die anzulegende Tabelle erhalten soll. Wertebereich: 0...23
TabStartIdx:	Startindex auf das Tabellenelement, ab dem die Punkte der anzulegenden Tabelle eingetragen werden sollen, siehe Funktionsbeschreibung. Wertebereich: 0...4095
pTmp0:	Startadresse der Struktur der temporären FLOAT-Variablen vom Typ MOTION_FS_TMP_FLOATVARS, siehe Funktionsbeschreibung.

Rückgabewert Funktionsstatus. Die möglichen Werte sind:
c_FS_LOADTAB_STAT_OK
c_FS_LOADTAB_STAT_ERR_ACC_PHASE_TOO_LONG
c_FS_LOADTAB_STAT_WARN_REMAINTIME_TOO_SMALL
Siehe auch Parameter pTmp0
Siehe auch Kapitel 9.6 "Konstanten" auf Seite 37

Beschreibung

Die Funktion legt eine Tabelle in der Sägeachse an, die die Beschleunigungs- und Synchronbewegung zwischen Motion-Master und der Sägeachse definiert. Abhängig davon, welche Funktion benutzt wird, ergibt sich eine lineare oder eine Sinus²-Geschwindigkeitsrampe für die Beschleunigungsbewegung der Sägeachse. Dabei wird die jeweilige MOTION_FS_TAB_DATA-Struktur beschrieben.

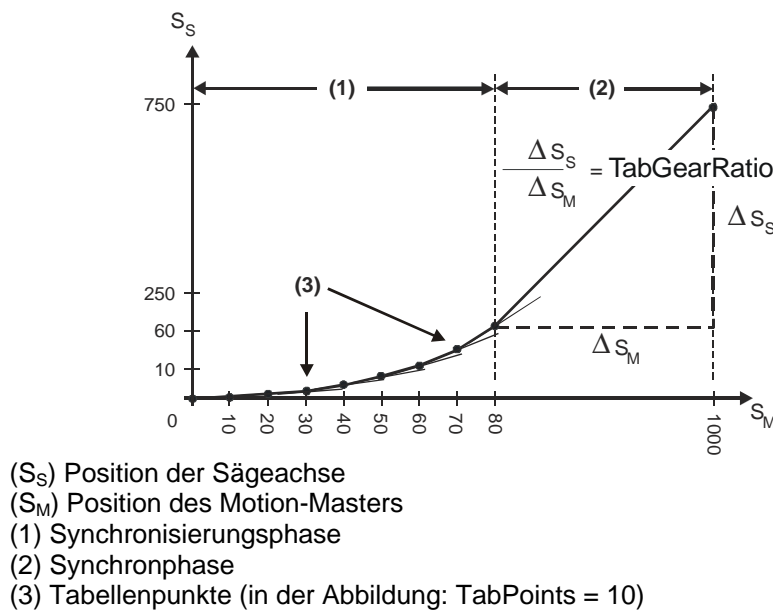


Abb. 21 Abbildung der Tabellenpunkte im Leit- Folgediagramm

Der Tabellenpositionsbereich des Motion-Master beginnt mit der Mitte des Masterpositionsbereich, der in der Sägeachse festgelegt ist. Und hat genau die Länge der größtmöglichen Schnittlänge MaxCuttingLength, die in der Funktion MotionFS_Ini angegeben wurde. Siehe dazu auch Kapitel 2.2 „Festlegen des Masterpositionsbereichs im Slave“ auf Seite 6. Der Tabellenpositionsbereich für die Sägeachse beginnt mit der Startposition SlaveStartPos. Das Ende ist dann abhängig vom Ende des Tabellenpositionsbereichs des Motion-Masters und der Angabe des Tabellenübersetzungsverhältnis TabGearRatio, siehe Abb. 21.



Wichtig

Die Sägeachse darf unter keinen Umständen zum Ende des Tabellepositionsbereich gelangen, da sie sonst abrupt anhält. Sie muss vor dem Ende entsprechend aus der Tabellenfunktion ausgekoppelt werden, siehe dazu Kapitel 3.6 „Ausklinken“ auf Seite 11. Die Länge der Tabelle (Länge des Masterpositionsbereichs der Tabelle) ist identisch mit dem Wert des Parameters MaxCuttingLength, der beim Aufruf der Funktion MotionFS_Ini übergeben wird.

pSlaveStartPos

Im Normalfall startet die Sägeachse von der Startposition aus. Es ist möglich, dass die Sägeachse während dem Betrieb auch von einer beliebigen anderen Position gestartet wird, ohne dass die Tabelle mit neuer Startposition neu konfiguriert werden muss. Dabei gibt es aber verschiedenes zu beachten, siehe dazu Kapitel 6.1 „Startposition der Sägeachse verändern“ auf Seite 24.

pMaxMasterSpeed / pMaxAcc

Während dem Betrieb muss die tatsächliche Mastergeschwindigkeit nicht genau mit dem angegebenen Wert übereinstimmen. Die angegebene Beschleunigung stellt sich aber bei der Sägeachse nur ein, wenn der Motion-Master auch tatsächlich die angegebene Geschwindigkeit fährt. Je langsamer die tatsächliche Mastergeschwindigkeit gegenüber der hier angegebenen ist, desto niedriger ist die Beschleunigung der Sägeachse und anders herum.

Wenn eine Mastergeschwindigkeit gefahren werden soll, die wesentlich von der hier angegebenen abweicht, muss eine neue Tabelle festgelegt werden, damit sich die angegebene Beschleunigung weitgehend an der Sägeachse einstellt.

pTabGearRatio

Im Normalfall ist das Tabellenübersetzungsverhältnis = 1. Für manche Anwendungen kann es aber erforderlich sein, die Sägeachse während der Synchronphase schneller oder langsamer als den Motion-Master zu bewegen, siehe dazu Kapitel 6.3 „Unterschiedliche Geschwindigkeit zwischen Sägeachse und Motion-Master“ auf Seite 26.

TabPoints

Je größer die Anzahl der TabPoints, umso feiner wird die Synchronisierungsphase aufgelöst. Die Synchronisierungsphase nimmt alle Punkte der Tabelle in Anspruch mit Ausnahme des letzten Punktes, siehe Abb. 21.

Der mögliche Maximalwert ist abhängig von der Anzahl der zu konfigurierenden Tabellen. Es stehen insgesamt 4096 Tabellenelemente für alle konfigurierten Tabellen zu Verfügung, siehe dazu Tabellenbeschreibung in Dokument „jm2xx_an_jetcontrol_bi_xxxx_benutzerinformation.pdf“.

