

SIEMENS

Applikationsbeschreibung • 10/2016

SIMOTION Print Standard



<https://support.industry.siemens.com/cs/ww/de/view/38195516>

Gewährleistung und Haftung

Hinweis

Die Anwendungsbeispiele sind unverbindlich und erheben keinen Anspruch auf Vollständigkeit hinsichtlich Konfiguration und Ausstattung sowie jeglicher Eventualitäten. Die Anwendungsbeispiele stellen keine kundenspezifischen Lösungen dar, sondern sollen lediglich Hilfestellung bieten bei typischen Aufgabenstellungen. Sie sind für den sachgemäßen Betrieb der beschriebenen Produkte selbst verantwortlich. Diese Anwendungsbeispiele entheben Sie nicht der Verpflichtung zu sicherem Umgang bei Anwendung, Installation, Betrieb und Wartung. Durch Nutzung dieser Anwendungsbeispiele erkennen Sie an, dass wir über die beschriebene Haftungsregelung hinaus nicht für etwaige Schäden haftbar gemacht werden können. Wir behalten uns das Recht vor, Änderungen an diesen Anwendungsbeispiele jederzeit ohne Ankündigung durchzuführen. Bei Abweichungen zwischen den Vorschlägen in diesem Anwendungsbeispiel und anderen Siemens Publikationen, wie z. B. Katalogen, hat der Inhalt der anderen Dokumentation Vorrang.

Für die in diesem Dokument enthaltenen Informationen übernehmen wir keine Gewähr.

Unsere Haftung, gleich aus welchem Rechtsgrund, für durch die Verwendung der in diesem Anwendungsbeispiel beschriebenen Beispiele, Hinweise, Programme, Projektierungs- und Leistungsdaten usw. verursachte Schäden ist ausgeschlossen, soweit nicht z. B. nach dem Produkthaftungsgesetz in Fällen des Vorsatzes, der groben Fahrlässigkeit, wegen der Verletzung des Lebens, des Körpers oder der Gesundheit, wegen einer Übernahme der Garantie für die Beschaffenheit einer Sache, wegen des arglistigen Verschweigens eines Mangels oder wegen Verletzung wesentlicher Vertragspflichten zwingend gehaftet wird. Der Schadensersatz wegen Verletzung wesentlicher Vertragspflichten ist jedoch auf den vertragstypischen, vorhersehbaren Schaden begrenzt, soweit nicht Vorsatz oder grobe Fahrlässigkeit vorliegt oder wegen der Verletzung des Lebens, des Körpers oder der Gesundheit zwingend gehaftet wird. Eine Änderung der Beweislast zu Ihrem Nachteil ist hiermit nicht verbunden.

Weitergabe oder Vervielfältigung dieser Anwendungsbeispiele oder Auszüge daraus sind nicht gestattet, soweit nicht ausdrücklich von der Siemens AG zugestanden.

Securityhinweise

Siemens bietet Produkte und Lösungen mit Industrial Security-Funktionen an, die den sicheren Betrieb von Anlagen, Systemen, Maschinen und Netzwerken unterstützen. Um Anlagen, Systeme, Maschinen und Netzwerke gegen Cyber-Bedrohungen zu sichern, ist es erforderlich, ein ganzheitliches Industrial Security-Konzept zu implementieren (und kontinuierlich aufrechtzuerhalten), das dem aktuellen Stand der Technik entspricht. Die Produkte und Lösungen von Siemens formen nur einen Bestandteil eines solchen Konzepts.

Der Kunde ist dafür verantwortlich, unbefugten Zugriff auf seine Anlagen, Systeme, Maschinen und Netzwerke zu verhindern. Systeme, Maschinen und Komponenten sollten nur mit dem Unternehmensnetzwerk oder dem Internet verbunden werden, wenn und soweit dies notwendig ist und entsprechende Schutzmaßnahmen (z.B. Nutzung von Firewalls und Netzwerksegmentierung) ergriffen wurden.

Zusätzlich sollten die Empfehlungen von Siemens zu entsprechenden Schutzmaßnahmen beachtet werden. Weiterführende Informationen über Industrial Security finden Sie unter <http://www.siemens.com/industrialsecurity>.

Die Produkte und Lösungen von Siemens werden ständig weiterentwickelt, um sie noch sicherer zu machen. Siemens empfiehlt ausdrücklich, Aktualisierungen durchzuführen, sobald die entsprechenden Updates zur Verfügung stehen und immer nur die aktuellen Produktversionen zu verwenden. Die Verwendung veralteter oder nicht mehr unterstützter Versionen kann das Risiko von Cyber-Bedrohungen erhöhen.

Um stets über Produkt-Updates informiert zu sein, abonnieren Sie den Siemens Industrial Security RSS Feed unter <http://www.siemens.com/industrialsecurity>.

Inhaltsverzeichnis

Gewährleistung und Haftung	2
1 Applikationsbeschreibung	6
1.1 Allgemeine Informationen.....	6
1.1.1 System Voraussetzungen	6
1.1.2 Changelog	6
1.2 Allgemeine SIMOTION Print Standard Automatisierungslösungskonzepte	7
1.2.1 Beispiel eines Sollwert Konzeptes im „verteilten Gleichlauf“	7
1.2.2 Beispiel möglicher Masterquellen für Folgeachsen	9
1.2.3 Beispiel verschiedener Achs Betriebsarten und Master	10
1.2.4 Beispiel der Sollwertkanal Möglichkeiten jeder Folgeachse	11
2 Applikationsstruktur	13
2.1 Print Standard Basis-Applikation.....	13
2.2 Segment Applikationen	13
2.2.1 Maschinen Segment: Flexo-Druck-Maschinen	14
2.2.2 Maschinen Segment: Tief Druck-Maschinen	14
2.2.3 Maschinen Segment: Offset-Druck-Maschinen.....	14
2.2.4 Maschinen Segment: Corrugated	14
2.2.5 Maschinen Segment: Post Press Printing.....	14
2.3 Modul Applikationen	15
2.3.1 Maschinen Modul: Positionieren des Formatzylinders/Rasterwalze zum Druckzylinder (Flexo).....	15
2.4 AddOn Applikationen.....	15
2.4.1 AddOn: Register Regelung	16
2.4.2 AddOn: Materialspeicher	16
2.4.3 AddOn: Kopplung Sigate/Simolink	16
2.4.4 AddOn: Rollenträger Ab-/Aufwickler	17
2.4.5 AddOn: TM41	17
3 Integration	18
3.1 Hintergrund zur Integrations- / Modifikationsbeschreibung.....	18
3.2 Integration von Programmen/Bibliotheken/FBs in das Maschinen-Projekt.....	18
3.2.1 Vorgehensweise zur Integration.....	19
3.3 Einfügen einer virtuellen Achse ins Projekt.....	20
3.4 Einfügen einer realen Achse ins Projekt	22
3.5 Einfügen eines Addierobjekts ins Projekt.....	25
3.6 Normierung Drehzahl und Drehmoment in SIMOTION & SINAMICS	27
4 Funktionsbeschreibung	28
4.1 Programm Struktur	28
4.2 Programme und Funktionsbausteine	30
4.2.1 Überblick über Programm-Units und Programme	30
4.2.2 Überblick Funktionsbausteine	32
4.2.3 Motion Funktionsbaustein IPO-synchron aufrufen.....	33
4.3 STDC - Die Print Standard Motion Schnittstelle	34
4.4 Print Standard Variablen Strukturen	39
sAxisConfigDataType.....	39
sAxisTOConfigDataType	44
sAxisDataType	46
sAxisDataRetainType.....	49
4.5 Print Standard Achs Betriebsarten	50
4.5.1 Überblick.....	50

4.5.2	Mode 0 – COAST_STOP	53
4.5.3	Mode 1 - E_STOP	54
4.5.4	Mode 2 - MAX_TORQUE_STOP	54
4.5.5	Mode 10 - REFERENCING	54
4.5.6	Mode 20 - POSITIONING.....	60
4.5.7	Mode 30 – SPEED_POS_CTRL	62
4.5.8	Mode 31 - SPEED_SPD_CTRL	63
4.5.9	Mode 40 - WINDING	63
4.5.10	Mode 50 – CAM_MODE.....	64
4.5.11	Mode 60 – GEAR_MODE	65
4.5.12	Mode 61 – SPEED_GEAR_MODE	67
4.5.13	Mode 100 - ERROR	67
4.5.14	Mode 101 - ERROR_TO_E_STOP	67
4.5.15	Mode 102 - ERROR_TO_COAST_STOP.....	68
4.6	Schnittstellen zur Maschinenlogik (SPS) und Antrieb	69
4.6.1	Schnittstelle zu Maschinenlogik	69
4.6.2	Schnittstelle zum Antrieb (z.B. SINAMICS).....	76
4.7	„Force“-axis Funktionalität.....	77
4.7.1	Automatisches Fehler quittieren.....	78
4.8	MOP - Motorisiertes Potentiometer	78
4.9	Spezifikation von Mastern und Kurvenscheiben	79
4.10	Register Funktionalität.....	80
4.10.1	Bedienung	80
4.10.2	Beispiele für Getriebefaktor ungleich 1	83
4.10.3	Einstellung von r64SynchAxisCycle bei Getriebefaktor ungleich 1.....	85
4.11	Spezifikation von dynamischen Daten	89
4.11.1	Anpassung der dynamischen Daten bei Format-Variablen Maschinen	90
4.12	Drehmoment-Funktionen.....	91
4.12.1	Drehmoment Begrenzung	92
4.12.2	Rücklauf / Vorlauf Bewegungssperre.....	93
4.12.3	Drehmoment Zusatzsollwert.....	94
4.13	Technologie-Erweiterungen: Wickler, Zugregelung und Register Regelung	95
4.13.1	Das „Tech-FB“ – Konzept (FBLPrint_TechAxis)	95
4.13.2	Beispiel-Anwendungen mit dem Tech-FB Konzept	100
4.13.3	Integration der Zugregelung	102
4.13.4	Integration des Wicklers	106
4.14	Verteilter Gleichlauf und I-Device Kommunikation.....	110
4.14.1	Verteilter Gleichlauf	110
4.14.2	I-Device Kommunikation	111
4.15	Antrieb/SINAMICS und TO Parametrierung via Scripting.....	112
4.15.1	AxisSimulationOnOff	112
4.15.2	ChangeTechAlarmType	112
4.15.3	LoggingLibrary.....	113
4.15.4	SetEncoderInterface.....	113
4.15.5	SinamicsSettings_OFFLINE	113
4.15.6	SimotionSettings_OFFLINE	113
5	Bedienung der Beispielapplikation.....	114
5.1	Print Standard Beispielprojekt TO-Verschaltung	115
5.2	Schritt für Schritt Demonstration des Print Standard	117
5.2.1	Verantwortung des Bedieners.....	117
5.2.2	Benötigte Hardware, Software und Projekt-Dateien	117
5.2.3	“Online gehen” in “8 einfachen Schritten”	118
5.2.4	Den Motor zum Drehen bringen in “3 einfachen Schritten”.....	121
5.2.5	Hinweise zum Verwenden der IT-Diag Seite	122

5.2.6	Demonstration einiger Betriebsarten.....	124
6	Applikations-Fehler- und Warnungen	128
6.1	Print Standard Applikations-Warnungen.....	130
6.2	Print Standard Initialisierungsfehler	132
6.3	Print Standard Applikations-Fehler	132
7	Abkürzungen.....	136
8	Literaturhinweise	137
9	Ansprechpartner	138
10	Historie.....	138

1 Applikationsbeschreibung

1.1 Allgemeine Informationen

1.1.1 System Voraussetzungen

Diese Applikation wurde überarbeitet und getestet mit:

- SIMOTION SCOUT V4.4 SP1 HF 4
- SINAMICS V4.7
- Technologiepaket Cam_Ext V4.4.0.4
- SIMATIC STEP 7 V5.5 + SP4 + HF1

1.1.2 Changelog

Alle Änderungen der Bibliothek „LPrint“ sind in einem separatem Changelog Dokument „SIMOTION_Print_Standard_Changelog_de.pdf“ zusammengefasst.

Außerdem beinhaltet die Bibliothek „LPrint“ eine Quelle „aVersion“ in der ebenfalls Änderungen zwischen den Versionen aufgelistet sind.

1.2 Allgemeine SIMOTION Print Standard Automatisierungslösungskonzepte

SIMOTION Print Standard ist ein flexibles, modulares und anpassbares Achs-Betriebsarten Softwarekonzept, das viele technologische Funktionen von Druckmaschinen beinhaltet.

Der SIMOTION Print Standard wurde konzipiert um die Kern Motion Control Funktionalität von Druckmaschinen Achsen/Antrieben bereitzustellen. Das Standard Software Paket kann dabei für viele Druckmaschinen Technologien eingesetzt werden.

Optional können Software Module z.B. zur Einbindung von Wickler, Zugregelung oder Register Regelung zusätzlich zur Standard Funktion eingebunden werden.

Anwender oder spezielle Maschinenfunktionen können leicht über die existierenden Standard Schnittstellen / Konzepte integriert werden.

Sogar anspruchsvolle technologische Anforderungen, wie veränderliches Format, die Synchronisation innerhalb mehrerer Produkte im Falzbereich oder unabhängige Registerverstellungskonzepte für Farb- und Schnittregister sind konzeptionell bereits abgedeckt.

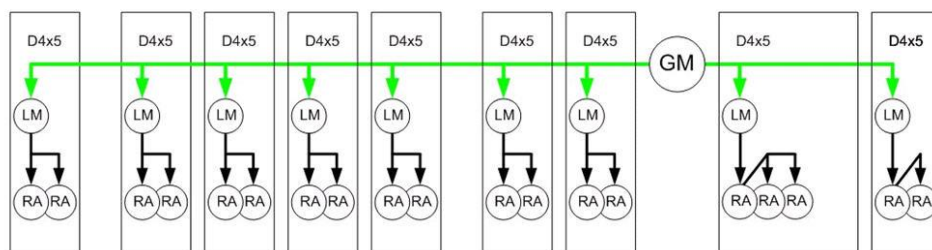
Die optimale Sollwertkanal-Struktur der Maschinen hängt stark von der Druck Technologie bzw. maschinenspezifischen Funktionen ab. Die dargestellten Sollwertkanal-Konzepte des Print Standard zeigen bereits viele Möglichkeiten auf und können leicht angepasst werden. Die verwendeten Standard Programme, Funktionsblöcke und Programmierkonzepte sind dabei unabhängig von der gewählten Sollwertstruktur.

Die Kern Funktionalität und bereitgestellten Programme sind vorbereitet, um sie ohne notwendige Veränderung einzusetzen.

1.2.1 Beispiel eines Sollwert Konzeptes im „verteilten Gleichlauf“

Ein Beispiel der Sollwertstruktur einer modularen Druckmaschine mit SIMOTION und verteiltem Gleichlauf zeigt die Abbildung 1-1.

Abbildung 1-1 Beispiel der Sollwert Verteilung (mittels SIMOTION Technologie Objekten) für das Segment einer modularen Illustrations-Offset Maschine



Ein oder mehrere virtuelle Globale Master (GM) Achsen in der Maschine sind möglich. Der globale Master ist die Sollwertquelle und könnte auch durch ein externes Gebersignal als realer Master ersetzt werden z.B. für Retrofit Anwendungen oder bei wellenlosen Erweiterungen von wellenbehafteten Maschinen.

Jedes Modul in diesem Beispiel hat eine eigene SIMOTION D CPU mit einem oder mehreren virtuellen Lokalen Master (LM) Achsen und alle notwendigen Realen

Achsen (RA), welche letztendlich einen SINAMICS Antrieb/Motor antreiben. Die RA's synchronisieren in diesem Fall immer über einen LM zum GM.

Das taktsynchrone Netzwerk für den verteilten Gleichlauf kann sowohl mit PROFIBUS als auch PROFINET IRT ausgeführt werden. Mit PROFIBUS ist ein Gleichlaufverbund nur zwischen Achsen auf dem Bus Master und Achsen auf den Bus Slaves möglich. Dadurch sind die Synchronisationskonzepte weniger flexibel und redundant verglichen mit dem Einsatz von PROFINET als Synchronisierungsnetzwerk. Zusätzlich hat PROFIBUS (wegen Beschränkungen des maximalen Datenverkehrs zwischen PROFIBUS Master und jedem PROFIBUS Slave) größere Einschränkungen bezüglich maximaler Anzahl von Stationen verglichen mit PROFINET.

Hinweis Im Fall von PROFIBUS muss der globale Master in der Synchronisation Bus Master Station sein.

Im Fall von verteilten Sollwertstrukturen mit PROFINET, sind Stern und Linien Topologien der PROFINET Verbindung möglich. Zudem können Achsen zwischen PROFINET Slave Stationen synchronisiert werden.

Hinweis Der SIMOTION Controller der als "sync-master" im Netzwerk agiert muss bei PROFINET nicht zwingend die virtuellen Leitwerte der Folgeachsen berechnen.

Das SIMOTION System berechnet und kompensiert die Übertragungstotzeiten automatisch. Diese Funktion ist besonders dann notwendig wenn bei verteiltem Gleichlauf mehr als eine SIMOTION in der Maschine synchronisiert werden.

Der/die Globale(n) Master (GM) repräsentiert den/die elektrischen Wellen der Gesamtmaschine während der/die Lokalen Master (LM) für die folgenden Einsatzfälle genutzt werden können:

- Abbild des Globalen Masters innerhalb jeder Slave Station, damit der Globale Master in jeder Station verfügbar ist. Dadurch wird das Nutz-Datenaufkommen auf dem Synchronisierungsbus (PROFIBUS oder PROFINET) minimiert.
- Der Lokale Master kann mit eigenem Drehzahlsollwert autark betrieben werden. Damit wird eine Möglichkeit geschaffen alle Realen Achsen die dem LM folgen im Gruppenbetrieb synchronisiert zu betreiben.
- Der Lokale Master kann sein Register/Phasenlage zum Globalen Master ändern. Dadurch ist eine gute Möglichkeit gegeben alle Realen Achsen die dem Lokalen Master folgen, zusätzlich zur aktuellen Maschinengeschwindigkeit, synchronisiert zu verschieben. Dies bietet sich z.B. an als Lösung für das Schnittregister bei Rollendruckmaschinen mit Querschneidern/Falzapparaten.

Falls der Lokale Master nur als „Kopie“ des Globalen Master eingesetzt wird ist keine Schnittstelle zum SPS Programm notwendig. Es besteht dann die Möglichkeit innerhalb des Print Standard Motion Programms den Lokalen Master automatisch immer auf absolute Synchronisierung (= identische Kopie) zum Globalen Master zu „forcen“.

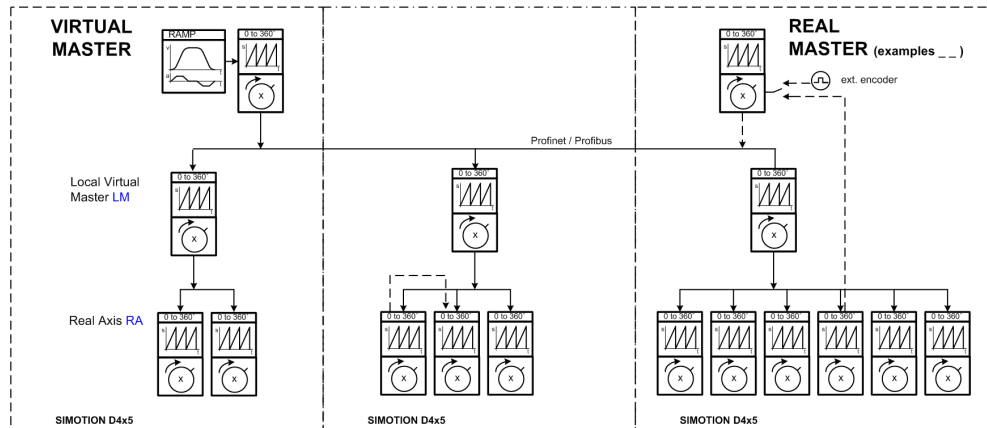
Falls der LM für Lokal Betriebsarten wie "Tippen" oder Drehzahlbetrieb von Realen Achsen genutzt werden soll, wird die gleiche Schnittstelle vom SPS Programm verwendet wie bei allen anderen Achsen.

Abhängig von der Maschinenteknologie kann es mehr als nur einen Globalen und damit auch Lokale Master geben.

Das vorgestellte Print Standard Achs Konzept bezüglich Anzahl der TO's (GM, LM etc.) wird meist abhängig von der zu realisierenden Maschinen Technologie / Segment erweitert oder minimiert. Dabei haben auch die SIMOTION CPU Leistung und die gewählten Abtastzeiten (DP, Servo, IPO) Einfluss.

1.2.2 Beispiel möglicher Masterquellen für Folgeachsen

Abbildung 1-2 Beispiel mögliche Master: Virtueller Master, Realer Master (Geber oder andere reale Achse)



Alle Achsen vom Typ "Folgeachsen" (virtuell oder real) haben die Möglichkeit zwischen mehreren (offline konfigurierten) möglichen Mastern umzuschalten.

Master für eine Achse kann auch eine andere reale Achse sein (bei PROFIBUS Netzwerk nur innerhalb der Station oder zwischen PROFIBUS Master und den anderen Stationen). Dies bietet zusätzliche Flexibilität um die optimale Sollwertverteilung für die Anforderungen der verschiedenen Maschinen Segmente zu finden.