TabStartIdx

Wenn mehrere Tabellen konfiguriert werden, ist darauf zu achten, dass sich die Tabellenbereiche nicht überlappen. Nächstes freie Tabellenelement für eine weitere Tabelle: TabStartIdx der letzten Tabelle + TabPoints der letzten Tabelle.

pTmp0

Die Funktion vergleicht die Länge des Beschleunigungsweges mit der minimalen Schnittlänge, die über die Variable Tmp7 in mm bzw. ° übergeben wird. Ist der Beschleunigungsweg länger als diese minimale Schnittlänge, dann wird die Funktion mit der Fehlermeldungen `c_FS_LOADTAB_STAT_ERR_ACC_PHASE_TOO_LONG` abgebrochen. In diesem Fall ist der Betrieb der minimalen Schnittlänge nicht mit der vorgegebenen Beschleunigung bzw. der Mastergeschwindigkeit möglich. Die Beschleunigung muss erhöht oder die Mastergeschwindigkeit verringert werden. Wenn `Tmp7 = 0`, dann wird zum Vergleich die größtmögliche Schnittlänge `MaxCuttingLength` verwendet, die in der Funktion `MotionFS_Ini` angegeben wurde.

Außerdem vergleicht die Funktion die Restzeit mit einem Vorgabewert, der über die Variable Tmp6 in Sekunden übergeben wird. Die Restzeit ist die Zeit, die der Sägeachse nach dem Erreichen des Synchronpunktes noch bleibt, um den Schnitt durchzuführen und zum Startpunkt zurückzukehren. Ist die Restzeit kleiner als der Vorgabewert in Tmp6, dann gibt die Funktion die Warnung `c_FS_LOADTAB_STAT_WARN_REMAINTIME_TOO_SMALL` zurück. Die Funktion wird dennoch korrekt durchlaufen und legt eine korrekte Tabelle an. Wenn `Tmp6 = 0`, dann wird der Vergleich mit einem Standard-Vorgabewert von 1 s durchgeführt. Grundsätzlich wird beim korrekten Funktionsdurchlauf über die Variable Tmp6 die tatsächliche Restzeit zurückgegeben.

Funktionsausführung

Die Funktion kann jederzeit ausgeführt werden, nachdem die Funktion `MotionFS_Ini` für die Sägeachse ausgeführt wurde. Die dynamische Definition der Tabelle, also während dem Betrieb, kann zeitkritisch sein. Deshalb ist die Empfehlung, die Tabellen während der Initialisierungsphase des Steuerungsprogramms zu initialisieren.

Achsregistermanipulation

Folgende Achsregister werden in der Sägeachse beschrieben: Tabellenkonfiguration: R410, R411, R413, R410 R411, R413, R443, R446, R440, R441, R442.

Beispiel

Es soll eine Tabelle mit linearer Geschwindigkeitsrampe und 50 Tabellenpunkten erzeugt werden. Die Mastergeschwindigkeit beträgt ca. 500 mm/s und die Sägeachse soll mit 1000 mm/s² auf den Synchronpunkt beschleunigen. Dabei startet die Sägeachse von der Position 2500. Außerdem läuft die Sägeachse während der Synchronphase genauso schnell wie der Motion-Master. Das Übersetzungsverhältnis in der Tabelle zwischen Motion-Master und Sägeachse ist also 1:1.

Da bereits eine Tabelle mit der Konfigurationsnummer 0 mit ebenfalls 50 Tabellenpunkten existiert soll für diese Tabelle die Tabellenkonfiguration 1 verwendet werden.

Die größtmögliche Schnittlänge beträgt 5000 mm und wurde schon in der Funktion MotionFS_Ini festgelegt. Während dem Betrieb kann sich die Schnittlänge ändern. Die kleinste Schnittlänge, die vorkommen kann beträgt 3500 mm. Zur Überprüfung durch die Funktion, ob die Länge der Beschleunigungsbewegung die minimale Schnittlänge übersteigt, wird diese in Millimeter in st_MotionFS_TmpFloatVars.Tmp7 angegeben.

Für den Schnitt mit der maximalen Schnittlänge werden ca. 5 Sekunden für Schnitt und Rückfahrt (ohne Beschleunigung) benötigt. Zur Überprüfung durch die Funktion, ob diese Zeit machbar ist, wird in st_MotionFS_TmpFloatVars.Tmp6 diese Zeit in Sekunden angegeben.

```
TASK t_Init
...
//--- Festlegen der Rahmenwerte
f_SlaveStartPos:= 2500;
f_MasterSpeed:= 500;
f_SlaveAcc:= 1000;
f_TabGearRatio:=1; //--- Tabellen-Übersetzungsverhältnis Motion-Master zu Sägeachse
                    = 1:1
st_MotionFS_TmpFloatVars.Tmp6:= 5;      //--- Vorgabewert der Restzeit
st_MotionFS_TmpFloatVars.Tmp7:= 3500;   //--- Minimale Schnittlänge

//--- Ausführen der Funktion
n_Return:=MotionFS_LoadTab_Linear(&st_MotionFS_Slave0Data,
&st_MotionFS_Slave0TabData1,&f_SlaveStartPos,&f_MasterSpeed,&f_SlaveAcc,
&f_TabGearRatio,50,1,50,&st_MotionFS_TmpFloatVars0);

//--- Auswerten des Rückgabewerts
CASE n_Return OF
    c_FS_LOADTAB_STAT_WARN_REMAINTIME_TOO_SMALL:
        //--- Warnung: Restzeit ist kleiner als Vorgabewert
        ...
        BREAK;
    c_FS_LOADTAB_STAT_ERR_ACC_PHASE_TOO_LONG:
        //--- Fehler: Beschleunigungsweg ist länger als minimalste Schnittlänge
        ...
        BREAK;
END_CASE;
...
END_TASK;
```

9.8.3 MotionFS_Reset

MotionFS_Reset (pSlaveData, pTabData, Option)

Parameter

pSlaveData:	Startadresse der Struktur vom Typ MOTION_FS_SLAVE_DATA für die Sägeachse, für die neue Produktionsperiode gestartet werden soll.
-------------	--

pTabData: Startadresse der Struktur vom Typ MOTION_FS_TAB_DATA, die die Daten der Tabelle enthält, mit der die neue Produktionsperiode gestartet werden soll.

Option: Reserviert für Erweiterungen: Immer = 0.

Rückgabewert Kein Rückgabewert

Beschreibung

Initialisiert die fliegende Säge Funktionen für eine neue Produktionsperiode. Zur Erklärung der Produktionsperiode siehe Kapitel 3 „Produktionsperiode und Sägezyklus“ auf Seite 9.

Funktionsausführung



Die Funktion MUSS vor jeder neuen Produktionsperiode aufgerufen werden. Dabei muss die MOTION_FS_TAB_DATA-Struktur verwendet werden, die auch bei der nächsten Startfunktion (MotionFS_Start, MotionFS_StartCapPos, etc.) verwendet wird.

Wichtig

Achsregistermanipulation

Folgende Achsregister werden in der Sägeachse beschrieben: R101, R410, R443, R412, R448, R449.

Beispiel

```
TASK t_Run
...
//--- Starte neue Produktionsperiode
MotionFS_Reset(&st_MotionFS_Slave0Data,&st_MotionFS_Slave0TabData0,0);
...
//--- Sägezyklen durchführen
WHILE ... DO
...
//--- Starte Sägezyklus
MotionFS_Start...(...);
...
END_WHILE;
...
END_TASK;
```

9.8.4 MotionFS_Start

MotionFS_Start (pSlaveData, pTabData, pCuttingLength, Option, pTmp0)

Parameter

pSlaveData: Startadresse der Struktur vom Typ MOTION_FS_SLAVE_DATA für die Sägeachse, für die ein neuer Sägezyklus gestartet werden soll.

pTabData: Startadresse der Struktur vom Typ MOTION_FS_TAB_DATA, die die Daten der Tabelle enthält, mit der der neue Sägezyklus gestartet werden soll.

pCuttingLength:	Adresse der FLOAT-Variablen, die die Sollschnittlänge in mm bzw. ° enthält, siehe Funktionsbeschreibung. Wertebereich der Schnittlänge: 0...Tabellenpositionsbereich des Motion-Master
Option:	Reserviert für Erweiterungen: Immer = 0.
pTmp0:	Startadresse der Struktur der temporären FLOAT-Variablen vom Typ MOTION_FS_TMP_FLOATVARS.

Rückgabewert Funktionsstatus. Die möglichen Werte sind:
c_FS_START_STAT_OK
c_FS_START_STAT_ERR_MASTER_POS_TOO_FAR
Siehe auch Kapitel 9.6 "Konstanten" auf Seite 37

Beschreibung

Die Funktion startet einen neuen Sägezyklus für eine klassische fliegende Sägeachse.

Vor dem ersten Sägezyklus, der mit dieser Funktion gestartet wird, muss entweder ein Kopfschnitt mit der Funktion MotionFS_StartInstant vorausgegangen sein oder das Material mit der Sägeeinheit ausgerichtet worden sein. Ist dies nicht der Fall ist der Schnittpunkt nicht „referenziert“. Zur Ausrichtung kann die Funktion MotionFS_GetMasterAlignPos benutzt werden.

Der Rückgabewert muss bei jedem Aufruf ausgewertet werden, um festzustellen, ob ein Fehler aufgetreten ist.

Die Funktion reduziert die aktuelle Position des Motion-Master um die Schnittlänge und berechnet einen neuen Startpunkt für die Sägeachse. Falls die reduzierte Masterposition bereits den Startpunkt überschritten hat, wird die Funktion mit der Fehlermeldung c_FS_START_STAT_ERR_MASTER_POS_TOO_FAR abgebrochen. Wenn die Masterposition vor dem Startpunkt liegt, startet die Funktion die Tabelle in der Sägeachse im Modus: „Warten, bis Masterposition den Startpunkt erreicht hat“. Der Start der Sägeachse wird dann von der Tabellenfunktion im Achsverstärker durchgeführt. Die Funktion überprüft danach, ob die Sägeachse den Synchronpunkt erreicht hat.



Falls kein Fehler aufgetreten ist, wird die Funktion erst dann beendet, wenn die Sägeachse synchron mit dem Motion-Master ist.

Wichtig

pCuttingLength

Als Sollschnittlänge kann ein beliebiger Wert angegeben werden, der aber nicht größer sein darf, als die größtmögliche Schnittlänge pMaxCuttingLength, die beim Aufruf der Funktion MotionFS_Ini angegeben wurde. Die Sollschnittlänge ist bezogen auf die Positionsskalierung des Motion-Masters.

Funktionsausführung

Die Funktion muss für jeden neuen Sägezyklus aufgeführt werden.

Achsregistermanipulation

Folgende Achsregister werden in der Sägeachse beschrieben: R101, R402, R410, R443, R446.

Beispiel

```
TASK t_Run
...
//--- Sägezyklen durchführen
WHILE ... DO
...
//--- Fahre Säge zum Startpunkt
...
//--- Starte Sägezyklus
f_CuttingLength:= 5000;
MotionFS_Start(&st_MotionFS_Slave0Data,&st_MotionFS_Slave0TabData0,
               &f_CuttingLength,0,&st_MotionFS_TmpFloatVars0);
//--- Führe Schnitt aus
...
END_WHILE;
...
END_TASK;
```

9.8.5 MotionFS_StartChangeTab

MotionFS_StartChangeTab (pSlaveData, pTabDataNext, pTabDataLast, pCuttingLength, Option, pTmp0)

Parameter

pSlaveData:	Startadresse der Struktur vom Typ MOTION_FS_SLAVE_DATA für die Sägeachse, für die ein neuer Sägezyklus gestartet werden soll.
pTabDataNext:	Startadresse der Struktur vom Typ MOTION_FS_TAB_DATA, die die Daten der Tabelle enthält, mit der der neue Sägezyklus gestartet werden soll.
pTabDataLast:	Startadresse der Struktur vom Typ MOTION_FS_TAB_DATA, die die Daten der Tabelle enthält, mit der der letzte Sägezyklus gestartet wurde.
pCuttingLength:	Adresse der FLOAT-Variablen, die die Sollschnittlänge in mm bzw. ° enthält, siehe Funktionsbeschreibung. Wertebereich der Schnittlänge: 0...Tabellenpositionsbereich des Motion-Master
Option:	Reserviert für Erweiterungen: Immer = 0.
pTmp0:	Startadresse der Struktur der temporären FLOAT-Variablen vom Typ MOTION_FS_TMP_FLOATVARS.

Rückgabewert Funktionsstatus. Die möglichen Werte sind:
c_FS_START_STAT_OK
c_FS_START_STAT_ERR_MASTER_POS_TOO_FAR
Siehe auch Kapitel 9.6 "Konstanten" auf Seite 37

Beschreibung

Die Funktion startet einen neuen Sägezyklus für eine klassische fliegende Sägeachse. Dabei soll der nächste Schnitt mit einer anderen Tabelle ausgeführt werden, als der letzte.

Vor dem Sägezyklus, der mit dieser Funktion gestartet wird, wurde ein Schnitt mit der Funktion MotionFS_StartInstant oder MotionFS_Start und einer anderen Tabelle ausgeführt.

Der Rückgabewert muss bei jedem Aufruf ausgewertet werden, um festzustellen, ob ein Fehler aufgetreten ist.

Die Funktion reduziert die aktuelle Position des Motion-Master um die Schnittlänge und berechnet einen neuen Startpunkt für die Sägeachse. Falls die reduzierte Masterposition bereits den Startpunkt überschritten hat, wird die Funktion mit der Fehlermeldung `c_FS_START_STAT_ERR_MASTER_POS_TOO_FAR` abgebrochen. Wenn die Masterposition vor dem Startpunkt liegt, startet die Funktion die Tabelle in der Sägeachse im Modus: „Warten, bis Masterposition den Startpunkt erreicht hat“. Der Start der Sägeachse wird dann von der Tabellenfunktion im Achsverstärker durchgeführt. Die Funktion überprüft danach, ob die Sägeachse den Synchronpunkt erreicht hat.



Falls kein Fehler aufgetreten ist, wird die Funktion erst dann beendet, wenn die Sägeachse synchron mit dem Motion-Master ist.

Wichtig

pCuttingLength

Als Sollschnittlänge kann ein beliebiger Wert angegeben werden, der aber nicht größer sein darf, als die größtmögliche Schnittlänge `pMaxCuttingLength`, die beim Aufruf der Funktion `MotionFS_Ini` angegeben wurde. Die Sollschnittlänge ist bezogen auf die Positionsskalierung des Motion-Masters.