Die Abbildung 1-2 zeigt wie eine Reale Achse (RA) abhängig von dem Maschinenstatus zwischen verschiedenen Mastern umschalten kann. z.B.

- Einer virtuellen Achse/Master folgen (z.B. LM folgt GM)
- Einer anderen realen Achse folgen (dabei kann gewählt werden ob dem Sollwert oder Istwert der Realen Achse gefolgt wird)
- Den Sollwerten erzeugt von einem externen Geber folgen

Hinweis

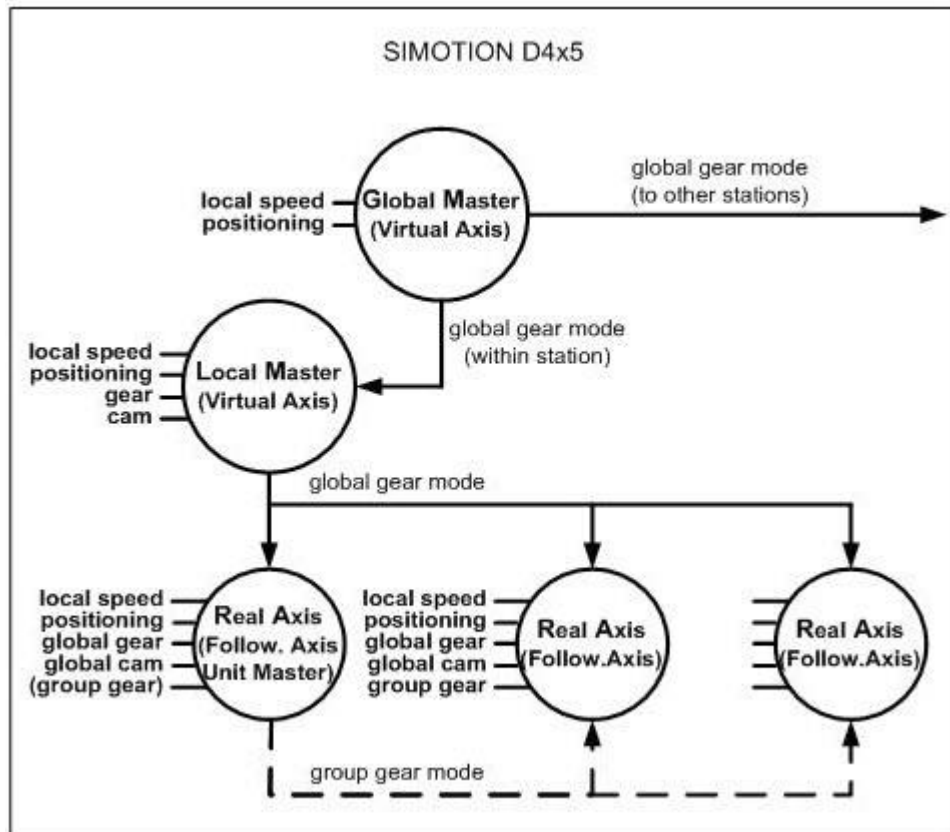
Eine ausführliche Beschreibung zur Projektierung des verteilten Getriebegleichlaufs und des Controller/Controller Querverkehr bietet der FAQ:

Projektierung einer PROFINET-IRT-Kommunikation mit verteiltem Gleichlauf (SIOS ID: 68014264)

1.2.3 Beispiel verschiedener Achs Betriebsarten und Master

Die Abbildung 1-3 zeigt einige Möglichkeiten bezüglich Achs Betriebsarten und Master Folgebetrieb auf. Als Beispiel wird nur die SIMOTION Station mit dem Globalen Leitwert GM gezeigt, sinngemäß gilt allerdings das gleiche auch für alle evtl. vorhandenen Slave Stationen bei verteilten Gleichlauf Konzepten.

Abbildung 1-3 Beispiel von Betriebsmodi und Synchronisiermodi zwischen mehreren Achsen



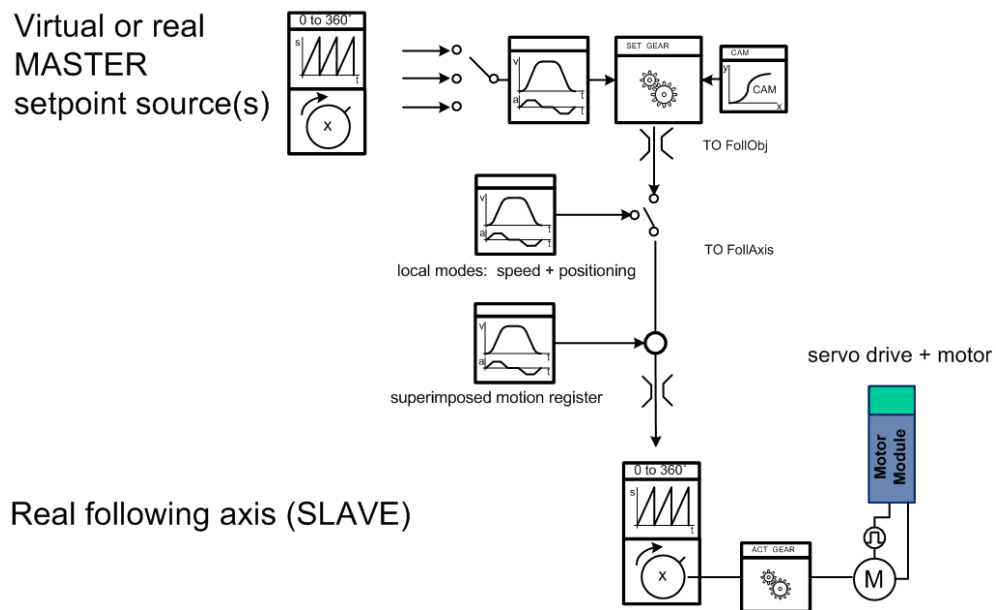
Abhängig vom Maschinen Betriebszustand können die Achsen viele verschiedene Betriebsarten einnehmen und dabei verschiedenen Mastern folgen.

Einige typische Beispiele:

- Während der Produktion der Maschine folgen normalerweise alle RA's den LM's und alle LM's ihrem GM. Während des Druckbetriebs können nun alle formathaltigen RA's z.B. Druckzylinder, ihr Register/Phasenlage zum LM und damit GM verändern.
- Der modulare Gruppenbetrieb
 - Der lokale Master kann in Drehzahlbetrieb (z.B. „SPEED_SPD_CTRL“) gefahren werden, wobei alle RA's in „GEAR_MODE“ dem LM folgen. Damit kann eine Gruppe von Achsen synchronisiert betrieben werden.
 - Eine reale Achse wird als Gruppenmaster in Drehzahlbetrieb gefahren und andere RA's folgen diesem realen „Gruppenmaster“.
- In den Lokal Betriebsarten (z.B. Drehzahl oder Positionier-Betriebsarten) kann jede Achse eigenständige Bewegungen ausführen unabhängig von allen anderen Achsen. Dies wird beispielsweise genutzt beim Einrichten der Maschine im Drehzahl Tipp Betrieb oder beim Positionieren der Plattenzylinder während des Druckplatten Wechsels.

1.2.4 Beispiel der Sollwertkanal Möglichkeiten jeder Folgeachse

Abbildung 1-4 Grundlegende Möglichkeiten im Sollwertkanal einer Folgeachse (virtuell oder real)



Die Abbildung zeigt die prinzipiellen Möglichkeiten im Sollwertkanal einer Folgeachse. Jede Folgeachse, real oder virtuell, hat diese Funktionalität.

Eine Folgeachse in SIMOTION besteht aus zwei Teilen:

- Das „Basis TO“ (TO-Achse) beinhaltet die Funktionalität der lokalen Betriebsarten wie Drehzahl oder Positionier Betrieb. Dies ist die gleiche Funktionalität wie bei einer Achse vom Typ „POS“.

- Das zusätzliche TO "Following Object" erweitert die notwendigen Funktionen für elektronisches Getriebe, Kurvenscheibe und Umschaltung zwischen mehreren möglichen Master Quellen.

Zu der Hauptbewegung "basic motion" (typischerweise die Maschinengeschwindigkeit Drehzahl/Lage vom GM) kann eine zweite Bewegung "superimposed motion profile" unabhängig überlagert werden. Dies kann z.B. zur Implementierung der Registerverstellung genutzt werden.

Hinweis

Beispiele zur technologischen Erweiterung des Sollwertkanals (Wickler, Zugregelung) sind im Kapitel [4.13.2](#) zu finden.

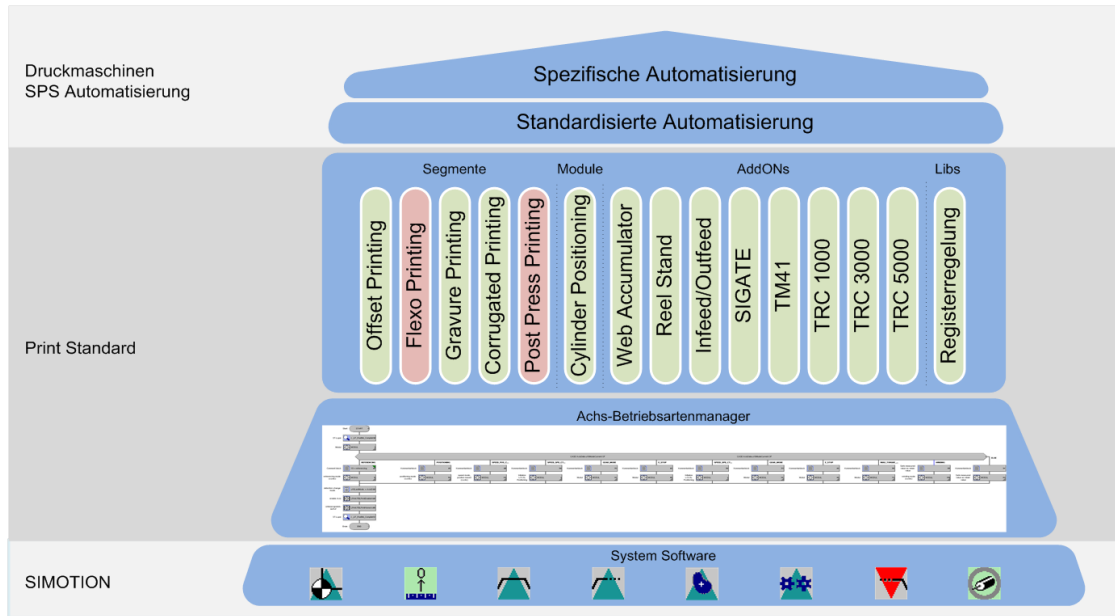
Das Sollwertkonzept des Beispielprojekts ist im Kapitel [5.1 Print Standard Beispielprojekt TO-Verschaltung](#) erläutert.

2 Applikationsstruktur

Diese Print Standard Dokumentation und das Demoprojekt sind die Grundlage für alle Segment-, Modul- und AddOn Applikationen. Im Folgenden werden die verschiedenen Applikationen kurz beschrieben. Detaillierte Informationen ist den Dokumentationen und den Beispielprojekten der jeweiligen Applikation zu entnehmen.

Die Abbildung 2-1 zeigt die Grundlegende Gesamtstruktur der Print Standard Applikation.

Abbildung 2-1 Print Standard Applikationsstruktur



2.1 Print Standard Basis-Applikation

Die Print Standard Basis Applikation bietet einen umfassenden und tiefgehenden Überblick bzgl. der unterschiedlichen Sollwertkonzepte für verschiedene Anwendungen und der Funktionalität die die Applikation liefert. Die unterschiedlichen Funktionen, Betriebsarten und das dazugehörige Softwarekonzept wird von Grund auf erklärt.

2.2 Segment Applikationen

Die Segment Applikationen sind speziell auf einen bestimmten Maschinentyp zugeschnittene Abwandlungen des Basiskonzeptes. Dadurch wird der Nutzer nicht mit Informationen die nicht seine Maschine betreffen verwirrt/gestört. Zudem kann die Segment Beschreibung in der Sprache der Maschinenklasse mehr aus Anwender Sicht verfasst werden.

Alle hier angesprochenen Segment Applikationen können aus dem Print Standard Master erzeugt werden.

2.2.1 Maschinen Segment: Flexo-Druck-Maschinen

In Flexo-Druckmaschinen werden viele Konzepte des Print Standard Master verwendet. z.B. Wickler, Zugregelung, Master für „Bahn“ und „Format“ etc.

Für dieses Segment gibt es eine Modul Beschreibung „Cylinder Positioning for Flexographic press“ speziell für die Automatisierung der Anilox/Plattenzylinder Positionierung für Zentraltrommelmaschinen (siehe Kapitel [2.3.1](#)).

2.2.2 Maschinen Segment: Tief Druck-Maschinen

In Tiefdruck-Maschinen werden viele Konzepte des Print Standard Master verwendet. z.B. Wickler, Zugregelung, Master für „Bahn“ und „Format“ etc.

Ein Beispielprojekt inkl. Dokumentation ist verfügbar:

- SIMOTION Print Standard for GRAVURE Printing (SIOS ID: 51345318)
<http://support.automation.siemens.com/WW/view/de/51345318>

2.2.3 Maschinen Segment: Offset-Druck-Maschinen

Für die meisten Offset Druckmaschinen kann die Komplexität des Print Standard Master Projekts minimiert werden. Nicht relevante Konzepte/Programmteile können entfernt werden.

Oft können die „kleinen“ TO's (Addier- und Formelobjekte) entfernt werden, da im Segment Illustrations-Offset meist einfachere Sollwert-Strukturen basierend auf einem oder zwei GM's mit verteilten LM's genügen. Beim Zeitungs-Offset werden oft LM's für jede Papierbahn erzeugt. Der Fokus liegt mehr auf höchste Redundanz und Modularität.

Ein Beispielprojekt inkl. Dokumentation ist verfügbar:

- SIMOTION Print Standard für OFFSET Druck (SIOS ID: 48946827)
<http://support.automation.siemens.com/WW/view/de/48946827>

2.2.4 Maschinen Segment: Corrugated

Für Wellpappe Druckmaschinen werden die Basis-Druckstandardfunktionen verwendet. Die erforderlichen Technologien für die speziellen Aggregate werden durch eine eigene Bibliothek und eigene Quellen realisiert.

Ein Beispielprojekt inkl. Dokumentation ist verfügbar:

- SIMOTION Print Standard für Wellpappe Druck (SIOS ID: 103498563)
<http://support.automation.siemens.com/WW/view/de/103498563>

2.2.5 Maschinen Segment: Post Press Printing

Dieses Maschinen-Segment ist spezialisiert auf Drucknachbearbeitung wie z.B. Stanzen.

Eine Dokumentation dieses Segments mit Beispielprojekt ist derzeit nicht verfügbar.

2.3 Modul Applikationen

Als Modul wird eine abgrenzbare optionale technologische Funktion verstanden meist unabhängig von den Maschinensegmenten (z.B. Wickler).

Einige der Modul Applikationen sind in ihrer Funktion und Implementierung bereits im Print Standard Master eingebunden. Die ausführliche Dokumentation der einzelnen Module ist in den separaten Modul-Applikationen enthalten.

2.3.1 Maschinen Modul: Positionieren des Formatzylinders/Rasterwalze zum Druckzylinder (Flexo)

Das Modul bietet eine maßgeschneiderte Lösung für die Positioniertechnologie der Formatzylinder/Rasterwalze bei Zentraltrommel-Flexo-Maschinen.

In diesen Maschinen werden oft die Anilox- und Plattenzylinder via Spindel oder Linearmotoren zusammen verfahren. Ein spezieller FB ist verfügbar zur Automatisierung dieser 4 Positionierachsen (auf beiden Seiten beider Zylinder) pro Druck-Deck.

Ein Beispielprojekt inkl. Dokumentation ist verfügbar:

- SIMOTION Zylinder Positionierung CI Flexo (SIOS ID: 48812779)
<http://support.automation.siemens.com/WW/view/de/48812779>

2.4 AddOn Applikationen

Als AddOn wird eine Erweiterung der Print Standard Basis Applikation verstanden. Die Erweiterung wird in Form einer zusätzlichen Bibliothek und zusätzlicher Programm-Units in ein Print Standard Basis Projekt integriert. Die AddOn Beispielapplikationen demonstrieren das Zusammenwirken mit dem Druck Standard.

2.4.1 AddOn: Register Regelung

Die Register Regelung ist ein optionales AddOn mit dem (in SIMOTION integriert) das Längs- und Seitenregister von Druckmaschinen geregelt werden kann.

Die Registerregelung findet Anwendung im Farbdruck, um die verschiedenen Farben der Druckwerke zueinander zu regeln. Auch für Stanzen o.ä. kann eine Schnittregisterregelung eingesetzt werden.

Die genaue Beschreibung dieses AddOns ist in gesonderten Beispielprojekten und Dokumentationen verfügbar:

- SIMOTION Print Standard Add-On Register Control TRC1000 (SIOS ID: 59753224)
<http://support.automation.siemens.com/WW/view/de/59753224>
- SIMOTION Print Standard Add-On Register Control TRC3000 (SIOS ID: 64896729)
<http://support.automation.siemens.com/WW/view/de/64896729>
- SIMOTION Print Standard Add-On Register Control TRC5000 (SIOS ID: 48956344)
<http://support.automation.siemens.com/WW/view/de/48956344>

2.4.2 AddOn: Materialspeicher

Soll z.B. ein Rollenwechsel an einer Anlage ohne fliegende Splice-Funktionalität am Wickler durchgeführt werden ohne die Anlage komplett anzuhalten, so ist dies nur durch befüllen bzw. entladen einer mechanischen Materialspeichereinrichtung möglich.

Zur Realisierung eines Materialspeichers ist ein separates Dokument mit Beispielprojekt verfügbar:

- SIMOTION Print Standard Add-On Web Accumulator (SIOS ID: 48947084)
<http://support.automation.siemens.com/WW/view/de/48947084>

2.4.3 AddOn: Kopplung Sigate/Simolink

Um z.B. bei Retrofits eine Master-Kopplung zwischen Anlagen mit SIMOLINK (Masterdrives) und PROFINET (SIMOTION/SINAMICS) zu ermöglichen ist eine Kopplungsbaugruppe „SIGATE“ erhältlich.

Beispielhaft ist der Einsatz des Moduls in einem Beispielprojekt inkl. Dokumentation beschrieben:

- SIMOTION Print Standard Add-On Sigate (SIOS ID: 48946460)
<http://support.automation.siemens.com/WW/view/de/48946460>

2.4.4 AddOn: Rollenträger Ab-/Aufwickler

Zusätzliche Programmteile zur Integration der Wickler Funktionalität sind im Print Standard Master Projekt bereits integriert. Der Hauptsollwert Kanal für die Wickler Beispiel Achse wurde um das TO „Formelobjekt“ erweitert.

Die eigentlichen Funktionalitäten des Wicklers (Zugregelung, Durchmesserrechner etc.) ist in einem separaten Funktionsbaustein, dem „FBWinder“ enthalten. Hierzu ist eine ausführliche Dokumentation verfügbar:

- SIMOTION Winder (SIOS ID: 35818687)

<http://support.automation.siemens.com/WW/view/de/35818687>

Unter dem Namen „Print_Standard_Addon_Reel_Stand_V22“ ist ein eigenständiges Dokument mit Software Projektbeispiel (lauffähig an einem Wickler Demonstrations-Schaltschrank) erhältlich, dieses Projekt zeigt wie der „FBWinder“ in den Print Standard eingebunden werden kann.

- SIMOTION Print Standard Add-On Rollenwechsler (SIOS ID: 48947256)

<http://support.automation.siemens.com/WW/view/de/48947256>

2.4.5 AddOn: TM41

Das Print Standard AddOn TM41 beschreibt in einem Beispielprojekt inkl. Dokumentation die Integration und Verwendung eines Terminal Modules TM41 innerhalb des Druck Standards.

Mit dem AddOn TM41 kann eine Master Position (Sollwert) als inkrementelles Gebersignal an einen anderen Controller oder an ein Registerregelungssystem übergeben werden.

- SIMOTION Print Standard Add-On TM41 (SIOS ID: 48956841)

<http://support.automation.siemens.com/WW/view/de/48956841>

3 Integration

3.1 Hintergrund zur Integrations- / Modifikationsbeschreibung

Das vielfältige Siemens TIA und insbesondere SIMOTION/SINAMICS Hardware und Software Produkt Portfolio unterstützt ein sehr weites Feld möglicher Maschinen Architekturen.

Maschinen unterscheiden sich im Konzept durch separate oder integrierte Hardware für Motion, SPS und HMI Level. Unterschiedliche Netzwerkmethoden werden benutzt. Einige Maschinen bevorzugen zentrale andere hoch modulare, dezentrale Maschinen Konzepte.

Aus diesen Gründen, nachdem die Maschinenarchitektur feststeht, macht es Sinn zunächst im Scout ein neues Projekt anzulegen mit den folgenden Schritten: (Diese Schritte sind in der allgemeinen Produktliteratur beschrieben)

- Erzeugen des NetPro/HW-Konfig Level, wenn nötig über mehrere SIMOTION Stationen.
- Die Print Standard Vorbelegung für alle takt synchronen Zeitscheiben ist 3ms: d.h. DP / Servo / IPO / (PN sync.) = 3ms / 3ms / 3ms / 3ms
- Anlegen der gesamten SINAMICS Hardware (mithilfe der SINAMICS Wizards).
- Anlegen der gesamten realen SIMOTION Achsen im SIMOTION Projekt.

Das folgende Kapitel beschreibt wie das Print Standard Demo Koffer Projekt / Konzept in ein Maschinenprojekt integriert werden kann.

Die prinzipiellen Schritte sind die folgenden:

- Die zusätzlichen virtuellen Technologie Objekte (TO's) (VA's, Addier-/ Formel Objekt) werden je nach Maschinentechnologie eingefügt und verbunden.
- Die Bibliotheken, Programme und FB's werden aus dem Print Standard Projekt kopiert.
- Die Print Standard Programme und Variablen Strukturen werden an Notwendigkeit der Maschine konfiguriert/angepasst.

3.2 Integration von Programmen/Bibliotheken/FBs in das Maschinen-Projekt

Seit Print Standard Version V3.0.0 hat sich die Programm-Struktur (Programm-Units) im Beispielprojekt geändert.

In der „alten“ Programm-Struktur gibt es eine Unit mit allen globalen Strukturvariablen und FB-Instanzen. Außerdem wird für jeden Druck Standard Funktionsbaustein-Typ ein Programm angelegt, in dem die entsprechenden FB's für alle Achsen aufgerufen werden.

Natürlich kann diese Variante weiterhin verwendet werden!

Die „neue“ Programm-Struktur stellt eine alternative dazu dar. Hier wird pro Achse oder ggf. auch pro Druckwerk eine Programm-Unit angelegt. In dieser Unit befinden sich alle globalen Strukturvariablen dieser Achse. Außerdem wird

beinhaltet die Unit für jede Task des SIMOTION Ablaufsystems ein Programm, in dem die entsprechenden Druck Standard FB's nur für diese Achse aufgerufen werden.

Im Beispielprojekt sind beide Steuerungen (Master und Slave) nach dieser Struktur aufgebaut.

3.2.1 Vorgehensweise zur Integration

- Die Bibliotheken LPrint und LConLib vom Print Standard Beispielprojekt in das Maschinenprojekt kopieren.
- Die Units dHMI, dMGlobal, pFault, pMShtdwn, pMStartup in das Maschinenprojekt kopieren.
- Eine der Achs-Units kopieren (diese kann dann beliebig vervielfacht werden)
- Mit der Suchen & Ersetzen Funktion die Achs-Namen auf die neuen Namen anpassen.
- Programme in das SIMOTION Ablaufsystem einhängen.
- Skripte und Watchtabellen in das Maschinenprojekt kopieren und anpassen.
- Kompilieren, den Compiler Meldungen folgen und ggf. die gemeldeten Probleme beheben.
- Nach erfolgreicher Kompilierung eine "Konsistenz Prüfung" ausführen und ggf. letzte Probleme beheben.

Hinweis

Aufgrund unterschiedlicher Typ-Namen muss ausgewählt werden ob die „alte“ oder „neue“ Converting Bibliothek verwendet wird. Über eine Prä-Processor Anweisung in LPrint\dTechData kann gewählt werden:

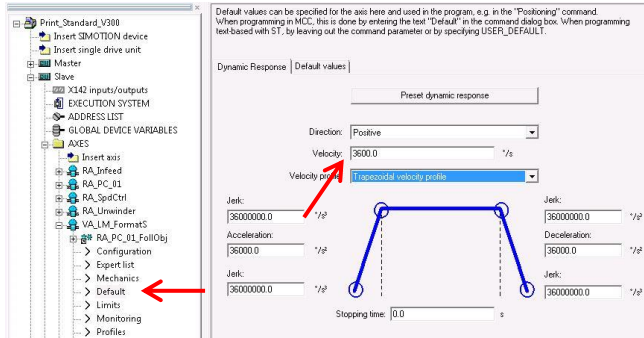
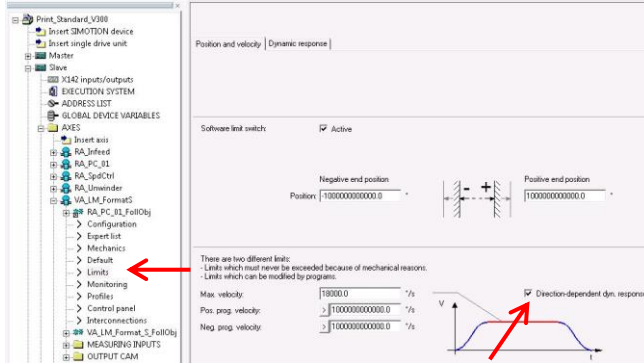
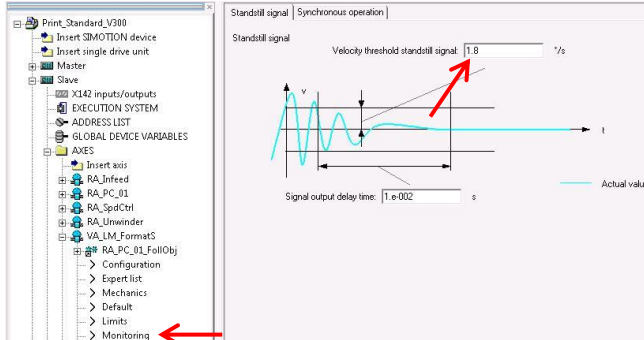
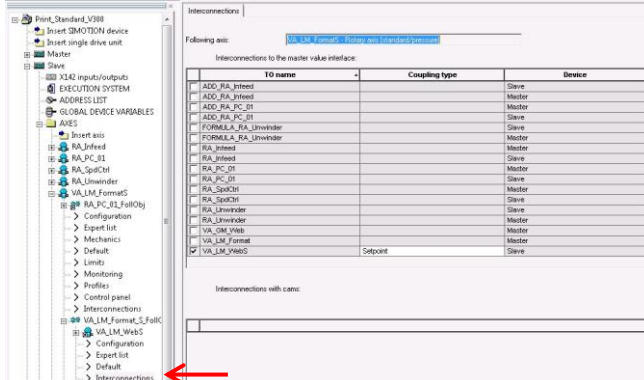
```
#define LConLib_1xx    für LConLib Version <= 1.3.0  
#define LConLib_2xx    für LConLib Version >= 2.0.0
```

3.3 Einfügen einer virtuellen Achse ins Projekt

Tabelle 3-1 Einfügen einer virtuellen Achse

Step	Action	Picture																																																																
1	<p>Neue SIMOTION Achse mit dem "Insert axis" Wizard mit folgenden Werten einfügen:</p> <ul style="list-style-type: none"> Achs-Technologie: Drehzahl-, Positionier- oder Gleichlaufachse Achstyp wählen: rotatorisch virtuell Mögliche Master am Gleichlaufobjekt (wenn vorhanden) vorbelegen 																																																																	
2	<p>In der Achs-Konfiguration im Tab „Einheiten“ die Inkremente/Position auf Maximalwert (1.000.000) einstellen</p>	<table border="1"> <thead> <tr> <th>Physical quantity</th> <th>Unit</th> </tr> </thead> <tbody> <tr><td>Increments/position</td><td>1000000Unit</td></tr> <tr><td>Ratio</td><td>%</td></tr> <tr><td>Time</td><td>s</td></tr> <tr><td>Speed</td><td>rpm</td></tr> <tr><td>Acceleration per revolution</td><td>1/(min²)</td></tr> <tr><td>Jerk per revolution</td><td>1/(min³)</td></tr> <tr><td>Voltage</td><td>V</td></tr> <tr><td>Frequency</td><td>Hz</td></tr> <tr><td>Torque</td><td>Nm</td></tr> <tr><td>Force</td><td>N</td></tr> <tr><td>Pressure</td><td>Pa</td></tr> <tr><td>Pressure change</td><td>Pa/s</td></tr> <tr><td>Force change</td><td>N/s</td></tr> <tr><td>Velocity controller integrator gain</td><td>1/s</td></tr> <tr><td>Velocity controller differentiation gain</td><td>s</td></tr> <tr><td>Percentage change</td><td>%/s</td></tr> <tr><td>Force controller P gain</td><td>1/(N*min)</td></tr> <tr><td>Pressure controller P gain</td><td>1/(Pa*min)</td></tr> <tr><td>Force controller I gain</td><td>1/(N*s*min)</td></tr> <tr><td>Pressure controller I gain</td><td>1/(Pa*s*min)</td></tr> <tr><td>Force controller D gain</td><td>s/(N*min)</td></tr> <tr><td>Pressure controller D gain</td><td>s/(Pa*min)</td></tr> <tr><td>Current</td><td>A</td></tr> <tr><td>Active power</td><td>W</td></tr> <tr><td>Temperature</td><td>°C</td></tr> <tr><td>Position</td><td>mm</td></tr> <tr><td>Velocity</td><td>mm/s</td></tr> <tr><td>Acceleration</td><td>mm/s²</td></tr> <tr><td>Jerk</td><td>mm/s³</td></tr> <tr><td>Leadscrew pitch</td><td>mm/rot</td></tr> <tr><td>Angle</td><td>°</td></tr> </tbody> </table>	Physical quantity	Unit	Increments/position	1000000Unit	Ratio	%	Time	s	Speed	rpm	Acceleration per revolution	1/(min ²)	Jerk per revolution	1/(min ³)	Voltage	V	Frequency	Hz	Torque	Nm	Force	N	Pressure	Pa	Pressure change	Pa/s	Force change	N/s	Velocity controller integrator gain	1/s	Velocity controller differentiation gain	s	Percentage change	%/s	Force controller P gain	1/(N*min)	Pressure controller P gain	1/(Pa*min)	Force controller I gain	1/(N*s*min)	Pressure controller I gain	1/(Pa*s*min)	Force controller D gain	s/(N*min)	Pressure controller D gain	s/(Pa*min)	Current	A	Active power	W	Temperature	°C	Position	mm	Velocity	mm/s	Acceleration	mm/s ²	Jerk	mm/s ³	Leadscrew pitch	mm/rot	Angle	°
Physical quantity	Unit																																																																	
Increments/position	1000000Unit																																																																	
Ratio	%																																																																	
Time	s																																																																	
Speed	rpm																																																																	
Acceleration per revolution	1/(min ²)																																																																	
Jerk per revolution	1/(min ³)																																																																	
Voltage	V																																																																	
Frequency	Hz																																																																	
Torque	Nm																																																																	
Force	N																																																																	
Pressure	Pa																																																																	
Pressure change	Pa/s																																																																	
Force change	N/s																																																																	
Velocity controller integrator gain	1/s																																																																	
Velocity controller differentiation gain	s																																																																	
Percentage change	%/s																																																																	
Force controller P gain	1/(N*min)																																																																	
Pressure controller P gain	1/(Pa*min)																																																																	
Force controller I gain	1/(N*s*min)																																																																	
Pressure controller I gain	1/(Pa*s*min)																																																																	
Force controller D gain	s/(N*min)																																																																	
Pressure controller D gain	s/(Pa*min)																																																																	
Current	A																																																																	
Active power	W																																																																	
Temperature	°C																																																																	
Position	mm																																																																	
Velocity	mm/s																																																																	
Acceleration	mm/s ²																																																																	
Jerk	mm/s ³																																																																	
Leadscrew pitch	mm/rot																																																																	
Angle	°																																																																	
3	<p>Im Tab „Mechanik“ die Modulo-Achse aktivieren und die Modulo-Länge (z.B. 360°) eingeben.</p> <p>Die Länge muss mit der Modulo-Länge der anderen TO's übereinstimmen.</p>																																																																	

3 Integration

Step	Action	Picture																																																
4	<p>Im Tab „Vorbereitung“ den Parameter „velocity“ auf z.B. 3600.0 %/s einstellen.</p> <p>Der Wert muss dem der anderen TO's übereinstimmen.</p> <p>(Auf diesen Wert bezieht sich die prozentuale Geschwindigkeitsvorgabe im Print Standard.)</p>																																																	
5	<p>Im Tab „Begrenzungen“ die Richtungsabhängige Dynamik „Direction-dependent dyn. response“ aktivieren.</p> <p>(Wird für die Vor- bzw. Rücklaufsperrung benötigt.)</p> <p>Die Begrenzungswerte selbst werden aus der Applikation heraus geschrieben.</p>																																																	
6	<p>Im Tab „Überwachungen“ die Geschwindigkeitsschwelle für die Stillstandsüberwachung auf einen möglichst kleinen Wert (z.B. 1.8 %/s) einstellen.</p>																																																	
7	<p>Bei einer Gleichlaufachse müssen die Verschaltungen des Gleichlaufobjekts im Tab „Verschaltungen“ vorgenommen werden.</p>	 <table border="1" data-bbox="901 1413 1364 1601"> <thead> <tr> <th>IO name</th> <th>Coupling type</th> <th>Device</th> </tr> </thead> <tbody> <tr><td>ADD_RA_Infeed</td><td>Slave</td><td>Slave</td></tr> <tr><td>ADD_PA_Infeed</td><td>Master</td><td>Master</td></tr> <tr><td>ADD_RA_PC_Off</td><td>Master</td><td>Slave</td></tr> <tr><td>FORMULA_RA_Unwinder</td><td>Slave</td><td>Slave</td></tr> <tr><td>FORMULA_RA_Unwinder</td><td>Master</td><td>Master</td></tr> <tr><td>RA_Infeed</td><td>Slave</td><td>Slave</td></tr> <tr><td>RA_PC_Off</td><td>Slave</td><td>Master</td></tr> <tr><td>RA_PC_Off</td><td>Slave</td><td>Slave</td></tr> <tr><td>RA_Unwinder</td><td>Slave</td><td>Master</td></tr> <tr><td>RA_SpdCtrl</td><td>Slave</td><td>Slave</td></tr> <tr><td>RA_Unwinder</td><td>Master</td><td>Master</td></tr> <tr><td>VA_LM_FormatS</td><td>Master</td><td>Master</td></tr> <tr><td>VA_Om_Web</td><td>Master</td><td>Master</td></tr> <tr><td>VA_LM_Format</td><td>Master</td><td>Master</td></tr> <tr><td>VA_LM_WebS</td><td>Slave</td><td>Slave</td></tr> </tbody> </table>	IO name	Coupling type	Device	ADD_RA_Infeed	Slave	Slave	ADD_PA_Infeed	Master	Master	ADD_RA_PC_Off	Master	Slave	FORMULA_RA_Unwinder	Slave	Slave	FORMULA_RA_Unwinder	Master	Master	RA_Infeed	Slave	Slave	RA_PC_Off	Slave	Master	RA_PC_Off	Slave	Slave	RA_Unwinder	Slave	Master	RA_SpdCtrl	Slave	Slave	RA_Unwinder	Master	Master	VA_LM_FormatS	Master	Master	VA_Om_Web	Master	Master	VA_LM_Format	Master	Master	VA_LM_WebS	Slave	Slave
IO name	Coupling type	Device																																																
ADD_RA_Infeed	Slave	Slave																																																
ADD_PA_Infeed	Master	Master																																																
ADD_RA_PC_Off	Master	Slave																																																
FORMULA_RA_Unwinder	Slave	Slave																																																
FORMULA_RA_Unwinder	Master	Master																																																
RA_Infeed	Slave	Slave																																																
RA_PC_Off	Slave	Master																																																
RA_PC_Off	Slave	Slave																																																
RA_Unwinder	Slave	Master																																																
RA_SpdCtrl	Slave	Slave																																																
RA_Unwinder	Master	Master																																																
VA_LM_FormatS	Master	Master																																																
VA_Om_Web	Master	Master																																																
VA_LM_Format	Master	Master																																																
VA_LM_WebS	Slave	Slave																																																

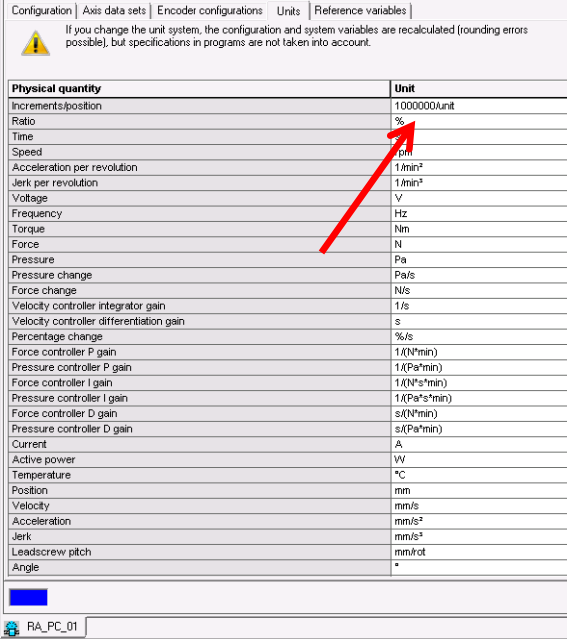
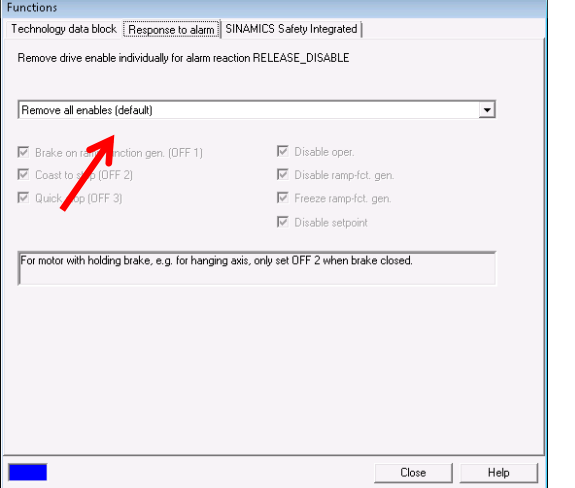
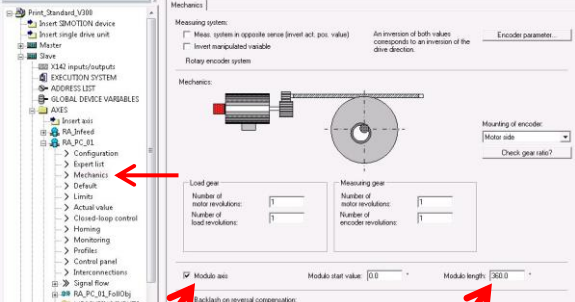
Step	Action	Picture
8	Variablen Definition und FB Aufrufe für neue Achse: <ul style="list-style-type: none"> • Existierende Achs-Unit kopieren • Achs-Namen mit Suchen & Ersetzen Funktion anpassen Neue Programme in das SIMOTION Ablaufsystem einhängen	
9	Speichern und Übersetzen des Scout Projekts. Ggf. Fehler (siehe Compiler-Nachrichten) beheben. Projekt mit Konsistenzprüfung überprüfen.	