Funktionsausführung

Die Funktion wird für jeden neuen Sägezyklus aufgerufen, der es erfordert eine andere Tabelle als beim letzten Schnitt zu verwenden.

Achsregistermanipulation

Folgende Achsregister werden in der Sägeachse beschrieben: R101, R402, R410, R412, R443, R446.

Beispiel

Im folgenden Beispiel wird abhängig von einem Schalter mit der CASE-Anweisung der Start mit Tabelle 0 oder mit Tabelle 1 veranlasst. Bei den Übergängen von einer Tabelle zur anderen muss die Funktion `MotionFS_StartChangeTab` verwendet werden.

```
TASK t_Run
...
//--- Sägezyklen durchführen
WHILE ... DO
...
//--- Fahre Säge zum Startpunkt
...
//--- Starte Sägezyklus
    f_CuttingLength:= 5000;
//--- Auswahl der Tabelle
CASE ...OF

    c_Tab0: //--- Start mit Tab0, wenn davor schon Tab0 verwendet wurde
        MotionFS_Start(&st_MotionFS_Slave0Data,&st_MotionFS_Slave0TabData0,
            &f_CuttingLength,0,&st_MotionFS_TmpFloatVars0);

    c_Tab0ToTab1: //--- Wechsel auf Tab1, wenn davor Tab0 verwendet wurde
        MotionFS_StartChangeTab(&st_MotionFS_Slave0Data,
            &st_MotionFS_Slave0TabData1,&st_MotionFS_Slave0TabData0,
            &f_CuttingLength,0,&st_MotionFS_TmpFloatVars0);

    c_Tab1: //--- Start mit Tab1, wenn davor schon Tab1 verwendet wurde
        MotionFS_Start(&st_MotionFS_Slave0Data,&st_MotionFS_Slave0TabData1,
            &f_CuttingLength,0,&st_MotionFS_TmpFloatVars0);
```

```

c_Tab1ToTab0: //-- Wechsel auf Tab0, wenn davor Tab1 verwendet wurde
               MotionFS_StartChangeTab(&st_MotionFS_Slave0Data,
               &st_MotionFS_Slave0TabData0,&st_MotionFS_Slave0TabData1,
               &f_CuttingLength,0,&st_MotionFS_TmpFloatVars);

               END_CASE;
               //-- Führe Schnitt aus
               ...
               END_WHILE;
               ...
END_TASK;

```

9.8.6 MotionFS_StartInstant

MotionFS_StartInstant (pSlaveData, pTabData, Option, pTmp0)

Parameter

pSlaveData:	Startadresse der Struktur vom Typ MOTION_FS_SLAVE_DATA für die Sägeachse, für die ein Sofortsägezyklus gestartet werden soll.
pTabData:	Startadresse der Struktur vom Typ MOTION_FS_TAB_DATA, die die Daten der Tabelle enthält, mit der der neue Sägezyklus gestartet werden soll.
Option:	Reserviert für Erweiterungen: Immer = 0.
pTmp0:	Startadresse der Struktur der temporären FLOAT-Variablen vom Typ MOTION_FS_TMP_FLOATVARS.

Rückgabewert Funktionsstatus. Die möglichen Werte sind:
c_FS_START_STAT_OK

Beschreibung

Die Funktion startet einen neuen Sägezyklus mit einem Sofortschnitt. Sobald die Funktion aufgerufen wird, benötigt es lediglich die Durchlaufzeit der Funktion, bis der Schnitt ausgeführt wird. Siehe dazu Kapitel 9.6.1 „c_FS_STARTINSTANT_TIME_OF_PROCESS“ auf Seite 39.



Wichtig

Wenn der Wert für **c_FS_STARTINSTANT_TIME_OF_PROCESS** kleiner gewählt wird, als die tatsächliche Totzeit ist, wird der Sofortschnitt nicht sofort ausgeführt. Die Sägeachse wartet mit aktiver Tabellenfunktion solange, bis die Masterposition wieder die Startposition für die Tabelle erreicht hat. Dies passiert erst nachdem die Masterposition einen kompletten Modulodurchlauf vollzogen hat.

Da der Start über das Steuerungsprogramm ausgeführt wird, enthält die Ausführung vom Startzeitpunkt der Funktion bis zum Schnittzeitpunkt eine nicht genau bestimmbare Verzögerungszeit. Zum Ausführen von Gut-Schnitten ist diese Funktion deshalb nicht geeignet. Die Funktion wird für klassische fliegende Sägen zur Ausführung des Kopfschnitts/Anschnitt für eine neue Produktionsperiode benutzt. Da der Kopfschnitt ein Ausschussteil ist, kann für diesen Fall die entstehende Schnittgenauigkeit in Kauf genommen werden.

Nach dem Kopfschnitt ist das Material „referenziert“. Deshalb wird für die weiteren Sägezyklen die Funktion MotionFS_Start oder MotionFS_StartChangeTab benutzt.



Wichtig

Falls kein Fehler aufgetreten ist, wird die Funktion erst dann beendet, wenn die Sägeachse synchron mit dem Motion-Master ist.

Funktionsausführung

Die Funktion wird bei jedem Start einer neuen Produktionsperiode aufgerufen, wenn ein Kopfschnitt für eine klassische fliegende Säge durchgeführt werden soll.

Achsregistermanipulation

Folgende Achsregister werden in der Sägeachse beschrieben: R101, R402, R410, R443, R446.

Beispiel

```
TASK t_Run
...
//--- Starte neue Produktionsperiode
MotionFS_Reset(&st_MotionFS_Slave0Data,&st_MotionFS_Slave0TabData0,0);

//--- Zuerst Kopfschnitt durchführen
MotionFS_StartInstant(&st_MotionFS_Slave0Data,&st_MotionFS_Slave0TabData0,
0,&st_MotionFS_TmpFloatVars0);

//--- Sägezyklen durchführen
WHILE ... DO
//--- Fahre Säge zum Startpunkt
...
//--- Starte Sägezyklus
MotionFS_Start...(...);
//--- Führe Schnitt aus
...
END_WHILE;
...
END_TASK;
```

9.8.7 MotionFS_StartCapPos

MotionFS_StartCapPos (pSlaveData, pTabData, pCapPos, pPosOffset, Option, pTmp0)

Parameter

pSlaveData:	Startadresse der Struktur vom Typ MOTION_FS_SLAVE_DATA für die Sägeachse, für die der Sägezyklus gestartet werden soll.
pTabData:	Startadresse der Struktur vom Typ MOTION_FS_TAB_DATA, die die Daten der Tabelle enthält, mit der der neue Sägezyklus gestartet werden soll.
pCapPos:	Adresse der FLOAT-Variablen, die die Position in mm bzw. ° des Motion-Master beim Capturezeitpunkt beinhaltet, siehe Funktionsbeschreibung. Wertebereich der Captureposition: Masterpositionsbereich
pPosOffset:	Adresse der FLOAT-Variablen, die den Abstand des Positionsoffset in mm bzw. ° enthält, siehe Funktionsbeschreibung. Wertebereich des Offsets: Masterpositionsbereich
Option:	Reserviert für Erweiterungen: Immer = 0.
pTmp0:	Startadresse der Struktur der temporären FLOAT-Variablen vom Typ MOTION_FS_TMP_FLOATVARS.