3.4 Einfügen einer realen Achse ins Projekt

Tabelle 3-2 Einfügen einer realen Achse

Step	Action	Picture
1	Einen neuen Antrieb mit dem "Insert drive" Wizard in SINAMICS mit folgenden Werten einfügen: <ul style="list-style-type: none"> • Kein Funktionsmodul • Drehzahlregelung (m. Geber) • p864=1 (wird später mit Script gesetzt) • Motor Module auswählen • Motor / Geber / Bremse auswählen 	
2	Neue SIMOTION Achse mit dem "Insert axis" Wizard mit folgenden Werten einfügen: <ul style="list-style-type: none"> • Achs-Technologie: Drehzahl-, Positionier- oder Gleichlaufachse • Achstyp wählen (z.B. Rundachse, elektrisch) • Antriebszuordnung herstellen • Geberzuordnung herstellen • Mögliche Master am Gleichlaufobjekt (wenn vorhanden) vorbelegen 	
3	Technologiedatenblock aktivieren (Telegramm wird bei Symbolischer Zuordnung automatisch erweitert)	

3 Integration

Step	Action	Picture																																																																
4	<p>In der Achs-Konfiguration im Tab „Einheiten“ die Inkremente/Position auf Maximalwert (1.000.000) einstellen</p>	 <p>Configuration Axis data sets Encoder configurations Units Reference variables</p> <p>If you change the unit system, the configuration and system variables are recalculated (rounding errors possible), but specifications in programs are not taken into account.</p> <table border="1"> <thead> <tr> <th>Physical quantity</th> <th>Unit</th> </tr> </thead> <tbody> <tr><td>Increments/position</td><td>1000000Unit</td></tr> <tr><td>Ratio</td><td>%</td></tr> <tr><td>Time</td><td>s</td></tr> <tr><td>Speed</td><td>rpm</td></tr> <tr><td>Acceleration per revolution</td><td>1/min²</td></tr> <tr><td>Jerk per revolution</td><td>1/min³</td></tr> <tr><td>Voltage</td><td>V</td></tr> <tr><td>Frequency</td><td>Hz</td></tr> <tr><td>Torque</td><td>Nm</td></tr> <tr><td>Force</td><td>N</td></tr> <tr><td>Pressure</td><td>Pa</td></tr> <tr><td>Pressure change</td><td>Pa/s</td></tr> <tr><td>Force change</td><td>N/s</td></tr> <tr><td>Velocity controller integrator gain</td><td>1/s</td></tr> <tr><td>Velocity controller differentiation gain</td><td>s</td></tr> <tr><td>Percentage change</td><td>%/s</td></tr> <tr><td>Force controller P gain</td><td>1/(N*min)</td></tr> <tr><td>Pressure controller P gain</td><td>1/(Pa*min)</td></tr> <tr><td>Force controller I gain</td><td>1/(N*s*min)</td></tr> <tr><td>Pressure controller I gain</td><td>1/(Pa*s*min)</td></tr> <tr><td>Force controller D gain</td><td>s/(N*min)</td></tr> <tr><td>Pressure controller D gain</td><td>s/(Pa*min)</td></tr> <tr><td>Current</td><td>A</td></tr> <tr><td>Active power</td><td>W</td></tr> <tr><td>Temperature</td><td>°C</td></tr> <tr><td>Position</td><td>mm</td></tr> <tr><td>Velocity</td><td>mm/s</td></tr> <tr><td>Acceleration</td><td>mm/s²</td></tr> <tr><td>Jerk</td><td>mm/s³</td></tr> <tr><td>Leadscrew pitch</td><td>mm/rot</td></tr> <tr><td>Angle</td><td>°</td></tr> </tbody> </table>	Physical quantity	Unit	Increments/position	1000000Unit	Ratio	%	Time	s	Speed	rpm	Acceleration per revolution	1/min ²	Jerk per revolution	1/min ³	Voltage	V	Frequency	Hz	Torque	Nm	Force	N	Pressure	Pa	Pressure change	Pa/s	Force change	N/s	Velocity controller integrator gain	1/s	Velocity controller differentiation gain	s	Percentage change	%/s	Force controller P gain	1/(N*min)	Pressure controller P gain	1/(Pa*min)	Force controller I gain	1/(N*s*min)	Pressure controller I gain	1/(Pa*s*min)	Force controller D gain	s/(N*min)	Pressure controller D gain	s/(Pa*min)	Current	A	Active power	W	Temperature	°C	Position	mm	Velocity	mm/s	Acceleration	mm/s ²	Jerk	mm/s ³	Leadscrew pitch	mm/rot	Angle	°
Physical quantity	Unit																																																																	
Increments/position	1000000Unit																																																																	
Ratio	%																																																																	
Time	s																																																																	
Speed	rpm																																																																	
Acceleration per revolution	1/min ²																																																																	
Jerk per revolution	1/min ³																																																																	
Voltage	V																																																																	
Frequency	Hz																																																																	
Torque	Nm																																																																	
Force	N																																																																	
Pressure	Pa																																																																	
Pressure change	Pa/s																																																																	
Force change	N/s																																																																	
Velocity controller integrator gain	1/s																																																																	
Velocity controller differentiation gain	s																																																																	
Percentage change	%/s																																																																	
Force controller P gain	1/(N*min)																																																																	
Pressure controller P gain	1/(Pa*min)																																																																	
Force controller I gain	1/(N*s*min)																																																																	
Pressure controller I gain	1/(Pa*s*min)																																																																	
Force controller D gain	s/(N*min)																																																																	
Pressure controller D gain	s/(Pa*min)																																																																	
Current	A																																																																	
Active power	W																																																																	
Temperature	°C																																																																	
Position	mm																																																																	
Velocity	mm/s																																																																	
Acceleration	mm/s ²																																																																	
Jerk	mm/s ³																																																																	
Leadscrew pitch	mm/rot																																																																	
Angle	°																																																																	
5	<p>In den Funktions-Einstellungen (Tab „Konfiguration“), im Tab „Reaktion bei Alarm“ die Reaktion auf „Alle Freigaben wegnehmen (default)“ einstellen.</p> <p>Bei hängenden Achsen müssen die Freigaben „OFF2“ und „Disable operation“ erhalten bleiben (die beiden Hacken dürfen nicht gesetzt werden)!</p>	 <p>Functions</p> <p>Technology data block Response to alarm SINAMICS Safety Integrated</p> <p>Remove drive enable individually for alarm reaction RELEASE_DISABLE</p> <p>Remove all enables (default)</p> <p><input checked="" type="checkbox"/> Brake on ramp-fct. gen. (OFF 1) <input checked="" type="checkbox"/> Disable oper.</p> <p><input checked="" type="checkbox"/> Coast to stop (OFF 2) <input checked="" type="checkbox"/> Disable ramp-fct. gen.</p> <p><input checked="" type="checkbox"/> Quick stop (OFF 3) <input checked="" type="checkbox"/> Freeze ramp-fct. gen.</p> <p><input type="checkbox"/> Disable setpoint</p> <p>For motor with holding brake, e.g. for hanging axis, only set OFF 2 when brake closed.</p>																																																																
6	<p>Im Tab „Mechanik“ die Modulo-Achse aktivieren und die Modulo-Länge (z.B. 360°) eingeben.</p> <p>Die Länge muss mit der Modulo-Länge der anderen TO's übereinstimmen.</p>	 <p>Print Standard V300</p> <p>Insert SIMOTION device</p> <p>Insert single drive unit</p> <p>Drive</p> <p>SI442 Inputs/Outputs</p> <p>DESCRIPTION SYSTEM</p> <p>ADDRESS LIST</p> <p>GLOBAL DEVICE VARIABLES</p> <p>AXES</p> <p>Insert axis</p> <p>RA_1nFeed</p> <p>RA_PC_01</p> <p>Configuration</p> <p>Expert list</p> <p>Mechanics</p> <p>Default</p> <p>Limits</p> <p>Actual value</p> <p>Closed-loop control</p> <p>Homing</p> <p>Monitoring</p> <p>Profiles</p> <p>Control panel</p> <p>Interconnections</p> <p>Signal flow</p> <p>RA_PC_01_FncObj</p> <p>MHA2nDnV3nVnVTS</p> <p>Mechanics</p> <p>Measuring system</p> <p>Mass system in opposite sense (invert act. pos. value) An inversion of both values corresponds to an inversion of the drive direction</p> <p>Encoder parameters</p> <p>Insert non-quoted variable</p> <p>Ridley encoder system</p> <p>Mechanics</p> <p>Mounting of encoder (Motor side) Check gear ratio?</p> <p>Load gear</p> <p>Number of motor revolutions: [] Number of motor revolutions: []</p> <p>Number of load revolutions: [] Number of encoder revolutions: []</p> <p><input checked="" type="checkbox"/> Modulo axis Modulo start value: [0.0] Modulo length: [360.0]</p> <p>Backlash on reversal compensation:</p>																																																																

3 Integration

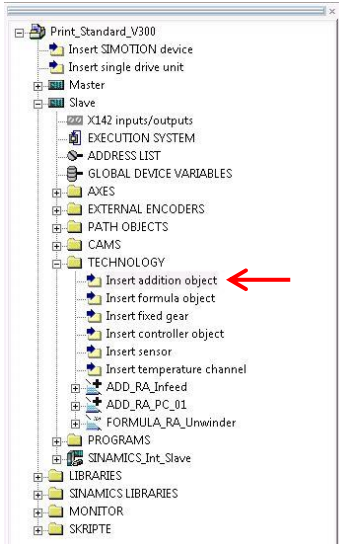
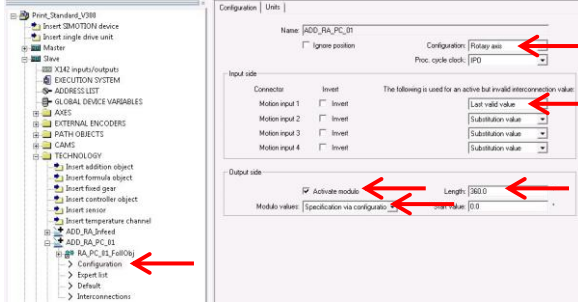
Step	Action	Picture
7	<p>Im Tab „Vorbelegung“ den Parameter „velocity“ auf z.B. 3600.0 %/s einstellen.</p> <p>Der Wert muss dem der anderen TO's übereinstimmen.</p> <p>(Auf diesen Wert bezieht sich die prozentuale Geschwindigkeitsvorgabe im Print Standard.)</p>	
8	<p>Im Tab „Begrenzungen“ die Richtungsabhängige Dynamik „Direction-dependent dyn. response“ aktivieren.</p> <p>(Wird für die Vor- bzw. Rücklaufsperrung benötigt.)</p> <p>Die Begrenzungswerte selbst werden aus der Applikation heraus geschrieben.</p>	
9	<p>Ist ein Gleichlaufobjekt vorhanden muss auch dort im Tab „Einheit“ der Wert „Inkremete/Position“ auf den Maximalwert (1.000.000/Unit) bzw. dem Wert entsprechend der anderen TO's eingestellt werden.</p>	
10	<p>Bei einer Gleichlaufachse müssen die Verschaltungen des Gleichlaufobjekts im Tab „Verschaltungen“ vorgenommen werden.</p>	
11	<p>Variablen Definition und FB Aufrufe für neue Achse:</p> <ul style="list-style-type: none"> • Existierende Achs-Unit kopieren • Achs-Namen mit Suchen & Ersetzen Funktion anpassen <p>Neue Programme in das SIMOTION Ablaufsystem einhängen</p>	
12	<p>Speichern und Übersetzen des Scout Projekts.</p> <p>Ggf. Fehler (siehe Compiler-Nachrichten) beheben.</p> <p>Projekt mit Konsistenzprüfung überprüfen.</p>	

VORSICHT Bei hängenden Achsen kann die Fehlerreaktion „Alle Freigaben wegenehmen“ zum Absacken oder Herunterfallen der Achse führen, wenn `TypeOfAxis.DriveControlConfig.disableModeSpecification = NO_SPECIFIC_MODE` eingestellt ist.

`TypeOfAxis.DriveControlConfig.disableModeSpecification = OFF1_AND_OFF3_MODE` einstellen.

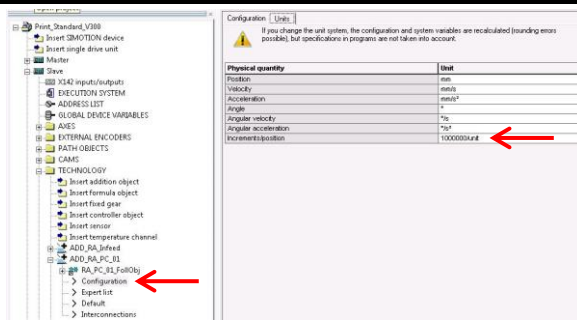
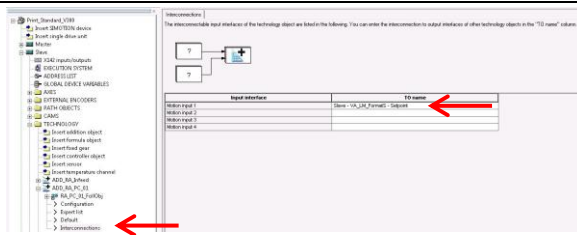
3.5 Einfügen eines Addierobjekts ins Projekt

Tabelle 3-3 Einfügen eines Addierobjekts

Step	Action	Picture
1	<p>Neues Addierobjekt mit dem "Insert addition object" Wizard einfügen:</p>	
2	<p>Im Tab „Konfiguration“ folgende Einstellungen vornehmen:</p> <ul style="list-style-type: none"> • Rundachse • IPO (gleicher Takt wie TO Achse) • Motion Input 1: Letzter gültiger Wert • Modulo Achse • Modulolänge 360° • Modulowerte: Spezifikation wie Konfiguration 	

Copyright © Siemens AG 2016 All rights reserved

3 Integration

Step	Action	Picture
3	Im Tab „Einheit“ den Wert „Inkremete/Position“ auf den Maximalwert (1.000.000/Unit) bzw. dem Wert entsprechend den anderen TO's einstellen.	
4	Im Tab „Verschaltungen“ den Motion Input 1 mit dem Master verbinden.	

3.6 Normierung Drehzahl und Drehmoment in SIMOTION & SINAMICS

Bei den Normierungen von Drehzahl und Drehmoments in SIMOTION sind einige wichtige Punkte zu beachten. SIMOTION und SINAMICS kommunizieren über den internen PROFIBUS miteinander. Drehzahl und Drehmoment werden in Prozent übergeben d.h. in beiden Systemen ist ein Normierungsfaktor hinterlegt, der festlegt wie der prozentuale Wert vom jeweiligen System (SIMOTION oder SINAMICS) bewertet wird. Die Tabelle 3-2 zeigt die einzustellenden Konfigurationsvariablen und Parameter.

Tabelle 3-2 Normierungsfaktoren SIMOTION/SINAMICS

Parameter / Achskonfigurationsvariable	Wert (E.g. Print Standard)	Beschreibung
p2000 (SINAMICS)	5000.0 1/min	Bezugsdrehzahl
TypeofAxis.SetPointDriverInfo.DriveData.speedReference (SIMOTION)	[1]nominal value [0] max value	Einstellung der Auswahl für die Normierung: Bezugsdrehzahl oder Maximaldrehzahl
TypeofAxis.SetPointDriverInfo.DriveData.maxSpeed (SIMOTION)	5000.0 1/min	Maximale Geschwindigkeit der Achse, dieser Wert ist die absolute Geschwindigkeitsgrenze die SIMOTION zulässt
TypeofAxis.SetPointDriverInfo.DriveData.nominalSpeed (SIMOTION)	5000.0 1/min	Bezugsdrehzahl in SIMOTION (muss gleich p2000 im Antrieb sein)
p2003 (SINAMICS)	0.6 Nm	Bezugsdrehmoment
TypeofAxis.SetPointDriverInfo.DriveData.maxTorque (SIMOTION)	0.6 Nm	Maximales Drehmoment der Achse, dieser Wert ist die absolute Drehmomentengrenze die SIMOTION über den internen PROFIBUS sendet und gleichzeitig das Bezugsdrehmoment. (muss gleich p2003 im Antrieb sein)

4 Funktionsbeschreibung

Der Abschnitt „Funktionsbeschreibung“ kann grob in folgende Abschnitte gegliedert werden:

Software Konzept, Programmstruktur und Applikationsvariablen

- Beschreibung des Programm Konzeptes
- Programm Struktur
- Programm und FB Beschreibung
- Variablenstrukturen

Funktionsbeschreibung

- Beschreibung der Achsbetriebsarten
- Schnittstellen (Motion Programm, SPS und Antrieb)

Drehmoment Funktionen und Technologische Erweiterung

- Drehmoment Funktionen
- Tech Task / Tech FB Konzept
- Nutzung Addier-, Formelobjekt und MotionIn
- Integration von Wickler und Zugregelung

Scripting

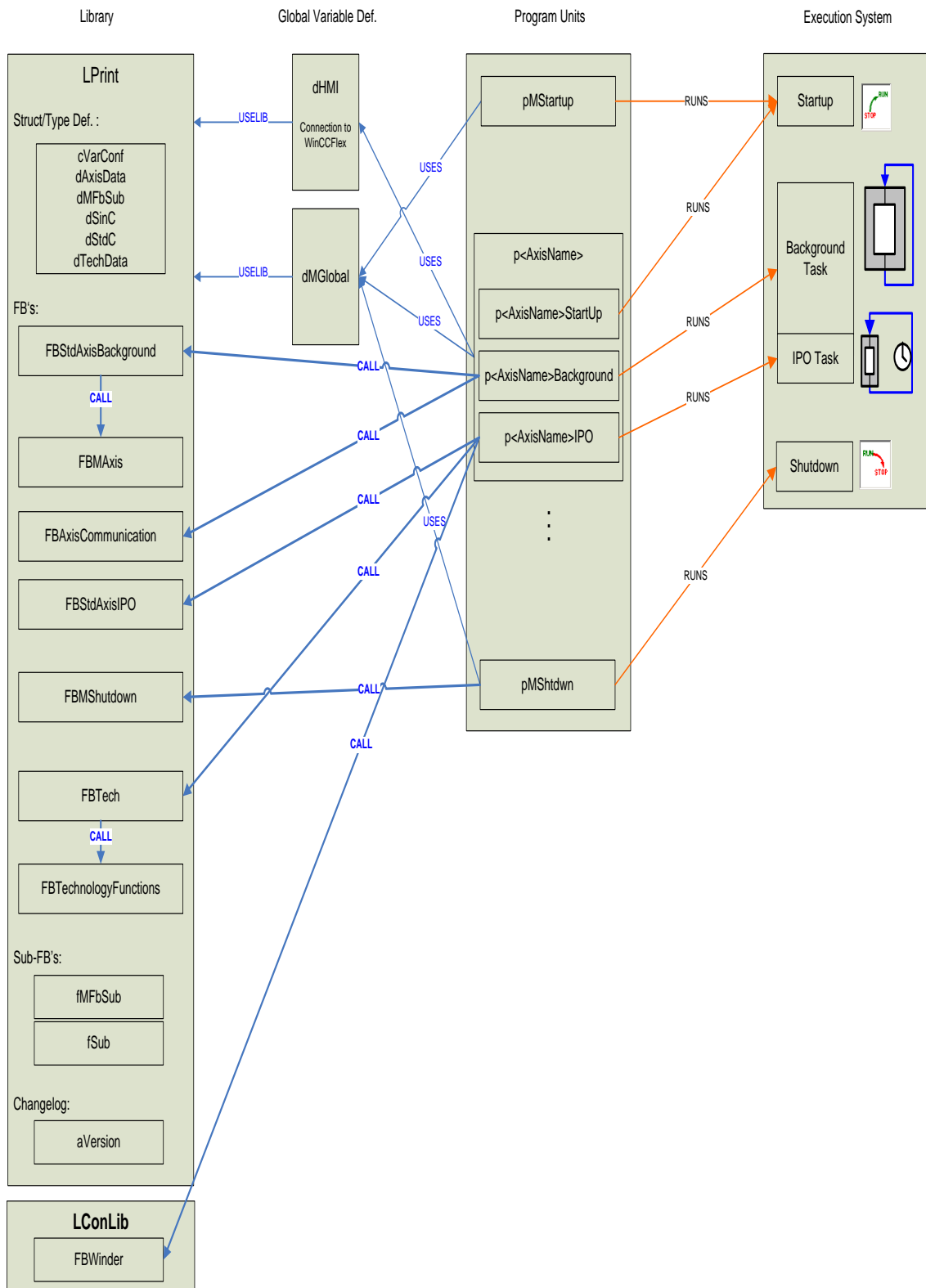
- Antriebs und TO Parametrierung

4.1 Programm Struktur

Die folgende Abbildung zeigt die einzelnen Teile des Print Standards und deren zusammenwirken. Auf der linken Seite die Standard Bibliothek „LPrint“. Sie enthält die Definitionen der Basis Strukturen und die Funktionsbausteine. Im mittleren Teil die globalen Variablen und die einzelne Programme mit den Aufrufen der Funktionsbausteine aus der Bibliothek. Im rechten Teil ist die Zuordnung der Programme im SIMOTION Task System dargestellt.

4 Funktionsbeschreibung

Abbildung 4-1 Programm Struktur



4.2 Programme und Funktionsbausteine

Die wichtigste Strategie des Print Standard Programmierkonzept basiert darauf, dass für jede Achse jeweils drei Haupt-Funktionsbausteine in zwei Haupt-Programmen aufgerufen werden:

- FBLPrint_StdAxisBackgrnd Background-Task-Programm
- FBLPrint_StdAxisIPO IPO-Task-Programm
- FBLPrint_MotionAxis Background-Task-Programm

Diese Funktionsbausteine befinden sich in der Bibliothek „LPrint“ in den Quellen „fStdAxis“ und „fMFb“.

4.2.1 Überblick über Programm-Units und Programme

dHMI

In dieser Unit wird für jede Achse eine zusätzliche, unabhängige STDclO Variablenstruktur angelegt, die mit der IT-Diag Seite verbunden ist.

Der Funktionsbaustein „FBLPrint_AxisCommunication“ jeder Achse kann dann eine Verbindung zwischen der HMI-STDclO-Struktur und der eigentlichen STDclO-Struktur herstellen.

dMGlobal

Definition von Achs-übergreifenden Variablen und Variablen-Strukturen.

Achs-spezifische Strukturen werden innerhalb der Achs-Unit global definiert, FB-Instanzen werden innerhalb der Achs-Unit lokal definiert.

pMStartup

In dieser Unit befindet sich das Startup Programm „pAxesStartup“.

Es wird einmalig in der Startup-Task nach Hochlauf der SIMOTION CPU (Stop-Run Übergang) aufgerufen und für Vorbelegungen (z.B. Dynamiken und Konfigurationen der Achsen) verwendet.

p<AxisName>

Diese Unit enthält drei Programme:

„p<AxisName>Startup“

Achs-Spezifische Vorbelegungen

“p<AxisName>Background“

“ FBLPrint_StdAxisBackgrnd“ Funktionsbaustein aufruf

“ FBLPrint_AxisCommunication“ Funktionsbaustein aufruf

“p<AxisName>IPO“

“ FBLPrint_StdAxisIPO“ Funktionsbaustein aufruf

“ FBLPrint_TechAxis“ Funktionsbaustein aufruf

pMShtdwn

Diese Unit enthält das Programm „pMShtdwn“. Dort wird der Bibliotheks-Funktionsbaustein “ FBLPrint_MotionShutDown” aufgerufen.

pFault

Diese Unit enthält zwei Programme:

„pPeripheralFault“

„pTechnologicalFault“

Beide Programme sind leere “Programm-Vorlagen” die mit Programmen zum System Fehlerhandling gefüllt werden können.

ACHTUNG	Beide Programme (ob leer oder mit Programm-Code gefüllt) müssen den jeweiligen Tasks im Ablaufsystem zugeordnet werden.
----------------	--

	Im Falle eines System Fehlers werden diese Tasks aufgerufen. Ist kein Programm zugeordnet, wechselt die SIMOTION CPU in den Betriebszustand STOP.
--	---

4.2.2 Überblick Funktionsbausteine

FBLPrint_StdAxisBackgrnd

Der Background Baustein beinhaltet u.a. folgende Funktionalität:

- Einmalige Initialisierung der verschiedenen benutzten Achstypen inkl. einer Konsistenzprüfung der programmierten Achs-Modi und Achstypen
- Verarbeitung aller STDcIO Eingangs Werte und internen Sollwerte
- Verarbeitung aller STDcIO Ausgangs Werte
- Erkennung eines angeforderten Betriebsarten-Wechsels
- Aufruf des Motion Funktionsbausteins „FBLPrint_MotionAxis“

FBLPrint_LPrint_StdAxisIPO

Dieser Funktionsbaustein wird für kritische Berechnungen verwendet, die im IPO-Takt berechnet werden müssen und unbedingt konsistent sein müssen.

FBLPrint_AxisCommunication

Dieses Funktionsbaustein realisiert den Datentransfer von allen Schnittstellen außerhalb SIMOTION oder innerhalb SIMOTION zur Print Standard STDc Motion Schnittstelle.

Über Eingangsvariablen des Funktionsbaustein kann ausgewählt werden, ob das SPS Programm von außerhalb SIMOTION (externe PLC, z.B. S7), innerhalb SIMOTION (interne PLC) oder eine WinCC flexible Applikation bzw. IT-Diag-Seiten die Print Standard Achsen ansteuert.

FBLPrint_MotionShutDown

Der Funktionsbaustein löst zur Sicherheit einen Stop Emergency Befehl an alle Achsen aus, falls die SIMOTION CPU z.B. aufgrund eines „nicht abgefangenen“ Systemfehlers von Run nach STOP wechselt, während Achsen in Betrieb sind.

FBLPrint_MotionAxis

Der MCC FB_M_Axis beinhaltet die Programmierung aller definierten Achsbetriebsarten in einem „case statement“. Ein Betriebsartwechsel ist jederzeit möglich.

Der FB_M_Axis kann von allen Achstypen aufgerufen werden, unabhängig davon ob alle Betriebsarten von der Achse unterstützt werden. Alle programmierten Bewegungsbefehle an Achsen werden ausschließlich von diesem Funktionsbaustein abgesetzt.

Jeder Zweig entspricht einer Betriebsart. Anwender Betriebsarten können schnell und unabhängig integriert werden indem diese vorhandene Struktur erweitert wird.

FBLPrint_TechAxis

Dieser Funktionsbaustein realisiert die Schnittstelle von Programmcode zu den SIMOTION Technologie-Objekten (Achsen, Addierobjekt, Formelobjekt).

Eine detaillierte Beschreibung ist im Kapitel [4.13 Technologie-Erweiterungen: Wickler, Zugregelung und Register Regelung](#) zu finden.

4.2.3 Motion Funktionsbaustein IPO-synchron aufrufen

Für spezielle Anwendungen kann es notwendig sein, den Motion Funktionsbaustein IPO-Synchron anstelle in der Background-Task aufzurufen.

Z.B. synchrone Positionierung von mehreren Achsen.

Dazu muss der Baustein „FBLPrint_StdAxisBackgrnd“ (beinhaltet den Bausteinaufruf von „FBLPrint_MotionAxis“), im Print Standard IPO Programm aufgerufen werden.

Während der Initialisierungsphase (ein Zyklus beim erstmaligen Bausteinaufruf) wird im Baustein lesend auf Konfigurationsvariablen der Technologieobjekte zugegriffen. Dies führt zu einem kurzzeitigen Laufzeitanstieg der Task.

Um einen IPO-Task Überlauf zu vermeiden wird empfohlen den Funktionsbaustein während der Initialisierung in der Background-Task, und erst anschließend in der IPO-Task aufzurufen.

Dazu muss die Funktionsbaustein-Instanz „FBLPrint_StdAxisBackgrnd“ in der Programm-Unit Implementation-Global angelegt und im Programcode zwei Mal (Background- und IPO-Programm) mit entsprechender IF-Bedingung aufgerufen werden.

Die Variable `axisTOConfigData.sInternal.boValid` kann verwendet werden um den Baustein von der Bearbeitung in der Background-Task auf die IPO-Task umzuschalten. Im Background Programm wird der Baustein nur aufgerufen, wenn `boValid = FALSE`, im IPO Programm wenn `boValid = TRUE`.

Hinweis

Wird die Initialisierung später zur Laufzeit erneut angestoßen (`axisConfigData.boRestartInitialization`) sollte der Funktionsbaustein für diese Zeit erneut in der Background-Task bearbeitet werden.

4.3 STDC - Die Print Standard Motion Schnittstelle

Die wichtigste Schnittstelle jeder Achse ist das STDcIO Interface, welche in der Bibliotheks-Unit „dStdC“ definiert ist.

Diese Schnittstelle ist konzipiert als die einzig notwendige Verbindung zwischen SPS Programm und der kompletten Motion Funktionalität des Print Standard.

Im Folgenden wird die “STDC” Schnittstelle beschrieben:

Tabelle 4-1 Control Structure STDcIO.IN

Name	Type	Beschreibung
boOff1EStop	BOOL	FALSE: Wechsel in Mode E_STOP TRUE: Wechsel in angewählten Mode (u16ModeNumber). Die Achse wird freigegeben!
boEnableSpeedSetpoint	BOOL	TRUE: Drehzahlsollwert von der Schnittstelle wird freigegeben: FALSE: interner Drehzahlsollwert wird auf Null gesetzt, egal ob MOP oder Sollwert Modus
boSynchronizeAngular	BOOL	Steigende Flanke: absolute Winkel-Synchronisation wird gestartet, (nur wenn Achse in mode 60 “GEAR_MODE” bereits “boSpeedSynchronous ist”
boRelativePositioning	BOOL	Steigende Flanke: Nächste Positionierung wird relativ ausgeführt FALSE: Nächste Positionierung wird absolut ausgeführt
boReferencingOnTheFly	BOOL	Steigende Flanke: Starte fliegendes referenzieren (Mode 30, 60) oder Speichern der Inkrementalgeberposition. siehe Beschreibung Betriebsart „REFERENCING“
boActivateAbsRegister	BOOL	Steigende Flanke: r64RegisterSetpoint wird in mode “GEAR_MODE” absolut angefahren FALSE: keine Register Verstellung
boModeAbsRegister	BOOL	TRUE: Register wird auf “kürzestem Weg” verstellt in mode “GEAR_MODE” FALSE: Register wird in Vorwärtsrichtung verstellt
boActivateRelRegister	BOOL	Steigende Flanke: r64RegisterSetpoint wird in mode “GEAR_MODE” relativ von aktuellem Register angefahren FALSE: keine Register Verstellung
boNegativeRelRegister	BOOL	TRUE: Register wird bei relativer Registerverstellung in negative Richtung verstellt FALSE: Register wird bei relativer Registerverstellung in positive Richtung verstellt
boInchRegisterAdvance	BOOL	TRUE: Register Tippen in positive Richtung, in mode “GEAR_MODE”
boInchRegisterRetard	BOOL	TRUE: Register Tippen in negative Richtung, in mode “GEAR_MODE”
boAcknowledgeFault	BOOL	Steigende Flanke: Print Standard Fehler werden quittiert; Die Achse muss sich dazu in der Betriebsart „E_STOP“ befinden
boStartMotion	BOOL	FALSE -> TRUE: Bewegung starten TRUE -> FALSE: Bewegung stoppen
boEnableTorqueLimiting	BOOL	TRUE: Drehmomentgrenzen sind aktiv FALSE: keine Verwendung der Drehmomentgrenzen

4 Funktionsbeschreibung

Name	Type	Beschreibung
boEnableReverseMotionInhibit	BOOL	TRUE: Rücklaufsperrung ist aktiviert FALSE: keine Rücklaufsperrung
boEnablePositiveMotionInhibit	BOOL	TRUE: Vorwärtssperre ist aktiviert FALSE: keine Vorwärtssperre
boMaskMotorBlockedFault	BOOL	TRUE: SINAMICS Fehler "motor blocked" wird unterdrückt FALSE: SINAMICS Fehler "motor blocked" wird nicht unterdrückt
boEnableAdditiveTorque	BOOL	TRUE: Drehmomentzusatzsollwert freigeben FALSE: Drehmomentzusatzsollwert sperren
boOpenBrake	BOOL	TRUE: Open brake (set axis stw bit 12)
boSetpointMode	BOOL	TRUE: Drehzahlsollwert erzeugt von MOP; FALSE: Drehzahlsollwert direkt von „STDCIO.IN.r64SpeedSetpoint“
boMopFaster	BOOL	MOP schneller
boMopSlower	BOOL	MOP langsamer
boMopHold	BOOL	MOP halten
r64SpeedSetpoint	LREAL	Drehzahlsollwert benutzt in Modi: SPEED_POS_CTRL SPEED_SPD_CTRL POSITIONING REFERENCING Die Einheit ist abhängig von dem Konfigurationsparameter AxisConfigData.eVelocityFormat
r64PositioningSetpoint	LREAL	Ziel Positionssollwert in mode „POSITIONING“; Die Einheit ist abhängig von der Konfiguration am Achs TO. z.B. Grad für "standard rotary axis".
r64RegisterSetpoint	LREAL	Register Sollwert Die Einheit ist abhängig von der Konfiguration am Achs TO. z.B. Grad für "standard rotary axis".
r64ActualCylinderFormatLength	LREAL	Aktuelles Format (bzw. Formatlänge) des angetriebenen Lastzylinders der Achse. Dieser Wert wird typischerweise bei "formatvariablen" Maschinen benötigt, wie z.B. Flexo und Tiefdruck. Das Verhältnis von aktuellem Format zu Referenzformat wird benutzt um Sollwerte und dynamische Daten (Hoch/Rücklaufzeiten etc.) anzupassen. Die Einheit dieses Wertes muss ebenso wie der reference Wert in der Variablen AxisConfigData.sCylinderFormat.r64ReferenceLength in [mm] angegeben werden. Beispiel: aktuelles Format 1100 mm 10% mehr als Referenz Format (z.B. 1000 mm): - AxisConfigData.sCylinderFormat.r64ReferenceLength = 1000 [mm] - STDCIO.IN.r64ActualCylinderFormatLength = 1100 [mm]
r64TorqueLimitPositive	LREAL	Positive Drehmomentgrenze in [%] von AxisTOConfigData.sInternal.r64MaxTorque bzw. AxisTOConfigData.sInternal.r64NominalTorque

4 Funktionsbeschreibung

Name	Type	Beschreibung
r64TorqueLimitNegative	LREAL	Negative Drehmomentgrenze in [%] von AxisTOConfigData.sInternal.r64MaxTorque bzw. AxisTOConfigData.sInternal.r64NominalTorque
r64AdditiveTorque	LREAL	Zusatzmoment in [%] von AxisTOConfigData.sInternal.r64MaxTorque bzw. AxisTOConfigData.sInternal.r64NominalTorque
r64DrawSetpoint	LREAL	Draw Sollwert (in % in Bezug auf i32GearNumSetpoint)
i32GearNumSetpoint	DINT	Getriebe Faktor Zähler (im Sollwertkanal) Vorbelegung: 100000
i32GearDenSetpoint	DINT	Getriebe Faktor Nenner (im Sollwertkanal) Vorbelegung: 100000
u16ModeNumber	UINT	Vorgabe der Betriebsart
u8MasterNumber	USINT	Die möglichen Master die eine Achse folgen kann werden in einem Array/Liste in "pMStartp" definiert (possibleMasters). Zwischen diesen Mastern kann mit diesem Wert "u8MasterNumber" umgeschaltet werden. Bei u8MasterNumber := 0, wird der defaultMaster (axisTOConfigData.sUser.toDefaultMaster) als Master-Objekt verschaltet.
u8CamNumber	USINT	Die möglichen Kurvenscheiben, die eine Achse benutzen kann, werden in einem Array/Liste in "pMStartp" definiert (possibleCams). Zwischen diesen Kurvenscheiben kann mit diesem Wert "u8CamNumber" umgeschaltet werden. Bei u8CamNumber := 0, wird die defaultCam (axisTOConfigData.sUser.toDefaultCam) als Master-Objekt verschaltet.

Tabelle 4-2 Control Structure STDclO.OUT

Name	Type	Beschreibung
boReadyToPowerUp	BOOL	TRUE: Antrieb bereit zum Einschalten; "ready bit" vom Antrieb
boSwitchOnInhibit	BOOL	TRUE: Antrieb ist in Einschaltsperr
boExternalSwitchOnInhibit	BOOL	TRUE = Print Standard Applikation ist in Einschaltsperr
boOperation	BOOL	TRUE: Antrieb ist eingeschaltet FALSE: Antrieb ist ausgeschaltet
boZeroSpeed	BOOL	TRUE: Antrieb ist innerhalb der Stillstandserfassung (SINAMICS, Function Plan 8010)
boSpeedReached	BOOL	TRUE: zeigt an, dass die Achse nicht mehr beschleunigt/verzögert -> der Drehzahl Sollwert ist erreicht
boSpeedSynchronous	BOOL	TRUE: zeigt an dass die Achse die Geschwindigkeit des Master erreicht hat.
boAngularSynchronous	BOOL	TRUE: zeigt an, dass die Achse den überlagerten Positionierbefehl zum Herstellen des absoluten Registerbezugs zum Master abgeschlossen hat. Damit hat der Slave den gewünschten Offset zum Master.
boRegistrationActive	BOOL	TRUE: Register Bewegung läuft, (relativ oder absolut)
boRegistrationDone	BOOL	TRUE: Register Bewegung ist abgeschlossen
boPositioningActive	BOOL	TRUE: Positionier Bewegung läuft

4 Funktionsbeschreibung

Name	Type	Beschreibung
boPositioningDone	BOOL	TRUE: Positionier Bewegung ist beendet
boReferencingActive	BOOL	TRUE: Referenzier Vorgang läuft
boReferencingDone	BOOL	TRUE: Referenzieren ist abgeschlossen / Achse ist referenziert Information kommt von Retain Variable axisDataRetain.boReferencingDone Vorsicht: Bei Motor bzw. Geber-Tausch muss axisDataRetain.boReferencingDone von Hand zurückgesetzt werden!
boVMPosIsSet	BOOL	TRUE: Position der virtuelle Achse wurde gesetzt
boEStopActive	BOOL	TRUE: E-Stop wird ausgeführt
boEStopDone	BOOL	TRUE: E-Stop beendet, Achse ist deaktiviert
boBrakeOpen	BOOL	TRUE: Haltebremse ist geöffnet FALSE: Haltebremse ist geschlossen
boSTOActive	BOOL	TRUE: STO (Safe torque off / früher: safe standstill) Sicherheits Funktion ist im Antrieb aktiviert. d.h. Antrieb kann nicht eingeschalten werden Hinweis: Die (automatische) Verbindung/Auswertung zum Antrieb funktioniert nur bei gesetzter Präprozessor-Anweisung „#define LPRINT_USE_SAFETY_CHANNEL“ in der Bibliotheks-Unit „cVarConf“. Die Anweisung kann nur mit SIMOTION Firmware >=V4.4 verwendet werden!
boWarning	BOOL	TRUE: Antrieb ist im Warnungs Zustand
boFault	BOOL	TRUE: Antrieb ist im Fehler Zustand
boApplicationWarning	BOOL	TRUE: Print Standard zeigt eine Applikations Warnung an
boApplicationError	BOOL	TRUE: Print Standard zeigt einen Applikationsfehler an
boTorqueLimitingActive	BOOL	TRUE: Drehmoment Grenzen sind aktiv
boPositiveMotionInhibitActive	BOOL	TRUE: Vorlaufsperrung ist aktiv
boNegativeMotionInhibitActive	BOOL	TRUE: Rücklaufsperrung ist aktiv
boPositiveLimitReached	BOOL	TRUE: Positive Drehmomentgrenze ist erreicht
boNegativeLimitReached	BOOL	TRUE: Negative Drehmomentgrenze ist erreicht
r64ActualSpeed	LREAL	Aktuelle Geschwindigkeit der Achse. Die Einheit ist abhängig von dem Konfigurationsparameter AxisConfigData.eVelocityFormat: DIRECT: r64ActualSpeed := Axis.motionStateData.commandVelocity NORMALIZED_WITHOUT_FORMAT_ADAPTION: r64ActualSpeed := Axis.motionStateData.commandVelocity * 100 / axisTOConfigData.sInternal.r64DefaultVelocity NORMALIZED_WITH_FORMAT_ADAPTION: r64ActualSpeed := Axis.motionStateData.commandVelocity / axisData.cylinderFormatLengthRatio * 100 / axisTOConfigData.sInternal.r64DefaultVelocity

4 Funktionsbeschreibung

Name	Type	Beschreibung
r64ActualPosition	LREAL	Aktuelle Position der Achse Einheit hängt ab von Konfiguration am Achs TO. z.B. Grad für "standard rotary axis".
r64ActualRegister	LREAL	Aktuelles Register/Phasenlage der Achse Einheit hängt ab von Konfiguration am Achs TO. z.B. Grad für "standard rotary axis".
r64FollowingError	LREAL	Aktuelle Lageabweichung = following error Einheit hängt ab von Konfiguration am Achs TO. z.B. Grad für "standard rotary axis".
r64ActualTorque	LREAL	Aktuelles Motor Drehmoment in [%] von AxisTOConfigData.sInternal.r64MaxTorque bzw. AxisTOConfigData.sInternal.r64NominalTorque
r64ActivePositiveLimit	LREAL	Aktive positive Drehmoment Grenze in [%] von AxisTOConfigData.sInternal.r64MaxTorque bzw. AxisTOConfigData.sInternal.r64NominalTorque
r64ActiveNegativeLimit	LREAL	Aktive negative Drehmoment Grenze in [%] von AxisTOConfigData.sInternal.r64MaxTorque bzw. AxisTOConfigData.sInternal.r64NominalTorque
i32ActualGearNum	DINT	Aktiver Getriebe Zähler (Sollwert Getriebe)
i32ActualGearDen	DINT	Aktiver Getriebe Nenner (Sollwert Getriebe)
u16ActualModeNumber	UINT	Rückmeldung des aktiven Betriebsmodus; In Betriebsarten, in denen Bewegungen gestartet werden, wird der Mode erst zurückgegeben wenn die Achse bereit ist.
u16ApplicationWarningNumber	UINT	Anzeige Applikationswarnungs-Nummer
u16ApplicationErrorNumber	UINT	Anzeige Applikationsfehler-Nummer
u8ActualMasterNumber	USINT	aktiver Master, Nummer ist festgelegt in possibleMaster Array/Liste in "pMStartp".
u8ActualCamNumber	USINT	aktive Kurvenscheibe, Nummer ist festgelegt in possibleCam Array/Liste in "pMStartp".

4.4 Print Standard Variablen Strukturen

Im Folgenden werden die Print Standard Variablen Strukturen die für jede Achse angelegt werden erklärt. Diese sind in der Bibliotheks-Unit "dAxisData" definiert.

sAxisConfigDataType

Die Struktur "AxisConfigData" wird hauptsächlich benutzt um die "dynamischen Daten" der Achse und allgemeine Konfigurationen vorzubelegen.

Dies beinhaltet z.B. die Reaktion auf Print Standard Applikationsfehler, die Auswahl des Referenziermodus, Motorpotentiometer- sowie Zylinderformat-Einstellungen.

Die dynamischen Werte müssen berechnet/vorbelegt werden in den Einheiten der angelegten SIMOTION Achse. z.B. Für "rotary axis" die Einheiten für Grad/Geschwindigkeit/Beschleunigung/Ruck sind °, °/s, °/s² or °/s³.