Rückgabewert Funktionsstatus. Die möglichen Werte sind:
c_FS_START_STAT_OK

Beschreibung

Die Funktion startet einen neuen Sägezyklus dem ein Captureereignis vorweg ging, siehe Kapitel 5 „Fliegende Säge mit Hardwaresignal“ auf Seite 18.

Der Schnittpunkt wird dabei von der Funktion durch die Parameter CapPos und PosOffset berechnet. Wobei mit CapPos die Masterposition beim Captureereignis angegeben wird und PosOffset den Abstand zwischen Capturesensor und Startposition der Sägeachse definiert, siehe Abb. 22. Dabei ist zu beachten, dass der Capturesensor vor der Startposition der Sägeachse positioniert wird. Der Abstand wird dann über PosOffset als negativer Wert angegeben.

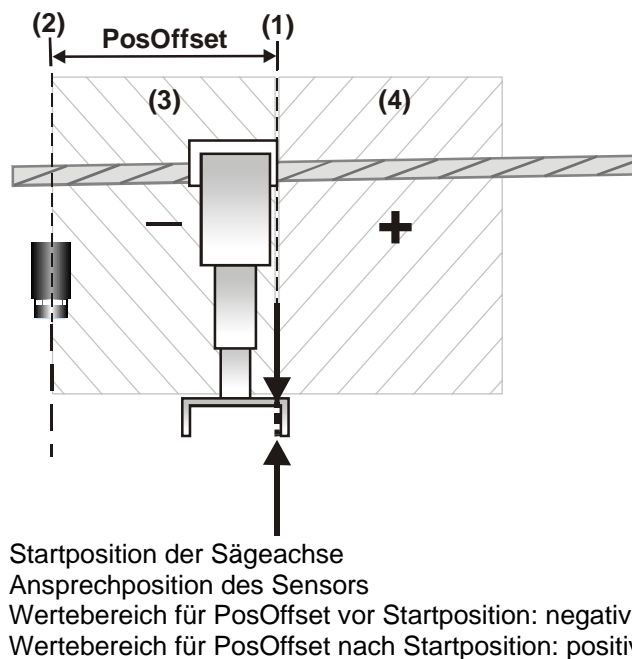


Abb. 22 Sensorabstand zur Startposition

Der Abstand zwischen Capturesensor und Startposition muss so groß sein, dass der Sägeachse nach dem Aufruf der Funktion noch genügend Zeit bleibt, sich auf den Schnittpunkt aufzusynchronisieren. Konkret heißt das, dass der Sensor mindestens etwas mehr als die Beschleunigungsstrecke der Sägeachse von der Startposition der Sägeachse entfernt sein muss.



Wichtig

Wenn der Abstand zwischen Capturesensor und Startposition der Sägeachse zu klein gewählt wird, wird der Schnitt nicht an dem gewünschten Punkt ausgeführt. Die Sägeachse wartet mit aktiver Tabellenfunktion solange, bis die Masterposition einen kompletten Modulodurchlauf vollzogen hat und beginnt dann den Sägezyklus.

Das Material wird bei jedem Zyklus durch den Capturesensor neu „referenziert“.



Wichtig

Falls kein Fehler aufgetreten ist, wird die Funktion erst dann beendet, wenn die Sägeachse synchron mit dem Motion-Master ist.

CapPos / PosOffset

Die Werte sind auf die Positionsskalierung des Motion-Masters bezogen.

Funktionsausführung

Die Funktion wird für jeden neuen Sägezyklus aufgerufen, der durch ein Hardwaresignal den Schnittpunkt definiert.

Achsregistermanipulation

Folgende Achsregister werden in der Sägeachse beschrieben: R101, R402, R410, R443, R446.

Beispiel

Im folgenden Beispiel befindet sich der Capturesensor 400 mm vor dem Startpunkt der Sägeachse.

```
VAR
    ...
    f_CapturePos:                FLOAT AT %VL 65080;
    ...
VAR_END;

TASK t_Run
    ...
    //--- Sägezyklen durchführen
    WHILE ... DO
        //--- Fahre Säge zum Startpunkt
        ...
        //--- Warte auf Captureereignis
        //    und speichere Captureposition in f_CapturePos
        f_CapturePos:= ...
        ...
        //--- Starte Sägezyklus
        MotionFS_StartCapPos(&st_MotionFS_Slave0Data,&st_MotionFS_Slave0TabData0,
                             &f_CapturePos,-400,0,&st_MotionFS_TmpFloatVars0);
        //--- Führe Schnitt aus
        ...
    END_WHILE;
    ...
END_TASK;
```

9.8.8 MotionFS_GetStartStatus

MotionFS_GetStartStatus (pSlaveData, Option)

Parameter

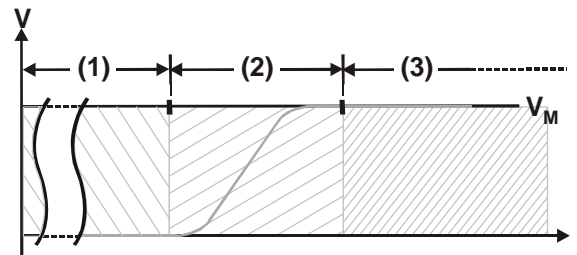
pSlaveData:	Startadresse der Struktur vom Typ MOTION_FS_SLAVE_DATA für die Sägeachse, deren Startstatus ermittelt werden soll.
Option:	Reserviert für Erweiterungen: Immer = 0.

Rückgabewert

Funktionsstatus. Die möglichen Werte sind:
 c_FS_STARTSTATUS_STAT_NO_TABLE_ACTIVE
 c_FS_STARTSTATUS_STAT_WAITING_FOR_START
 c_FS_STARTSTATUS_STAT_IN_RAMP
 c_FS_STARTSTATUS_STAT_SYNCHRON
 c_FS_STARTSTATUS_STAT_ERR_INCORRECT_STATUS
 Siehe auch Kapitel 9.6 "Konstanten" auf Seite 37

Beschreibung

Die Funktion gibt den Startstatus eines Sägezyklus zurück, der bereits mit einer der Funktionen MotionFS_Start, MotionFS_StartChangeTab, MotionFS_StartInstant oder MotionFS_StartCapPos gestartet wurde.



- (1) c_FS_STARTSTATUS_STAT_WAITING_FOR_START = Sägeachse wartet, bis Master die Startposition erreicht hat
 (2) c_FS_STARTSTATUS_STAT_IN_RAMP = Sägeachse in Rampe
 (3) c_FS_STARTSTATUS_STAT_SYNCHRON = Sägeachse ist synchron

Abb. 23 Status des Sägezyklus

Funktionsausführung

Die Funktion wird zyklisch während der Startphase aufgerufen.