Tabelle 4-3 sAxisConfigDataType:

Name	Datentyp	Beschreibung
au16AxisAllowedModeArray	'ARRAY [1..12] OF UINT'	Gültige Druck Standard Betriebsarten für die jeweiligen Achs-Typen. Der Parameter kann in einem StartUp-Programm z.B. mit folgenden Variablen vorbelegt werden: DRIVE_AXIS_MODE_ARRAY, POSITIONING_AXIS_MODE_ARRAY, FOLLOWING_AXIS_MODE_ARRAY, EXTENC_MODE_ARRAY, WINDING_DRIVE_AXIS_MODE_ARRAY, WINDING_POS_AXIS_MODE_ARRAY, GEAR_MODE_ARRAY, CAM_MODE_ARRAY
eErrorReaction	'eErrorReaction Type'	Konfiguration der gewünschten Reaktion der Achse auf einen Applikationsfehler: ERROR_COAST_STOP: Die Achse "trudelt aus" ERROR_E_STOP: Die Achse stoppt mit E-Stop Rampe Virtuelle Achsen müssen Error_E_Stop nutzen!
sMOP	'sMOPTyp'	Unterstruktur Motorpotentiometer
r64Min	LREAL	Maximaler Ausgangswert des MOP. Einheit abhängig von dem Konfigurationsparameter „eVelocityFormat“
r64Max	LREAL	Minimaler Ausgangswert des MOP. Einheit abhängig von dem Konfigurationsparameter „eVelocityFormat“
sCylinderFormat	'sAxisCylinderFormatConfigType'	Unterstruktur Zylinderformat-Konfiguration
r64MinLength	LREAL	Für „format-variable“ Maschinen: Der minimal zulässige Zylinder (Umfangs)- Wert in Bezug auf den Referenzzylinder. [mm]

4 Funktionsbeschreibung

Name	Datentyp	Beschreibung
r64MaxLength	LREAL	Für „format-variable“ Maschinen: Der maximal zulässige Zylinder (Umfangs)- Wert in Bezug auf den Referenzzylinder.[mm]
r64ReferenceLength	LREAL	Für “format-variable” Maschinen: Zylinder-Referenzwert zur notwendigen Skalierung der dynamischen Daten / Drehzahlen. [mm]
sHoming	'sAxisHomingConfigurationType'	Unterstruktur Referenzier-Konfiguration
eReferencingMode	'eRefModeType'	Konfiguration des Referenzier Modus: NO_MODE_SELECTED: default-Wert ACTIVE: Aktives Referenzieren PASSIVE: passives Referenzieren SET_AXIS_POSITION: Setzen der Position einer virtuellen Achse REF_ON_THE_FLY: Fliegendes referenzieren in Betriebsart 30 oder 60
r64HomePosition	LREAL	Angabe des gewünschten Positionssatzwertes für die Referenzier Modi
u16MeasInMaxNumOfRevolutions	UINT	Anzahl der Zylinderumdrehungen bevor eine Applikationswarnung ausgegeben wird während der Messtaster aktiv ist. (default = 1)
sLocalDyn	'sAccelDynamicsType'	Dynamiken im lokalen Betrieb (Mode 30/31)
sPosDyn	'sAccelDynamicsType'	Dynamiken im Positionier-Modus (Mode 20)
sClutchDyn	'sAccelDynamicsType'	Dynamiken zum Ein- und Auskuppeln in den Modes 50, 60 und 61.
sDrawDyn	'sAccelDynamicsType'	Dynamiken bei einem neuen Draw Sollwert bzw. Getriebeänderung
sAbsRegisterDyn	'sVeloDynamicsType'	Dynamiken für absolute Register Bewegungen im Mode 60
sRelRegisterDyn	'sVeloDynamicsType'	Dynamiken für relative Register Bewegungen im Mode 60
sInchRegisterDyn	'sVeloDynamicsType'	Dynamiken für tippen Register Bewegung im Mode 60
sSyncDyn	'sVeloDynamicsType'	Dynamiken zum Herstellen von Winkelgleichlauf im Mode 60 („boSynchronizeAngular“)
sSyncConfig	'sAxisSyncConfigurationType'	Unterstruktur Synchronisations-Konfiguration
eSyncMode	'eSyncModeType'	Einstellung der Richtung zum absoluten Aufsynchronisieren auf die Masterachse: Forward (nur vorwärts) Forward_with_window (vorwärts, rückwärts innerhalb des Fensters) Backward (nur rückwärts) Backward_with_window (rückwärts, vorwärts innerhalb des Fensters) Shortest_way (kürzester Weg)

4 Funktionsbeschreibung

Name	Datentyp	Beschreibung
r64SyncWindow	LREAL	Wenn der zu korrigierende Synchronisierfehler innerhalb dieses Fensters liegt, dann wird auf "kürzestem Weg" korrigiert, außerhalb des Fensters findet die Korrektur in entsprechender Richtung statt.
r64SynchAxisCycle	LREAL	Die Synchronisier-Achszykluslänge kann unabhängig von Achs TO modulo.length gesetzt werden. Die Achse wird dann nur innerhalb dieses Zyklus synchronisieren. (z.B. nur innerhalb $360/24 = 15$ Grad bei einem „Falz Auslage Fächer“ mit 24 Fächern.) Siehe dazu auch Einstellung von r64SynchAxisCycle bei Getriebefaktor ungleich 1
eSyncStartCondition	eLPrintSyncStartConditionType	Auswahl ob STDcIO Bit boSynchronizeAngular Flanken- oder Zustandsgetriggert ausgewertet wird. EDGE_TRIGGERED: (Neue) steigende Flanke notwendig um Synchronisation zu starten. STATE_TRIGGERED: Synchronisation wird erneute angestoßen wenn boSynchronizeAngular = TRUE und STDcIO.OUT Bit boAngularSynchronous = FALSE
sEStopDyn	'sStopDynamics Type'	Dynamiken für E_STOP (Mode 1)
sAxisCtrlSettings	'sAxisCtrlSettingsType'	Unterstruktur Achs-Steuerungs-Konfiguration
eldleStateMode	'eldleStateMode Type'	Abhängig von dieser Einstellung wird die Achse am Ende eines E-Stops in den Zustand Einschaltbereit oder Einschaltsperrung gesetzt. READY_TO_POWER_UP: Einschaltbereit – Das Bit „STDcIO.out.boReadyToPowerUp“ wird nach einem E-Stop wieder gesetzt (AUS2 und AUS3 werden gesetzt). SWITCH_ON_INHIBIT: Einschaltsperrung - Das Bit „STDcIO.out.boReadyToPowerUp“ wird nach einem E-Stop nicht gesetzt (AUS2 und AUS3 werden nicht gesetzt). Bei virtuellen Achsen ist der Parameter nicht relevant.
tBreakReleaseTime	Time	tBreakReleaseTime <> T#0ms: Nach dem setzen der Achsfreigabe wird der Start einer Bewegung um diese Zeit verzögert, um das Öffnen einer Motorhaltebremse zu ermöglichen. Bei T#0ms (default-Einstellung) ist die Verzögerung deaktiviert.

4 Funktionsbeschreibung

Name	Datentyp	Beschreibung
tBreakClosingTime	Time	<p>tBreakClosingTime <> T#0ms: Ermöglicht das Schließen einer Motorhaltebremse ohne dass eine hängende Achse absacken kann.</p> <p>Der Befehl _disableAxis am Ende des E-STOP modes wird in zwei Schritten ausgeführt: Schritt 1: Nur AUS1 wird zurückgenommen. Schritt 2: Nach der spezifizierten Zeit werden alle Achsfreigaben zurückgenommen.</p> <p>tBreakClosingTime = T#0ms: Alle Achsfreigaben werden sofort zurückgenommen (default-Einstellung).</p>
eVelocityFormat	'eLPrintVelocityFormatType'	<p>Abhängig von dieser Einstellung wird der Geschwindigkeitssollwert („STDcIO.IN.r64SpeedSetpoint“) interpretiert: DIRECT: Der Sollwert wird in der Einheit der Achse interpretiert. NORMALIZED_WITHOUT_FORMATADAPTION: Prozentuale Auswertung basierend auf „AxisTOConfigData.sInternal.r64DefaultVelocity“ NORMALIZED_WITH_FORMATADAPTION: Prozentuale Auswertung basierend auf „AxisTOConfigData.sInternal.r64DefaultVelocity“ * „AxisData.r64CylinderFormatlengthRatio“</p>
eFaultAckForcedAxis	'eLPrintFaultAckForcedAxisType'	<p>NO: Fehler müssen vom Anwender quittiert werden. AUTO: Fehler werden automatisch quittiert, die Achse synchronisiert wieder auf den Master auf.</p>
tFaultAckForcedEncOutputTime	TIME	<p>Zeitintervall in dem versucht wird anstehende Fehler automatisch zu quittieren. Wirkt nur bei geforcten TM41 Achsen!</p>
eCamSettings	'eLPrintCamSettingsType'	<p>ABSOLUTE_IMMEDIATELY: Der _enableCaming Befehl in der Betriebsart 50 – „CAM_MODE“ wird mit den in der Bibliothek vorbelegten Einstellungen aufgerufen. USER_DEFAULT: Der _enableCaming Befehl wird mit den Default Einstellungen am TO aufgerufen.</p>
ePosAbsDirection	eLPrintPosAbsDirectionType	<p>Mittels des Parameters kann die Richtung bei absoluten Positioniervorgängen von Moduloachsen definiert werden. BY_VALUE: Richtung abhängig vom Vorzeichen des Geschwindigkeitssollwertes r64SpeedSetpoint POSITIVE: Positive Drehrichtung NEGATIVE: Negative Drehrichtung SHORTEST_WAY: Positionieren auf kürzestem Weg POSITIVE_WITH_WINDOW: Positive Drehrichtung, negativ, wenn innerhalb des Fensters NEGATIVE_WITH_WINDOW: Negative Drehrichtung, positiv, wenn innerhalb des Fensters</p>

4 Funktionsbeschreibung

Name	Datentyp	Beschreibung
eTorqueValueSource	eLPrintTorqueValueSourceType	Auswahl Drehmoment Schnittstelle von STDCIO_INTERFACE AXISDATA_INTERFACE EXTERNAL_VALUES
r64PosAbsWindow	LREAL	Wenn die Positionierstrecke innerhalb dieses Fensters liegt, dann wird auf "kürzestem Weg" positioniert, außerhalb des Fensters findet die Positionierung in entsprechender Richtung statt.
r64SpeedWindow	LREAL	Innerhalb dieses Drehzahlfensters wird eine automatische Reduktion der Drehmomentengrenzen für die Funktion Rücklaufsperr durchgeföhrt. Die Einheit entspricht der Einheit der MotionStateData.commandVelocity am TO.
boEnableTorqueFunctions	BOOL	Freischalten der Drehmoment-Funktionen der Achse (FBLPrint_Torquefunctions) im FBLPrint_StdAxislpo
boRestartInitialization	BOOL	Mit steigender Flanke kann die Initialisierungsphase des „FBStdAxisBckgrnd“ neu gestartet werden. Damit können z.B. die Daten aus „AxisTOConfigData.sInternal“ neu belegt werden.

Tabelle 4-4 'sVeloDynamicsType'

Name	Datentyp	Beschreibung
r64Velo	LREAL	Geschwindigkeits-Sollwert
r64Accel	LREAL	Beschleunigungs-Sollwert
r64Decel	LREAL	Verzögerungs-Sollwert
r64AccelJerkStart	LREAL	Verrundung (Ruck) bei Start der Beschleunigung
r64AccelJerkEnd	LREAL	Verrundung (Ruck) bei Stopp der Beschleunigung
r64DecelJerkStart	LREAL	Verrundung (Ruck) bei Start der Verzögerung
r64DecelJerkEnd	LREAL	Verrundung (Ruck) bei Stopp der Verzögerung
eProfile	'EnumProfile'	Dynamik Profil: SMOOTH, TRAPEZOIDAL

Tabelle 4-5 'sAccelDynamicsType'

Name	Datentyp	Beschreibung
r64Accel	LREAL	Beschleunigungs-Sollwert
r64Decel	LREAL	Verzögerungs-Sollwert
r64AccelJerkStart	LREAL	Verrundung (Ruck) bei Start der Beschleunigung
r64AccelJerkEnd	LREAL	Verrundung (Ruck) bei Stopp der Beschleunigung
r64DecelJerkStart	LREAL	Verrundung (Ruck) bei Start der Verzögerung
r64DecelJerkEnd	LREAL	Verrundung (Ruck) bei Stopp der Verzögerung
eProfile	'EnumProfile'	Dynamik Profil: SMOOTH, TRAPEZOIDAL

Tabelle 4-6 'sStopDynamicsType'

Name	Datentyp	Beschreibung
r64DecelJerkStart	LREAL	Verrundung (Ruck) bei Start der Verzögerung
r64DecelJerkEnd	LREAL	Verrundung (Ruck) bei Stopp der Verzögerung
r64Decel	LREAL	Verzögerungs-Sollwert
eProfile	'EnumProfile'	Dynamik Profil: SMOOTH, TRAPEZOIDAL
eAbortAcceleration	EnumAxisAbortAcceleration	Angabe, ob bei Ruck-geführten Stoppen (eProfile = SMOOTH) eine vorhandene Beschleunigung auf 0 gesetzt werden soll. NO: Eine eventuell vorhandene Beschleunigung wird über die Ruckbegrenzung abgebaut. Somit ist eine Geschwindigkeitszunahme, die bis zum Abbau der Beschleunigung anhält, möglich. YES: Eine eventuell vorhandene Beschleunigung wird nicht über die Ruckbegrenzung, sondern sofort abgebaut

Hinweis Die Einheiten der obigen Dynamiken sind abhängig von der entsprechenden Einheit der Achse in der TO-Konfiguration.
Für Gleichlaufachsen ist z.B. [°/s] zu wählen.

Hinweis Die obigen Verrundungswerte (Ruck) werden nur dann verwendet (sind aktiv), wenn am Konfigurationsparameter eProfile = SMOOTH gewählt wird.
Bei eProfile = TRAPEZOIDAL sind nur die Beschleunigungs-Dynamiken aktiv.

sAxisTOConfigDataType

Die Struktur "AxisTOConfigData" wird benutzt um alle Technologie Objekte für die verschiedenen generischen Funktionsblöcke zur Verfügung zu stellen.

Die Struktur ist unterteilt in:

sUser

Die sUser Parameter müssen durch den Anwender verschaltet werden.

sInternal

Die sInternal Paramter werden aufgrund der sUser-Parameter-Verschaltungen innerhalb der Applikation automatisch belegt.

Hier ist keine Anpassung durch den Benutzer notwendig!

Im Funktionsbaustein „FBLPrint_StdAxisBckgrnd“ wird diese Struktur auf Plausibilität überprüft.

4 Funktionsbeschreibung

Tabelle 4-7 sAxisTOConfigDataType

Name	Datentyp	Beschreibung
sUser		
toAxis	ANYOBJECT	Achs-Zuweisung (SIMOTION Achs-Name)
toMeasuringInput	MeasuringInputType	Messtaster-Zuweisung (SIMOTION MI-Name)
toDefaultMaster	ANYOBJECT	default Master Zuweisung; der default Master ist angewählt, wenn u8MasterNumber = 0
toDefaultCam	CamType	default Cam Zuweisung (Kurvenscheibe); die default Cam ist angewählt, wenn u8CamNumber = 0
toSetAxis	FollowingAxis	Achs-Zuweisung (Folgeachse) für Referenziermodus „SET_AXIS_POSITION“
atoPossibleMaster	ARRAY OF ANYOBJECT	Array der „möglichen Master“; der anzuwählende Master wird über „STDcIO.IN.u8MasterNumber“ ausgewählt
atoPossibleCams	ARRAY OF CamType	Array der „möglichen Kurvenscheiben-Master“; der anzuwählende Master wird über „STDcIO.IN.u8CamNumber“ ausgewählt
eDriveType	eLPrintDriveTypeType	STANDARD: drehzahl geregelter Antrieb (default) U_F_CONTROLLED: U/f gesteuerter Antrieb HINWEIS: siehe auch Parameter axisData.eAxisCyclicInterface!
sInternal		
toDriveAxis	DriveAxis	Systemvariablen Struktur des Achs Typs „DriveAxis“ (auch für Pos- und Following-Achsen)
toPositioningAxis	PosAxis	Systemvariablen Struktur des Achs Typs „PosAxis“ (auch für Following-Achsen)
toFollowingAxis	FollowingAxis	System Variablen Struktur des Achs Typs „FollowingAxis“
toFollowingObj	FollowingObjectType	System Variablen Struktur des „Folgeobjekts“ einer Folgeachse
toSetAxisFollObj	FollowingObjectType	System Variablen Struktur des FolgeAchsObjekts. Genutzt zum Setzen einer virtuellen Achse (Referenziermodus „SET_AXIS_POSITION“)
toAdditionObject	_AdditionObjectType	Aktives Addierobjekt der Master Array Liste oder als “default Master“
toFormulaObject	_FormulaObjectType	Aktives Formelobjekt der Master Array Liste oder als “default Master“
eAxisTechnology	eAxisTechnologyType	Diese Variable zeigt den verschalteten Achstyp an: Drive-, Pos-, Following-Axis
eAxisType	eAxisType	Diese Variable zeigt die Art der Achse an: reale Achse oder virtuelle Achse
eTechnology	eTechnologyType	Zeigt an, ob master ein Addierobjekt oder Formelobjekt ist
r64ModuloLength	LREAL	Modulolänge der Achse wie in den Achs Konfigurationsdaten bestimmt; (nur für Positionier- und Folge-Achsen)
boTechnologyData	BOOL	TRUE: Technologiedatenblock ist aktiv
r64NominalTorque	LREAL	Nennmoment der Ache; Vom Antrieb ausgelesen

4 Funktionsbeschreibung

Name	Datentyp	Beschreibung
r64MaxTorque	LREAL	Maximales Drehmoment der Achse; Vom Antrieb ausgelesen
r64MaxVelocity	LREAL	Maximale Drehzahl der Achse; Vom Antrieb ausgelesen
r64DefaultVelocity	LREAL	Default Geschwindigkeit der Achse aus der TO-Konfiguration: userDefaultDynamics.Velocity; Wenn AxisConfigData.eVelocityFormat" = NORMALIZED_WITH_FORMATADAPTION oder NORMALIZED_WITHOUT_FORMATADAPTION, basiert der STDclO.IN.r64SpeedSetpoint prozentual auf dieser Geschwindigkeit.
eTypeOfEncoder	eEncoderType	Typ des verwendeten Motorgebers
eTorqueReference	EnumAxisReferenceMaxNominal	Referenzwert für Drehmoment-Normierung: Max_Value, Nominal_Value
boSimulation	BOOL	TRUE: Achse ist in Simulation
boValid	BOOL	Zeigt an, dass die Initialisierungs der TO config Daten beendet ist. Die Initialisierung kann durch den Anwender über das bit „axisConfigData.boRestartInitialization“ erneut gestartet werden.

sAxisDataType

Die Struktur "AxisData" wird als interne Sollwert und Variablenstruktur genutzt. Dies erlaubt z.B. notwendige Berechnungen von Sollwerten nach Empfang von der STDc Schnittstelle vorzunehmen, bevor die Werte im „FBLPrint_MotionAxis“ wirksam werden.

Der Benutzer kann diese Struktur als Diagnosestruktur verwenden.

Hier ist typischerweise keine Anpassung durch den Benutzer notwendig.