Wichtig

Die Funktion gibt nur einen gültigen Status zurück, wenn sie parallel zur bereits laufenden Startfunktion (MotionFS_Start, MotionFS_StartChangeTab, MotionFS_StartInstant oder MotionFS_StartCapPos) und während der Synchronphase bis zum Ausklinken der Sägeachse, aufgerufen wird.

Achsregistermanipulation

Die Funktion beschreibt keine Achsregister in der Sägeachse.

Beispiel

Im folgenden Beispiel wird die Startfunktion im Task t_Run ausgeführt und die Statusfunktion im Task t_Diag.

```
VAR
  ...
  n_Status:                INT AT %VL 4;
  ...
VAR_END;
```



```
TASK t_Run
...
//--- Sägezyklen durchführen
WHILE ... DO
    //--- Fahre Säge zum Startpunkt
    ...
    //--- Starte Sägezyklus
    MotionFS_Start...(..);
    //--- Führe Schnitt aus
    ...
END_WHILE;
...
END_TASK;

TASK t_Diag
...
WHILE TRUE DO //--- Endlosschleife
    //--- Zeige Statusinformation
    IF ... THEN //--- Nur wenn Produktionsperiode läuft
        //--- Lese Status
        n_Status:= MotionFS_GetStartStatus(&st_MotionFS_Slave0Data, 0);

        //--- Zeige Status auf Display
        CASE n_Status OF
            c_FS_STARTSTATUS_STAT_NO_TABLE_ACTIVE:
                //--- Zeige Status = nix, Displaystellen werden gelöscht
                DISPLAY_TEXT (0, 35, '$');
                BREAK;
            c_FS_STARTSTATUS_STAT_WAITING_FOR_START:
                //--- Zeige Status = warten auf Motion-Master
                DISPLAY_TEXT (0, 48, 'W');
                BREAK;
            c_FS_STARTSTATUS_STAT_IN_RAMP:
                //--- Zeige Status = in Rampe
                DISPLAY_TEXT (0, 48, '/');
                BREAK;
            c_FS_STARTSTATUS_STAT_SYNCHRON:
                //--- Zeige Status = synchron
                DISPLAY_TEXT (0, 48, '=');
                BREAK;
        END_CASE;
    END_IF;
END_WHILE;
...
END_TASK;
```

9.8.9 MotionFS_GetActCutLength

MotionFS_GetActCutLength (pSlaveData, Direction, Option, pTmp0, pActCutLength)

Parameter

pSlaveData:	Startadresse der Struktur vom Typ MOTION_FS_SLAVE_DATA für die Sägeachse, deren aktuelle Materiallänge ermittelt werden soll.
Direction:	Schalter für die Richtung der Materiallängenzählung, siehe Funktionsbeschreibung. Wertebereich: 0;1 0 = Aufwärtszählung 1 = Abwärtszählung

Option:	Reserviert für Erweiterungen: Immer = 0.
pTmp0:	Startadresse der Struktur der temporären FLOAT-Variablen vom Typ MOTION_FS_TMP_FLOATVARS.
pActCutLength:	FLOAT-Rückgabewert: Adresse der FLOAT-Variablen, die die aktuelle Materiallänge in mm bzw. ° enthält, die von der Funktion berechnet wird.
	Wertebereich der Materiallänge: siehe unten

Rückgabewert Kein Rückgabewert der Funktion selbst, aber siehe Parameter pActCutLength

Beschreibung

Die Funktion gibt während der Startphase des Sägezyklus, für eine klassische fliegende Säge, die bereits aufgelaufene Materiallänge zurück. Die Auflaufphase ist die Zeit während die Sägeachse auf die Startposition des Motion-Masters wartet und die Synchronisationsrampe durchführt, bis sie synchron mit dem Motion-Master ist. Die Funktion wird dabei parallel zu der Startfunktion MotionFS_Start oder MotionFS_StartChangeTab in einem separaten Task ausgeführt.

Der Wertebereich des Rückgabeparameters ActCutLength ist aus den nachfolgenden Abbildungen ersichtlich.

- (CP) Schnittpunkt
- (Nominal Cutting Length) Sollschnittlänge
- (Modulo Range) Modulopositionsbereich
- (1) Auflaufphase: aufsteigend von 0 bis Sollschnittlänge
- (2) Schnittphase: absteigend von – (Modulopositionsbereich – Sollschnittlänge) bis 0

Abb. 24 Aktuelle Materiallänge – Aufwärtszählung

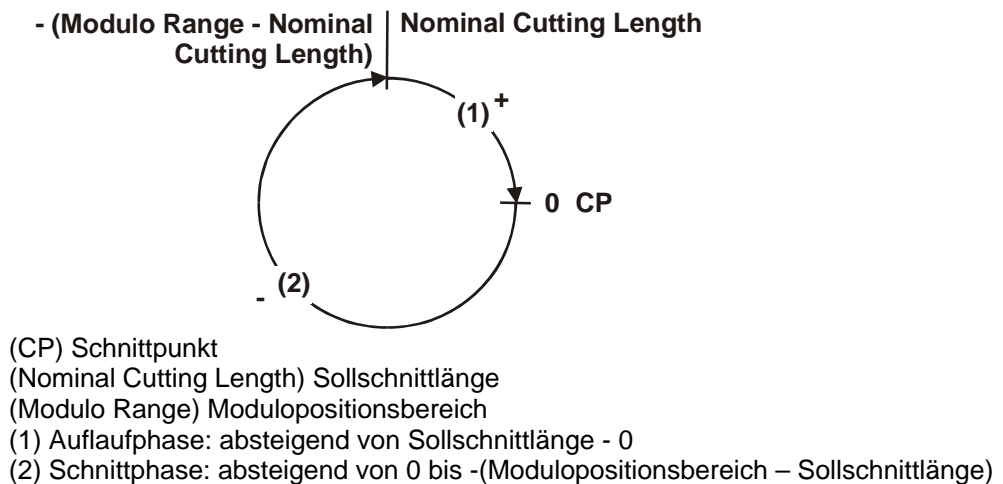


Abb. 25 Aktuelle Materiallänge – Abwärtszählung

Um ab dem Erreichen der Sollschnittlänge den Wert anzuhalten, um z.B. nur noch die Sollschnittlänge auszugeben, kann über das Anwenderprogramm der Rückgabeparameter der Funktion auf < 0 abgefragt werden. Ist dies der Fall, dann befindet sich die Sägeachse in der Schnittphase und die Sollschnittlänge ist somit erreicht. Siehe dazu das Beispiel zur Funktion.

pActCutLength

Die Materiallänge ist bezogen auf die Positionsskalierung des Motion-Masters.

Funktionsausführung

Die Funktion wird zyklisch während der Startphase aufgerufen.



Wichtig

Die Funktion gibt nur einen gültigen Wert zurück, wenn sie parallel zur bereits laufenden Startfunktion MotionFS_Start oder MotionFS_StartChangeTab und während der Synchronphase aufgerufen wird.

Der genaue Zeitraum der Funktionsausführung ist wie folgt: Zwischen gegebenen Kommando 46 durch die Startfunktion (Rückgabewert der Funktion MotionFS_GetStartStatus > c_FS_STARTSTATUS_STAT_NO _TABLE_ACTIVE) bis zum Ausklinken.

Achsregistermanipulation

Die Funktion beschreibt keine Achsregister in der Sägeachse.

Fehlersuche

Wenn keine sinnvolle Materiallänge zurückgegeben wird, kann es auch daran liegen, dass die festgelegte Sollschnittlänge mit der aktuellen Geschwindigkeit des Motion-Master und der Beschleunigung des Sägeachse nicht machbar ist.

Beispiel

Im folgenden Beispiel wird die Startfunktion im Task t_Run ausgeführt und die Funktion MotionFS_GetActCutLength im Task t_Diag. Die Materiallänge soll aufwärts gezählt werden.

```
VAR
    ...
    f_ActCutLength:    FLOAT AT %VL 65071;
    ...
VAR_END;

TASK t_Run
    ...
    //--- Sägezyklen durchführen
    WHILE ... DO
        //--- Fahre Säge zum Startpunkt
        ...
        //--- Starte Sägezyklus
        MotionFS_Start...(...,f_CuttingLength,...);
        //--- Führe Schnitt aus
        ...
    END_WHILE;
    ...
END_TASK;

TASK t_Diag
    ...
    WHILE TRUE DO    //--- Endlosschleife
        //--- Zeige aktuelle Materiallänge
        //    Aber nur wenn Tabellenfunktion aktiv
        IF MotionFS_GetStartStatus(&st_MotionFS_Slave0Data,0) >
            c_FS_STARTSTATUS_STAT_NO_TABLE_ACTIVE THEN
            MotionFS_GetActCutLength(&st_MotionFS_Slave0Data,0,0,
                &st_MotionFS_TmpFloatVars1,&f_ActCutLength);
            //--- Wenn in Schnittphase, dann wird f_ActCutLength negativ.
            //    In diesem Fall soll der Ausgabewert auf dem Display die
            //    Sollschnittlänge beinhalten. Falls die aktuelle Materiallänge >
            //    Sollschnittlänge ist, soll auch ein negativer Wert bis max. -500
            //    vorkommen dürfen.
            IF f_ActCutLength < -500 THEN
                f_ActCutLength:= f_CuttingLength;
            END_IF;
            //--- Zeige auf Display
            DISPLAY_VALUE (0, 35, f_ActCutLength);
            ...
        END_IF;
    END_WHILE;
    ...
END_TASK;
```

9.8.10 MotionFS_GetDistToCutPoint

MotionFS_GetDistToCutPoint (pSlaveData, Option, pTmp0, pDistCutPoint)

Parameter

pSlaveData:	Startadresse der Struktur vom Typ MOTION_FS_SLAVE_DATA für die Sägeachse, deren Distanz ermittelt werden soll.
Option:	Reserviert für Erweiterungen: Immer = 0.

pTmp0: Startadresse der Struktur der temporären FLOAT-Variablen vom Typ MOTION_FS_TMP_FLOATVARS.
pDistCutPoint: FLOAT-Rückgabewert: Adresse der FLOAT-Variablen, die die aktuelle Distanz in mm bzw. ° enthält, die von der Funktion berechnet wird, siehe Funktionsbeschreibung.

Wertebereich der Distanz: siehe unten

Rückgabewert Kein Rückgabewert der Funktion selbst, aber siehe Parameter pDistCutPoint

Beschreibung

Die Funktion gibt während der Startphase des Sägezyklus die aktuelle Distanz zum Schnittpunkt zurück. Die Funktion wird dabei parallel zu der aktuell verwendeten Startfunktion in einem separaten Task ausgeführt. Die Funktion kann bei allen Startfunktion (MotionFS_Start, MotionFS_StartChangeTab, MotionFS_StartInstant oder MotionFS_StartCapPos) verwendet werden.

Der Wertebereich des Rückgabeparameters DistCutPoint ist aus der nachfolgenden Abbildungen ersichtlich. Der Wertebereich liegt immer im positiven Zahlenbereich.

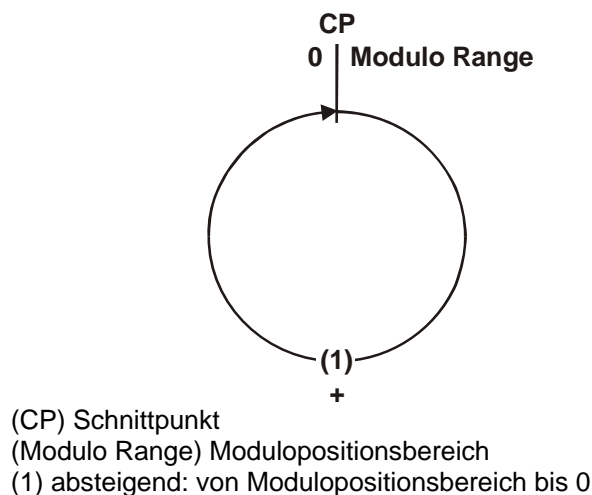


Abb. 26 Distanz zum Schnittpunkt

pDistCutPoint

Die Distanz ist bezogen auf die Positionsskalierung des Motion-Masters.

Funktionsausführung

Die Funktion wird zyklisch während der Startphase aufgerufen.



Die Funktion gibt nur einen gültigen Wert zurück, wenn sie parallel zur bereits laufenden Startfunktion und während der Synchronphase aufgerufen wird.

Wichtig Der genaue Zeitraum der Funktionsausführung ist wie folgt: Zwischen gegebenen Kommando 46 durch die Startfunktion (Rückgabewert der Funktion MotionFS_GetStartStatus > c_FS_STARTSTATUS_STAT_NO _TABLE_ACTIVE) bis zum Ausklinken.

Achsregistermanipulation

Die Funktion beschreibt keine Achsregister in der Sägeachse.

Fehlersuche

Wenn keine sinnvolle Distanz zurückgegeben wird, kann es auch daran, dass die festgelegte Sollschnittlänge mit der aktuellen Geschwindigkeit des Motion-Master und der Beschleunigung des Sägeachse nicht machbar ist.

Beispiel

Im folgenden Beispiel wird die Startfunktion im Task t_Run ausgeführt und die Funktion MotionFS_GetDistToCutPoint im Task t_Diag.

```
VAR
    ...
    f_ActCutLength:    FLOAT AT %VL 65071;
    ...
VAR_END;

TASK t_Run
    ...
    //--- Sägezyklen durchführen
    WHILE ... DO
        //--- Fahre Säge zum Startpunkt
        ...
        //--- Starte Sägezyklus
        MotionFS_Start...(...,f_CuttingLength,...);
        //--- Führe Schnitt aus
        ...
    END_WHILE;
    ...
END_TASK;

TASK t_Diag
    ...
    WHILE TRUE DO    //--- Endlosschleife
        //--- Zeige aktuelle Distanz zum Schnittpunkt
        //    Aber nur wenn Tabellenfunktion aktiv
        IF MotionFS_GetStartStatus(&st_MotionFS_Slave0Data, 0) >
            c_FS_STARTSTATUS_STAT_NO_TABLE_ACTIVE THEN
            MotionFS_GetDistToCutPoint(&st_MotionFS_Slave0Data,0,
                &st_MotionFS_TmpFloatVars1,&f_ActCutLength);
            //--- Distanz auf Display anzeigen
            DISPLAY_VALUE (0, 35, f_ActCutLength);
            ...
        END_IF;
    END_WHILE;
    ...
END_TASK;
```