Tabelle 4-8 sAxisDataType

Name	Datentyp	Beschreibung
u16ModeCurrent	UINT	Betriebsmodus, erzeugt durch Auswertung von: <ul style="list-style-type: none"> • STDclO.In.u16ModeNumber • STDclO.In.boOff1Estop • HWOff1Estop • Axisname.actormonitoring.cyclicinterface • boApplicationError
r64SpeedSetpoint	LREAL	Interner Drehzahl Sollwert, erzeugt durch: <ul style="list-style-type: none"> • STDclO.In.boEnableSpeedSetpoint • STDclO.In.boSetpointMode • STDclO.In.r64SpeedSetpoint • STDclO.In.i32GearNumSetpoint • STDclO.In.i32GearDenSetpoint
r64PositioningSetpoint	LREAL	Interner Positionier Sollwert (z.B. in [°])
r64RegisterSetpoint	LREAL	Interner Register Sollwert (z.B. in [°])
i32GearNumSetpoint	DINT	Interner Getriebe Faktor (Zähler)
i32GearDenSetpoint	DINT	Interner Getriebe Faktor (Nenner)

4 Funktionsbeschreibung

Name	Datentyp	Beschreibung
toCurrentMaster	'MasterType'	aktiver Master, wie angewählt mittels STDclO.In.u8MasterNumber, nach Konsistenz Prüfung
toCurrentCam	'CamType'	Aktive Kurvenscheibe, wie angewählt mittels STDclO.In.u8CamNumber, nach Konsistenz Prüfung
r64CalcDiffSync	LREAL	Errechnete Synchronisier Lage Abweichung zwischen Master und Slave Achse (z.B. [°]), berechnet in „FBLPrint_StdAxisIPO“
r64CylinderFormatLengthRatio	LREAL	Faktor zwischen Referenz Format und aktuellem Zylinder Format, nach Konsistenz Prüfung
r64ActiveSynchAxisCycle	LREAL	Interne variable zu AxisConfigData.sSyncConfig.r64SynchAxisCycle
boAdditiveTorqueActive	BOOL	TRUE, wenn TO-axis.additivetorqueIN.state = ACTIVE
boTorqueLimitsActive	BOOL	TRUE, wenn TO-axis.torqueLimitNegativeIN.state = ACTIVE OR TO-axis.torqueLimitPositiveIN.state = ACTIVE
boChangeCam	BOOL	Anforderung Wechsel Cam
boChangeMaster	BOOL	Anforderung Wechsel Master
eAxisCyclicInterface	EnumActiveInactive	Status der zyklischen Schnittstelle zwischen Antrieb und Achse. Wenn axisTOConfigData.sUser.eDriveType = STANDARD, wird der Parameter intern verschaltet. Wenn axisTOConfigData.sUser.eDriveType = U_F_CONTROLLED, muss der Parameter vom Benutzer auf ACTIVE gesetzt werden, sobald der Antrieb hochgelaufen ist.
sFormatAdaption	sLPrintFormatAdaptionType	Unterstruktur zur Anpassung des Getriebefaktors durch PrintAddOn Insetting
sWinderControl	sLPrintWinderControlType	Unterstruktur für den Betrieb einer Wickelachse
sTorqueValues	sLPrintTorqueValuesType	Unterstruktur Drehmomentschnittstelle
sAxisCW1	'sLPrintT105STW1Type'	Antriebssteuerwort 1
sAxisCW2	'sLPrintT105STW2Type'	Antriebssteuerwort 2
sAxisSW1	'sLPrintT105ZSW1Type'	Antriebsstatuswort 1
sAxisSW2	'sLPrintT105ZSW2Type'	Antriebsstatuswort 2
sAxisSafetyState1	'sLPrintSafetyState1Type'	Safety Statutswort
sStateLog	'sStateLogType'	Informationen über aktuelle Betriebsart und State; State-Log-Puffer
sAppMsgLog	'sApplicationMessageLogType'	Informationen über aktuelle Fehler, Warnungen, Fehlernummer und Warnungsnummer; Fehler-Puffer

4 Funktionsbeschreibung

Tabelle 4-9 'sStateLogType'

Name	Datentyp	Beschreibung
eActMode	'eLPrintModeType'	Aktuelle Betriebsart (z.B. GEAR_MODE)
eActState	'eLPrintStateType'	Aktueller State innerhalb einer Betriebsart
u16LogPointer	UINT	Zeiger auf aktuellen/letzten Eintrag im State-Log-Puffer „asLog“
asLog	'ARRAY OF sStateArrayType'	State-Log-Puffer mit „Mode“, „State“, Zeitstempel und zwei Zusatzwerten „i32CmdState“ und „i32AddValue“, welche Auskunft über Error-ID's und Rückgabewerten von Systemfunktionen liefern

Tabelle 4-10 'sApplicationMessageLogType'

Name	Datentyp	Beschreibung
boAppError	BOOL	Ein Applikationsfehler steht an
u16AppErrorNo	UINT	Applikationsfehlernummer (siehe Kapitel 6)
boAppWarning	BOOL	Eine Applikationswarnung steht an
u16AppWarningNo	UINT	Applikationswarnungsnummer (siehe Kapitel 6)
u16LogPointer	UINT	Zeiger auf aktuellen/letzten Eintrag im Fehler-Puffer „asLog“
asLog	'ARRAY OF sMsgInfoType'	Fehler-Puffer mit Fehlertyp, Fehlernummer und Zeitstempel

Tabelle 4-11 sFormatAdaptionType

Name	Datentyp	Beschreibung
eAdaptionMode	eLPrintGearAdaptionModeType	ACTIVE: Getriebeanpassung durch Insetting freigeben
i32GearBasicFactor	DINT	Faktor zu Anpassung der Auflösung des additiven Getriebesollwertes
i32AddGearNumSetpoint	DINT	Additiver Getriebesollwert Zähler
i32AddGearDenSetpoint	DINT	Additiver Getriebesollwert Nenner

Tabelle 4-12 sWinderControlType

Name	Datentyp	Beschreibung
boWinderEnable	BOOL	Signal Wicklerbaustein aktiv
boWinderCtrlEnable	BOOL	Signal Wickler Zugregelung aktiv
boSpliceDone	BOOL	Signal Splice durchgeführt; Bei TRUE wird Wickelachse abhängig von eAfterSpliceMode gestoppt.
eAfterSpliceMode	eLPrintAfterSpliceModeType	STOP_AFTER_SPLICE: Achse wird gestoppt und bleibt in Regelung. REWIND_AFTER_SPLICE: Achse wird gestoppt und direkt im Anschluss in Entgegengesetzt Richtung verfahren.

4 Funktionsbeschreibung

Tabelle 4-13 sTorqueValuesType

Name	Datentyp	Beschreibung
r64AdditiveTorque	LREAL	[Nm] Zusatzdrehmoment – wird intern geschrieben auf TOAchse.defaultAdditiveTorque
r64UpperTorqueLimit	LREAL	[Nm] Obere Drehmomentgrenze – wird intern geschrieben auf TOAchse.defaultTorqueLimitPositive
r64LowerTorqueLimit	LREAL	[Nm] Untere Drehmomentgrenze – wird intern geschrieben auf TOAchse.defaultTorqueLimitNegative

sAxisDataRetainType

Die "AxisDataRetain" Struktur enthält Werte die nach Spannung aus/ein (d.h. Verlust der 24V Spannungsversorgung der SIMOTION CPU) wieder zur Verfügung stehen sollen/müssen.

Tabelle 4-14 sAxisDataRetainType

Name	Datentyp	Beschreibung
r64ActiveRegisterSetpoint	LREAL	Letzter „Gesamt Register Wert“, nur für following Achsen [°]
boReferencingDone	BOOL	Internes Bit. Nicht zur Auswertung in Benutzerapplikation geeignet! Zeigt bei realen Achsen mit Absolutwertgeber (mit Inkrementalwertgeber nach Speichern der Position) an ob die Achse bereits Referenziert wurde. Vorsicht: Nach Motor bzw. Gebertausch muss axisDataRetain.boReferencingDone manuell zurückgesetzt werden!
r64MeasuredValueRefOnTheFly	LREAL	Abstand Achsposition von der Gebernullmarke [°] nach passiven referenzieren
boRefValueSaved	BOOL	Gültiger Abstand zur Gebernullmarke gespeichert

4.5 Print Standard Achs Betriebsarten

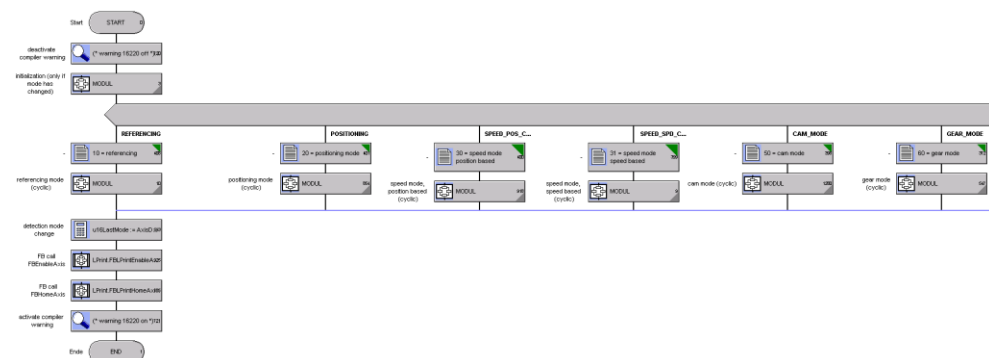
Der SIMOTION Print Standard bietet eine modulare Achs Betriebsarten Steuerung. Jede Achse kann völlig unabhängig von anderen Achsen verschiedene Motion Control Funktionen ausführen, aber alle Achsen nutzen dennoch dafür die exakt selben Funktionsbausteine. Zusätzlich nutzen alle Achsen auch die gleichen Variablen-Struktur-Typen und Schnittstellen. Jede Achse legt ihre eigene unabhängige Schnittstelle zur SPS (STDcIO) an.

4.5.1 Überblick

Diese Version des Print Standard hat die folgenden implementierten Standard Achs/Antriebs Betriebsarten. Alle Motion Control Befehle werden nur in einem Funktionsbaustein "FBLPrint_MotionAxis" in Motion Control Chart programmiert:

- COAST_STOP mode # 0
- E_STOP mode # 1
- MAX_TORQUE_STOP mode # 2
- REFERENCING mode # 10
- POSITIONING mode # 20
- SPEED_POS_CTRL mode # 30
- SPEED_SPD_CTRL mode # 31
- WINDING mode # 40
- SPLICE_TO_WINDING mode # 41
- SPLICE_TO_STOP mode # 42
- CAM_MODE mode # 50
- GEAR_MODE mode # 60
- SPEED_GEAR_MODE mode # 61
- ERROR mode # 100
- ERROR_TO_E_STOP mode # 101
- ERROR_TO_COAST_STOP mode # 102

Abbildung 4-2 Ausschnitt aus dem MCC Funktionsbaustein "FBLPrint_MotionAxis" (Betriebsartenbaustein)



Hinweis Falls eine Betriebsart benötigt wird welche derzeit nicht im Print Standard verfügbar ist, kann sehr schnell und einfach eine neue Betriebsart erzeugt und hinzugefügt werden. Dies geschieht durch Erweiterung des Funktionsbausteins „FBLPrint_MotionAxis“ um einen neuen Zweig.

Bedingungen zum Betriebsarten-Wechsel

Im Allgemeinen ist es möglich zu jederzeit zwischen den verschiedenen Betriebsarten zu wechseln, sofern die Achse die angewählte Betriebsart unterstützt.

Unterstützt die Achse die angewählte Betriebsart, wird der aktuelle Bewegungsbefehl abgebrochen und mit dem Bewegungsbefehl der neuen Betriebsart begonnen.

Wir die angewählte Betriebsart von dem Achs-Typ nicht unterstützt, wird ein Applikationsfehler ausgegeben, da die Achse die geforderte Betriebsart nicht erfüllen kann. (z.B. POS-Achse wird in GEAR_MODE geschaltet)

Tabelle 4-15 Mögliche Betriebsarten abhängig vom SIMOTION Achs Typ

Mode	Real Drive Axis	Real Positioning Axis	Real Following Axis	Virtual Positioning Axis	Virtual Following Axis
COAST_STOP	ja	ja	ja	n/m	n/m
E_STOP	ja	ja	ja	ja	ja
MAX_TORQUE_STOP	ja	ja	ja	n/m	n/m
REFERENCING	n/v	ja	ja	ja	ja
POSITIONING	n/v	ja	ja	ja	ja
SPEED_SPD_CTRL	ja	ja	ja	ja	ja
SPEED_POS_CTRL	n/v	ja	ja	ja	ja
WINDING	ja	ja	ja	n/m	n/m
SPLICE_TO_WINDING	n/m	ja	ja	n/m	n/m
SPLICE_TO_STOP	n/m	ja	ja	n/m	n/m
CAM_MODE	n/v	n/v	ja	n/v	ja
GEAR_MODE	n/v	n/v	ja	n/v	ja
SPEED_GEAR_MODE	n/v	n/v	ja	n/v	ja
ERROR	ja	ja	ja	ja	ja
ERROR_TO_ESTOP	ja	ja	ja	ja	ja
ERROR_TO_COAST_STOP	ja	ja	ja	ja	ja

n/v: nicht verfügbar, n/m: nicht möglich

ALLOWED_MODE_ARRAY

Im Print Standard sind in der Bibliotheksquelle „cVarConf“ verschiedene sog. „Mode Arrays“ definiert:

- DRIVE_AXIS_MODE_ARRAY
- POSITIONING_AXIS_MODE_ARRAY
- FOLLOWING_AXIS_MODE_ARRAY
- EXTENC_MODE_ARRAY
- WINDING_DRIVE_AXIS_MODE_ARRAY
- WINDING_POS_AXIS_MODE_ARRAY
- GEAR_MODE_ARRAY
- CAM_MODE_ARRAY

Innerhalb dieser Array ist definiert, welche Betriebsarten mit diesem „Mode Array“ erlaubt sind.

z.B. sind mit „DRIVE_AXIS_MODE_ARRAY“ nur die Betriebsarten
„E_STOP“
„COAST_STOP“
„MAX_TORQUE_STOP“
„SPEED_SPD_CTRL“
erlaubt.

Eines dieser „Mode Arrays“ muss dem Konfigurationsparameter
„AxisConfigData.au16AxisAllowedModeArray“ übergeben werden.

Achs-Freigaben, „E_STOP“ und Ausschalten

Wurde die Betriebsart „E_STOP“ angewählt, nimmt die Achse keine weiteren Betriebsartenwechsel mehr an, sondern muss ausgeschaltet werden (disableAxis). Das Ausschalten findet in der Betriebsart „E_STOP“ nach Ablauf der E-Stop Verzögerung automatisch statt. Das bedeutet, dass bewusst aus Sicherheitsgründen, während der Verzögerung im E-Stop kein Betriebsartenwechsel oder neuer Motion Befehl möglich ist.

Achs Freigabe Befehle sind im Print Standard aus SPS Sicht nicht notwendig, da diese in den jeweiligen Betriebsarten des Print Standard bereits integriert sind. D.h. die Achsen werden bei einem Betriebsartenwechsel automatisch nach Bedarf freigegeben. DisableAxis sind beinhaltet in den Modi „COAST_STOP“, „E_STOP“, „MAX_TORQUE_STOP“, „REFERENCING (ACTIVE)“ und „ERROR“. Zudem können die Achsen aufgrund von SIMOTION System Fehlern ausgeschaltet werden.

Wird eine ausgeschaltete Achse in eine Betriebsart geschaltet, in der Bewegungen ausgeführt werden können, wird die neue Betriebsart erst aktiv wenn die Achse eingeschaltet ist.

Eine Achse ist oder wechselt in die Betriebsart „E_STOP“ und wird ausgeschaltet, falls vorgegeben über die STDcIO Standard Schnittstelle (u16ModeNumber) oder aufgrund einer der folgenden Bedingungen:

- das Schnittstellen Bit „boOff1Estop“ ist FALSE

- das optionale Applikationssignal „HWOFF1Estop“ ist FALSE
- die Systemvariable „axisname.actormonitoring.cyclicinterface“ ist INACTIVE (Kommunikationsproblem zwischen SIMOTION Achse und dem zugehörigen SINAMICS Antrieb)

HWOFF1Estop

Optional kann ein „HWOFF1Estop“ Klemmensignal auf einen 24V Eingang verdrahtet werden um einen E-Stop auszulösen. Als Vorbelegung ist diese Funktion deaktiviert (vorbelegt mit TRUE), damit ein Betrieb des Print Standard ohne 24V Verdrahtung möglich ist.

Sonderfall des Umschaltens von/nach „SPEED_SPD_CTRL“ bei Positionier/Folgeachsen:

In einigen Fällen kann es technologisch notwendig sein, dass eine Positionier/Folgeachse auf rein drehzahlgeregelten Betrieb umschaltet werden muss (Betriebsart „SPEED_SPD_CTRL“). In diesem Fall wird sich ein Lagefehler aufbauen da der Lageregler und Lageüberwachungen gesperrt sind.

VORSICHT

Wird die Achse im Betrieb auf eine lagegeregelt Betriebsart zurück geschaltet, kann es zu einem deutlichen Sollwert-Sprung an der Achse führen.

Um dies zu vermeiden sollten die Positionier/Folgeachsen nach „SPEED_SPD_CTRL“ Betrieb ausgeschaltet werden (Betriebsart „E_STOP“).

Start Bit Funktionalität

Ab Print Standard V2.2.1.0 sind alle Basis-Betriebsarten in denen Bewegungen ausgeführt werden mit Start-Bit Funktionalität. D.h. bei Anwahl einer Betriebsart werden alle vorbereitenden Schritte ausgeführt, die eigentliche Bewegung z.B. Positionieren, Getriebegleichlauf starten etc. wird aber erst mit der positiven Flanke am Start-Bit (STDclO.IN.boStartMotion) gestartet.

Mit der Start Bit Funktionalität wird lediglich die Grundbewegung der Achse gestartet, beispielsweise in „GEAR_MODE“ das Aufsynchronisieren mit dem aktuellen Getriebefaktor auf den Master. Alle der Grundbewegung folgenden Funktionalitäten wie z.B. Winkelsynchronität herstellen, Register verfahren, Änderung des Getriebefaktors oder des Geschwindigkeitssollwertes bleiben unabhängig.

Ist diese Funktionalität nicht erwünscht (Funktionalität der Print Standard Versionen <V2.2.1.0), muss das Bit Default mäßig auf TRUE gesetzt werden.

4.5.2 Mode 0 – COAST_STOP

In diesem Modus wird ein AUS2 Kommando an den Antrieb gesendet d.h. der Antrieb „trudelt aus“. Dieser Modus kann von jeder realen Achse (drive-, positioning- und following- axis) genutzt werden.

Dieser Modus kann nicht von virtuellen Achsen genutzt werden!

Hinweis Der Modus sollte nicht unterbrochen werden, da ansonsten ein undefiniertes Einschalten auf die evtl. noch drehende Achse ausgeführt wird.

4.5.3 Mode 1 - E_STOP

Die Achse wird stillgesetzt mit der definierten E-Stop Rampe und Verrundungszeiten. Während des Stillsetzens nimmt die Achse keine neuen Betriebsart/Motion-Befehle an. Nur bei Stillstand und nach dem automatischen Ausschalten kann die Achse in eine neue Betriebsart geschaltet werden.

Die „E-Stop“-Zeitprofile Hoch/Rücklaufzeit und Verrundungen können in der Struktur „AxisConfigData“ eingestellt werden.

4.5.4 Mode 2 - MAX_TORQUE_STOP

Dieser Modus implementiert einen Not-Halt mit maximal zulässiger Verzögerung. Die Achse verzögert an den eingestellten Strom/Drehmoment Grenzen des Antriebs (SINAMICS).

VORSICHT Bei diesem Stopp ist ein Reißen der Bahn möglich!

VORSICHT Die Mechanik kann aufgrund der maximalen Verzögerungen beschädigt werden!

Möglicherweise müssen die Strom/Drehmoment-Grenzen des Antriebs(SINAMICS) verringert werden.

Dieser Modus kann nicht von virtuellen Achsen genutzt werden!

Hinweis Der Modus sollte nicht unterbrochen werden, da ansonsten ein undefiniertes Einschalten auf die evtl. noch drehende Achse ausgeführt wird.

4.5.5 Mode 10 - REFERENCING

Der Referenziermodus kann sowohl für reale als auch virtuelle Positionier-/ Folge-Achsen angewählt werden. Drehzahl-Achsen können nicht in diesen Modus geschaltet werden!

Folgende Referenzier-Modi stehen zur Verfügung:

- ACTIVE (nur für reale Achsen)
- PASSIVE (nur für reale Achsen)
- REF_ON_THE_FLY (nur für reale Achsen)
- SET_AXIS_POSITION (nur für virtuelle Achsen)

Die Referenziermodi können über die Konfig-Variable „AxisConfigData.sHoming.eReferencingMode“ angewählt werden.

ACTIVE

Dieser Modus kann verwendet werden, um die Position einer realen Positionier- oder Gleichlaufachse durch eine Referenzierbewegung zu setzen.

Mit steigender Flanke des Bits „STDcIO.IN.boStartMotion“ startet die Achse eine Referenzierbewegung. Voraussetzung ist

- STDcIO.IN.boEnableSpeedSetpoint = TRUE und
- STDcIO.IN.r64SpeedSetpoint \neq 0

Für die Bewegung werden die Dynamiken „AxisConfigData.sLocalDyn“ verwendet.

Achse mit Absolutwertgeber:

Bei Verwendung eines Absolutwertgebers (AxisTOConfigData.sInternal.eTypeOfEncoder = ABSOLUTE) muss ein TO Messtaster an der Achse parametrierbar sein!

Bei einem Trigger-Ereignis des Messtasters stoppt die Achse automatisch. Die Position der Achse wird auf den Wert der Variable „AxisConfigData.sHoming.r64HomePosition“ gesetzt.

Hinweis

Wird nach mehr Umdrehungen als „axisConfigData.sHoming.u16MeasInMaxNumOfRev“ kein Messereignis ausgelöst, wird eine Applikationswarnung ausgegeben. Default-Einstellung = 1 Umdrehung.

Achse mit Inkrementalwertgeber:

Bei Verwendung eines Inkrementalwertgebers (AxisTOConfigData.sInternal.eTypeOfEncoder = INCREMENTAL) wird die Position beim Überfahren der Gebernulldarstellung auf die Position „AxisConfigData.sHoming.r64HomePosition“ gesetzt. Die Achse stoppt automatisch.

ACHTUNG

Reichen die Achs-Dynamiken (sLocalDy) nicht aus, um die Achse an der Nullmarke zu stoppen, reversiert die Achse bei Einstellung des Achs-Konfigurationsparameters TypeOfAxis.homing.direction \neq „Only positive/negative direction“ um auf die gesetzte Position zurückzufahren.

Mit TypeOfAxis.homing.direction = Only positive/negative direction erfolgt ggf. eine weitere Umdrehung um die zu setzende Position nach dem Überfahren zu erreichen.

PASSIVE

Dieser Modus kann verwendet werden, um die Position einer realen Positionier- oder Gleichlaufachse im Stillstand zu setzen.

Der Mode Referencing wird durch `boOff1EStop = TRUE` aktiv, jedoch wird nicht die Achsfreigabe gegeben. Die Achse bleibt ein- oder ausgeschaltet.

Die Funktionalität wird mit steigender Flanke des Bits „`STDcIO.IN.boStartMotion`“ gestartet.

Achse mit Absolutwertgeber:

Die Achse bewegt sich nicht. Die Position der Achse wird auf den Wert der Variablen „`AxisConfigData.sHoming.r64HomePosition`“ gesetzt.

Achse mit Inkrementalgeber:

Die Achse bewegt sich nicht. Die Position der Achse wird auf den Wert der Variablen „`AxisConfigData.sHoming.r64HomePosition`“ gesetzt.

Speichern der Inkrementalgeberposition:

Ab Print Standard Version \geq V2.2.1.0 besteht die Möglichkeit, die Position des Inkrementalgebers nach passivem Referenzieren über eine interne oder externe Geber-Nullmarke Spannungsausfallsicher in einer Retain-Variablen zu speichern.