9.8.11 MotionFS_GetMasterAlignPos

MotionFS_GetMasterAlignPos (pSlaveData, pTabData, Option, pTmp0, pMasterAlignPos)

Parameter

pSlaveData: Startadresse der Struktur vom Typ MOTION_FS_SLAVE_DATA für die Sägeachse, die manuell ausgerichtet werden soll.

pTabData:	Startadresse der Struktur vom Typ MOTION_FS_TAB_DATA, die die Daten der Tabelle enthält, mit der nächste Sägezyklus gestartet werden soll.
Option:	Reserviert für Erweiterungen: Immer = 0.
pTmp0:	Startadresse der Struktur der temporären FLOAT-Variablen vom Typ MOTION_FS_TMP_FLOATVARS.
pMasterAlignPos:	FLOAT-Rückgabewert: Adresse der FLOAT-Variablen, die die aktuelle Ausrichtposition in mm bzw. ° enthält, die von der Funktion berechnet wird, siehe Funktionsbeschreibung. Wertebereich der Ausrichtposition: Masterpositionsbereich

Rückgabewert Kein Rückgabewert der Funktion selbst, aber siehe Parameter pMasterAlignPos

Beschreibung

Die Funktion wird zum manuellen „referenzieren“ des Materials mit der Werkzeugeinheit verwendet. Sie gibt die Position zurück, die das Material (Motion-Master) für die richtige Platzierung des ersten Schnitts haben muss. Voraussetzung dafür ist, dass der Schnittpunkt am Material und die Startposition der Sägeachse auf gleicher Höhe liegen. Siehe dazu Abb. 11 im Kapitel 4.2“ Variante 2: Start mit Ausrichtung von Material und Werkzeugeinheit“ auf Seite 16.



Der Rückgabewert ist nur gültig, wenn die Produktionsperiode auch mit der Tabelle gestartet wird, die zur Ausführung der Funktion angegeben wurde.

Wichtig

Zum Ausrichten wird folgendermaßen vorgegangen:

- Schnittpunkt am Material wird mit Startposition auf gleiche Höhe gebracht, z.B. mit manuellem Tippbetrieb. Die nachfolgende Schnittgenauigkeit hängt unter anderem von dieser Ausrichtung ab.
- Position des Motion-Masters wird auf den Rückgabewert der Funktion gesetzt.

pMasterAlignPos

Die Ausrichtposition ist bezogen auf die Positionsskalierung des Motion-Masters.

Funktionsausführung

Die Funktion wird für eine klassische fliegende Säge vor einer Produktionsperiode ausgeführt, wenn es keinen Kopfschnitt geben soll.

Achsregistermanipulation

Die Funktion beschreibt keine Achsregister in der Sägeachse.

Beispiel

Im folgenden Beispiel wird die Produktionsperiode mit Tabelle 0 gestartet.

```

VAR
    ...
    f_AlignPos:          FLOAT    AT %VL 65078;
    ...
VAR_END;

TASK t_Run
    ...
    //-- Motion-Master auf Ausrichtungspunkt referenzieren (setzen).
    //  Zuvor ist das Material und die Werkzeugeinheit an der Startposition der
    //  Sägeachse manuell ausgerichtet worden.
    MotionFS_GetMasterAlignPos(&st_MotionFS_Slave0Data,&st_MotionFS_Slave0TabData0,
                               0,&f_AlignPos,&st_MotionFS_TmpFloatVars0);
    MotionHome ( Ax, Grundstellung setzen, f_AlignPos);

    //-- Sägezyklen durchführen
    WHILE ... DO
        //-- Fahre Säge zum Startpunkt
        ...
        //-- Starte Sägezyklus
        MotionFS_Start(...,f_CuttingLength,...);
        //-- Führe Schnitt aus
        ...
    END_WHILE;
    ...
END_TASK;

```

9.8.12 MotionFS_SetSyncCorrection

MotionFS_SetSyncCorrection (pSlaveData, pCorrection, Option)

Parameter

pSlaveData:	Startadresse der Struktur vom Typ MOTION_FS_SLAVE_DATA für die Sägeachse, die angepasst werden soll.
pCorrection:	Adresse der FLOAT-Variablen, die den Änderungswert für die Gleichlaufanpassung enthält. Wertebereich der Korrektur: FLOAT-Bereich
Option:	Reserviert für Erweiterungen: Immer = 0.

Rückgabewert Kein Rückgabewert

Beschreibung

Die Funktion passt den Gleichlauf der Sägeachse zum Motion-Master an, um den Wert der über den Parameter Correction übergeben wird. Siehe dazu 7.3 „Gleichlaufanpassung“ auf Seite 28.

Die Funktion addiert den Wert des Parameters Correction auf den Parameter SyncFactor der Struktur MOTION_FS_SLAVE_DATA, siehe Kapitel 9.5.1 „MOTION_FS_SLAVE_DATA“ auf Seite 35. Beim Aufruf einer Startfunktion wird dann der Wert des Parameters SyncFactor direkt in Achsregister R446 „Tabelle – Slaveskalierung“ geschrieben.



Die Anpassung wirkt sich erst beim nächsten Sägezyklus aus. Außerdem verschiebt sich der Schnittpunkt ab der Anpassung.

Wichtig



Die Gleichlaufenanpassung mit dieser Funktion wird nur während der Inbetriebnahme-phase angewandt, um den Korrekturfaktor zu ermitteln. Im Produktionsbetrieb wird der Gleichlauf über die Getriebefaktoren im Motion-Master und Sägeachse angepasst.

Wichtig

Funktionsausführung

Die Funktion wird während der Inbetriebnahme zur Gleichlaufenanpassung vor dem Aufrufen einer Startfunktion ausgeführt.

Achsregistermanipulation

Die Funktion beschreibt keine Achsregister in der Sägeachse.

Beispiel

Im folgenden Beispiel wird eine Gleichlaufenanpassung um einen Faktor von -0,2 durchgeführt, d.h. die Sägeachse läuft nach der Anpassung um den Faktor 0,2 langsamer als vorher.

```
VAR
    ...
    f_SyncFactorCorr:   FLOAT AT %VL 65081;
    ...
VAR_END;

TASK t_Run
    ...
    //-- Gleichlaufenanpassung um den Faktor 0,2
    IF ...Then //-- Nur wenn Anpassung erwünscht
        f_SyncFactorCorr:= -0.2;
        MotionFS_SetSyncCorrection(&st_MotionFS_Slave0Data,&f_SyncFactorCorr,0);
    END_IF;

    //-- Sägezyklen durchführen
    WHILE ... DO
        //-- Fahre Säge zum Startpunkt
        ...
        //-- Starte Sägezyklus
        MotionFS_Start(...,f_CuttingLength,...);
        //-- Führe Schnitt aus
        ...
    END_WHILE;
    ...
END_TASK;
```