Hinweis

Die Funktionalität des Speicherns der Position über die interne Nullmarke ist nur bei einer 1:1 Kopplung zwischen Motor und Mechanik verwendbar. Es muss sichergestellt sein, dass sich die mechanischen Verhältnisse nach Spannung AUS/EIN nicht verändern!

Das Speichern und Wiederherstellen der Position erfolgt in drei Schritten:

1. Passives Referenzieren

- `STDcIO.IN.u16ModeNumber = 10` (REFERENCING)
Mit steigender Flanke am Bit „`STDcIO.IN.boStartMotion`“ wird die Position der Achse auf den Wert der Variablen „`AxisConfigData.sHoming.r64HomePosition`“ gesetzt. Die Achse bewegt sich dabei nicht.

2. Speichern der Position (wenn „`AxisDataRetain.boRefValueSaved`“ = FALSE)

Um das Speichern des Abstands der Referenzier-Position zur Nullmarke durchzuführen müssen folgende Variablen gesetzt sein:

- `STDcIO.IN.u16ModeNumber = 30` oder `60`
- `AxisConfigData.sHoming.eReferencingMode = PASSIVE`

Die Funktionalität wird mit steigender Flanke des Bits „`STDcIO.IN.boReferencingOnTheFly`“ gestartet.

Eine Achsbewegung ist notwendig, da die Nullmarke (intern oder extern) überfahren werden muss. Die Achse misst den Abstand und speichert diesen in der Variablen „`AxisDataRetain.r64MeasuredValueRefOnTheFly`“.

3. Wiederherstellen der Position (wenn „AxisDataRetain.boRefValueSaved“ = TRUE)

Ist ein gespeicherter Wert nach dem Hochlaufen der Steuerung vorhanden wird dies über die Variable „AxisDataRetain.boRefValueSaved“ angezeigt.

Um die Position nach Hochlauf der Steuerung wiederherzustellen, müssen folgende Variablen gesetzt sein:

- STDclO.IN.u16ModeNumber = 30 oder 60
- AxisConfigData.sHoming.eReferencingMode = PASSIVE

Die Funktionalität wird mit steigender Flanke des Bits „STDclO.IN.boReferencingOnTheFly“ gestartet.

Eine Achsbewegung ist notwendig, da die Nullmarke (intern oder extern) überfahren werden muss. Bei überfahren der Nullmarke wird die Achs-Position gesetzt.

Hinweis

Um eine neue Distanz zwischen Referenzier-Position und Gebernulmarke zu speichern, muss die Retain-Variable „AxisDataRetain.boRefValueSaved“ per Hand auf FALSE zurückgesetzt werden!

Je nachdem ob auf die interne Geber-Nullmarke oder eine externe Nullmarke über einen Digitaleingang referenziert werden soll, muss dies an der Achse und im Antrieb konfiguriert werden. Die einzustellenden Variablen sind in folgender Tabelle zusammengefasst:

Tabelle 4-16 Achs/Antriebskonfiguration passives Referenzieren

Parameter / Achskonfigurationsvariable	Wert	Beschreibung
SIMOTION – TO Axis: TypeOfAxis.NumberOfEncoders.Encoder1.IncHomingEncoder.PassiveHomingMode	ZM_PASSIVE[2] or CAM_PASSIVE[3]	Referenziermodus: [2]: Geber-Nullmarke [3]: Externe Nullmarke
SINAMICS – Antrieb: p495	Kein Nullmarkenersatz (Gebernulmarke) [0] or DI/DO 9..15 [1..6]	Nullmarkenersatz Eingangsklemme, Geber 1 [0]: Geber-Nullmarke [1..6]: Digital Eingang 1..6

REF_ON_THE_FLY

Dieser Modus kann verwendet werden, um die Position einer realen Positionier- oder Gleichlaufachse parallel zu einer Bewegung (Mode 30 oder Mode 60) auf den Wert „AxisConfigData.sHoming.r64HomePosition“ zu setzen.

Die Funktionalität wird gestartet, in dem das Bit „STDclO.IN.boReferencingOnTheFly“ einer Achse im Mode 30 oder Mode 60 gesetzt wird und der Referenzier-Modus REF_ON_THE_FLY ausgewählt ist.

Hinweis Wird das Bit „STDcIO.IN.boReferencingOnTheFly“ im Mode 20 (REFERENCING) gesetzt, wird der Fehler 1027 (LPRINT_ERR_REF_NO_OR_INVALID_MODE_SELECTED) ausgegeben.

Achse mit Absolutwertgeber:

Bei Verwendung eines Absolutwertgebers (AxisTOConfigData.sInternal.eTypeOfEncoder = ABSOLUT) muss ein TO Messtaster an der Achse parametrierbar sein!

Bei einem Trigger-Ereignis des Messtasters wird die Position der Achse auf die Position „AxisConfigData.sHoming.r64HomePosition“ gesetzt.

Achse mit Inkrementalwertgeber:

Bei Verwendung eines Inkrementalgebers (AxisTOConfigData.sInternal.eTypeOfEncoder = INCREMENTAL) wird die Position beim Überfahren der Gebernullmarke auf die Position „AxisConfigData.sHoming.r64HomePosition“ gesetzt.

Hinweis Soll die Achse im GEAR_MODE fliegend referenziert werden ist das absolute synchronisieren und das absolute Register ab dem Zeitpunkt an dem die Variable boReferencingOnTheFly = TRUE ist gesperrt. Diese Funktionen können erst bei boReferencingOnTheFly = FALSE wieder genutzt werden.

SET_AXIS_POSITION

Dieser Modus kann verwendet werden, um die Position einer virtuellen Achse auf...

- die aktuelle Position einer anderen Achse oder
- auf den Wert „AxisConfigData.sHoming.r64HomePosition“

zu setzen.

Hinweis Wird eine reale Achse in den Mode SET_AXIS_POSITION geschaltet, wird der Fehler 1025 (LPRINT_ERR_REF_NO_VIRTUAL_AXIS) ausgegeben.

Die Funktionalität wird jeweils mit dem Bit „STDcIO.IN.boStartMotion“ gestartet.

VA Position auf aktuelle Position einer anderen Achse setzen:

Diese Funktionalität kann z.B. verwendet werden, einen Globalen Master auf die Position einer Schnittmesser-Achse bei/vor Maschinenstart zu setzen. Dadurch wird die Notwendigkeit einer Synchronisierbewegung der Schnittmesser-Achse vermieden.

Hierfür muss die Konfig-Variable „AxisTOConfigData.sUser.toSetAxis“ versorgt werden.

Die Position der virtuellen Achse wird auf die Position der Achse „toSetAxis“ gesetzt. Die virtuelle Achse bewegt sich dabei nicht, sondern wird sofort in einem Abtastzyklus gesetzt.

ACHTUNG	Es ist zu beachten, dass zum Setzzeitpunkt keine realen Achsen auf die zu setzende virtuelle Achse gekuppelt (z.B. Mode 60 – GEAR_MODE) sind. Ansonsten entsteht an den realen Achsen ein Sollwertsprung!
----------------	--

VA Position auf „r64HomePosition“ setzen:

Ist der Modus SET_AXIS_POSITION angewählt und in der Struktur „AxisTOConfigData.sUser.toSetAxis“ keine andere Achse angegeben (toSetAxis = TO#NIL), wird die virtuelle Achse auf die Position in der Variablen „AxisConfigData.sHoming.r64HomePosition“ gesetzt.

4.5.6 Mode 20 - POSITIONING

Diese Betriebsart kann für reale und virtuelle Positionier- und Folge-Achsen ausgewählt werden. Wird eine Drehzahl-Achse in diese Betriebsart geschaltet, wird ein Applikationsfehler ausgegeben.

Die Dynamiken während des Positionierens sind definiert in „AxisConfigData.sPosDyn“.

Die folgenden Schnittstellen-Parameter legen die Funktion fest:

- u16ModeNumber = 20
- boRelativePositioning = TRUE / FALSE
- boEnableSpeedSetpoint = TRUE
- r64SpeedSetpoint $\neq 0$
- r64PositioningSetpoint = x [°]

Über die Variable „boRelativePositioning“ kann ausgewählt werden, ob der „r64PositioningSetpoint“ absolut oder relativ verwendet wird. Falls TRUE ist die nächste Positionierung relativ zur derzeitigen Position der Achse. Falls FALSE wird der Sollwert absolut interpretiert. d.h. der Bezug ist zur Null Position der Achse.

Die Positionierungen nutzen die Variable „r64SpeedSetpoint“ als Drehzahlvorgabe. Der Drehzahlsollwert muss freigegeben („boEnableSpeedSetpoint“) und ungleich Null sein wenn die Positionierung gestartet wird, ansonsten gibt die Applikation eine Warnung aus und die Bewegung wird nicht gestartet.

Hinweis

Die Einheit des Drehzahlsollwerts basiert auf dem Konfigurationsparameter „AxisConfigData.eVelocityFormat“.

DIRECT: Wert wird direkt interpretiert. Die Einheit entspricht der Einheit am TO.

NORMALIZED_WITHOUT_FORMATADAPTION:

Drehzahlsollwert in [%] von „AxisTOConfigData.sInternal.r64DefaultVelocity“

NORMALIZED_WITH_FORMATADAPTION:

Drehzahlsollwert in [%] von „AxisTOConfigData.sInternal.r64DefaultVelocity“ * „AxisData.r64CylinderFormatLengthRatio“

Siehe dazu auch Kapitel [Anpassung der dynamischen Daten bei Format-Variablen Maschinen](#)

Die Positionierbewegung wird durch eine positive Flanke der Variablen „boStartMotion“ gestartet. Wechselt das Bit während der Bewegung von TRUE nach FALSE wird die Positionierung sofort gestoppt.

Im relativen Positioniermodus kann durch eine erneute positive Flanke des Bits mehrere Male um den gleichen Weg verfahren werden.

Wird der Print Standard ohne start bit Funktionalität verwendet und die Achse soll mehrere relative Positionierungen nacheinander durchführen, wird folgendes Vorgehen vorgeschlagen:

- r64PositioningSetpoint = 10° -> bewegt die Achse um 10°

- $r64PositioningSetpoint = 0^\circ$ -> bewegt die Achse um 0°
- $r64PositioningSetpoint = 10^\circ$ -> bewegt die Achse um 10°

Wird die Achse von einer anderen Betriebsart mit Drehzahl ungleich Null in den Positionier Modus geschaltet während „ $r64SpeedSetpoint$ “ Null ist, wird die interne $commandVelocity$ des Achs-TO's verwendet um die erste Position anzufahren. Diese Funktionalität wird z.B. verwendet wenn der Plattenzylinder vom Druckbetrieb in einer Bewegung auf die erste Plattenwechsel Position stillgesetzt werden soll.

Der Drehzahlsollwert ($r64SpeedSetpoint$) kann während eines Positioniervorgangs fliegend verändert werden. Bei absoluter Positionierung wird die Restlänge mit der neuen Geschwindigkeit verfahren.

Bei relativer Positionierung wird erneut um die gesamte relative Länge verfahren (die Positionierlänge wird dadurch größer!).

Für die Geschwindigkeitsanpassung werden die Positionier-Dynamiken „ $AxisConfigData.sPosDyn$ “ verwendet.

Auch der Positionssollwert kann fliegend verändert werden. Die Achse verwendet immer die letzte vorgegebene Zielposition bzw. relative Länge.

Drehrichtung beim Positionieren:

Beim relativen Positionieren definiert das Vorzeichen des Position Sollwertes „ $r64PositioningSetpoint$ “ die Positionierrichtung:

- $r64PositioningSetpoint > 0$ = positive Produktionsrichtung
- $r64PositioningSetpoint < 0$ = negative Produktionsrichtung

Beim absoluten Positionieren von Modulo-Achsen ist die Richtung abhängig vom Parameter $axisConfigData.ePosAbsDirection$.

- **BY_VALUE:** wird der Drehzahlsollwert „ $r64SpeedSetpoint$ “ genutzt um die gewünschte Drehrichtung vorzugeben.
 - $r64SpeedSetpoint > 0$ = positive Produktionsrichtung
 - $r64SpeedSetpoint < 0$ = negative Produktionsrichtung
- **POSITIVE:** Positionieren in positive Richtung
- **NEGATIVE:** Positionieren in negative Richtung
- **SHORTEST_WAY:** Positionieren auf kürzestem Weg
- **POSITIVE_WITH_WINDOW:** Positionierung in positive Richtung. Liegt die Sollposition innerhalb des Fensters, wird auf kürzestem Weg positioniert.
- **NEGATIVE_WITH_WINDOW:** Positionierung in negative Richtung. Liegt die Sollposition innerhalb des Fensters, wird auf kürzestem Weg positioniert.

Für die Positionierung mit Fenster kann die Fenstergröße über die Variable „ $AxisConfigData.r64PosAbsWindow$ “ eingestellt werden.

VORSICHT	<p>Wird ein Bewegungsbefehl (z.B. „SPEED_POS_CTRL“) durch eine relative Positionierung abgelöst, kann es zum reversieren der Achse kommen!</p> <p>Dies tritt auf, wenn die Positionier-Dynamiken nicht ausreichen (zu klein sind), um die vorher anstehende Geschwindigkeit innerhalb der relativen Positionierlänge abzubauen.</p>
-----------------	--

4.5.7 Mode 30 – SPEED_POS_CTRL

Diese Betriebsart wird genutzt um einen lokalen (lagegeregelten) Drehzahl- oder Tipp-Betrieb zu implementieren, sowie die Maschinengeschwindigkeit am GM zu generieren.

Dieser Modus funktioniert nur mit Positionier- oder Folge-Achsen. (Der Drehzahl Modus von Drehzahl-Achsen ist „SPEED_SPD_CTRL“)

Die Achse wird lagegeregelt betrieben und nutzt den Drehzahlsollwert um damit intern auch einen Lagesollwert zu erzeugen. Bei virtuellen Achsen ist die angezeigte Lageabweichung immer Null, da definitionsgemäß bei virtuellen Achsen Sollwert und Istwert identisch sind.

Der lokale Drehzahlmodus kann wie folgt implementiert werden:

- u16ModeNumber = 30
- boEnableSpeedSetpoint = TRUE
- r64SpeedSetpoint <> 0

Hinweis

Die Einheit des Drehzahlsollwerts basiert auf dem Konfigurationsparameter „AxisConfigData.eVelocityFormat“.

DIRECT: Wert wird direkt interpretiert. Die Einheit entspricht der Einheit am TO.

NORMALIZED_WITHOUT_FORMATADAPTION:

Drehzahlsollwert in [%] von „AxisTOConfigData.sInternal.r64DefaultVelocity“

NORMALIZED_WITH_FORMATADAPTION:

Drehzahlsollwert in [%] von „AxisTOConfigData.sInternal.r64DefaultVelocity“ * „AxisData.r64CylinderFormatLengthRatio“

Siehe dazu auch Kapitel [Anpassung der dynamischen Daten bei Format-Variablen Maschinen](#)

Mit dem Bit “boStartMotion” wird die Bewegung gestartet. Eine fallende Flanke an „boStartMotion“ oder Drehzahlsollwert = 0 stoppt die Bewegung.

Der Drehzahlsollwert kann während des Betriebs fliegend verändert werden.

Das Vorzeichen von “r64SpeedSetpoint” definiert die Drehrichtung der Achse.

4.5.8 Mode 31 - SPEED_SPD_CTRL

Diese Betriebsart hat die gleiche Funktionalität wie Mode 30 „SPEED_POS_CTRL“, allerdings ist keine Lageregelung aktiv. Dadurch kann diese Betriebsart auch von Drehzahl-Achsen genutzt werden.

Die Nutzung der Schnittstelle ist identisch mit der der Betriebsart „SPEED_POS_CTRL“.

Diese Betriebsart kann u.U. auch für Positionier- oder Folgeachsen (Achsen mit Lageregelung) sinnvoll sein.

VORSICHT **Da der Lageregler in dieser Betriebsart deaktiviert ist, baut sich während des Verfahrens der Achse ein Lagefehler auf.**

Wird die Achse von dieser Betriebsart fliegend in eine Betriebsart mit Lageregelung geschaltet, kann ein Sollwertsprung entstehen.

Es wird daher empfohlen, die Achse nach „SPEED_SPD_CTRL“ Betrieb über die Betriebsart „E_STOP“ auszuschalten bevor in eine andere Betriebsart gewechselt wird!

4.5.9 Mode 40 - WINDING

Der Modus „WINDING“ koppelt eine Drehzahl- oder Positionierachse an ein TO Formelobjekt. Über dieses können Bewegungsvektoren sowie verschiedene Faktoren nach einer definierbaren Formel zu einem resultierenden Bewegungsvektor als Leitwert für die Achse verrechnet werden. Im Print Standard wird so aus dem Geschwindigkeitsvektor der Masterachse, einer additiven Geschwindigkeit für die Zugregelung sowie dem aktuellen Rollendurchmesser eine Sollgeschwindigkeit für eine Wickelachse berechnet.

Dazu werden innerhalb der Betriebsart ein Formelobjekt sowie eine fest hinterlegte Formel aktiviert. Des Weiteren wird das MotionIn Interface der Achse freigeschaltet.

Abhängig von den Variablen „boWinderEnable“ und „boWinderCtrlEnable“ in sAxisData.sWinderControl die mit den entsprechenden Wickler Interfaceparametern verschaltet werden müssen wird das MotionIn mit unterschiedlichen Dynamiken aktiviert:

Tabelle 4-17 Dynamik-Auswahl abhängig vom Zustand des Wicklers

boWinderEnable	boWinderCtrlEnable	Dynamik	Verwendung
FALSE	-	sLocalDyn	Stopp Befehl
TRUE	FALSE	sClutchDyn	Aufsynchronisieren auf Master Geschwindigkeit
TRUE	TRUE	sWinderOperationDyn	Wirken begrenzend während dem Wickelbetrieb

Wickler und Drehmoment-Funktionen

Für Wickler die in einer drehmomentgeregelten Betriebsart betrieben werden sollen oder mit Drehmomentvorsteuerung arbeiten muss folgendes beachtet werden:

Wird die Druck Standard STDclO Schnittstelle von einer externen PLC über den Kommunikationsbaustein FBLPrint_AxisCommunication angesteuert sollten die Drehmomentwerte vom Wicklerbaustein (torqueLimits und additiveTorque) auf die Drehmomentschnittstelle in sAxisData.sTorqueValues in der Einheit [Nm] verschaltet werden. Dazu muss der Parameter sAxisConfigData.eTorqueValueSource auf AXISDATA_INTERFACE gesetzt werden.

ACHTUNG	Prinzipiell müssen die Druck Standard Drehmomentfunktionen aktiviert werden um die Drehmomentschnittstelle zur Achse nutzen zu können. Siehe dazu Kapitel 4.12 Drehmoment-Funktionen .
----------------	--

Fliegender Rollenwechsel (Splice)

Die Betriebsart WINDING kann neben der eigentlichen Wickelfunktion auch in Verbindung mit einem fliegenden Rollenwechsel (Splice) verwendet werden. Über die Struktur sAxisData.sWinderControl werden die notwendigen Informationen von der jeweiligen Wickler-Instanz an den Druck Standard übergeben.

Der Parameter „boSpliceDone“ signalisiert dem Druck Standard den ausgeführten Splice (altes Material abgeschnitten, neues Material angeklebt).

Abhängig vom Parameter „eAfterSpliceMode“ wird der (alte) Wickel durch einen Stopp Befehl zum Stehen gebracht (STOP_AFTER_SPLICE) oder nach dem abbremesen in Entgegengesetzte Richtung beschleunigt um abgewickeltes Material wieder aufzuwickeln (REWIND_AFTER_SPLICE). Die in STDclO.r64SpeedSetpoint definierte Geschwindigkeit wird mit den Dynamiken aus sLocalDyn ausgeführt und kann mit „boStartMotion“ -> FALSE oder einen Modewechsel nach E-STOP beendet werden.

Hinweis	Genauere Informationen zu der Realisierung eines Wicklers mit dem Print Standard können der AddOn Applikation „Print Standard AddOn Reel Stand“ sowie dem Kapitel 4.13.4 Integration des Wicklers dieser Dokumentation entnommen werden.
----------------	--

4.5.10 Mode 50 – CAM_MODE

Dieser Modus bietet die Möglichkeit eine Achse mit Kurvenscheibenverhalten einem Leitwert folgen zu lassen. Es sind eine oder mehrere Kurvenscheiben möglich.

Der „CAM_MODE“ kann mit realen oder virtuellen Achsen ausgeführt werden. Die aktive Kurvenscheibe (und der aktive Master) kann über „STDclO.IN.u8CamNumber“ ausgewählt werden. Die möglichen Kurvenscheiben

(und Master) werden in Arrays konfiguriert („AxisTOConfigData.sUser.atoPossibleMasters“, „AxisTOConfigData.sUser.atoPossibleCams“). Zur Laufzeit des Systems ist es dann möglich, zwischen den konfigurierten Kurvenscheiben (Mastern) zu wechseln.

Hinweis

Auch an den FolgeObjekt TOs müssen die möglichen Kurvenscheiben (Master) angegeben werden.

4.5.11 Mode 60 – GEAR_MODE

Diese Betriebsart koppelt eine Achse mit wählbarem elektronischem Getriebe auf einen Master. Sie kann dabei dem Master Sollwert oder Istwert folgen. Master kann eine andere Achse (real oder virtuell) oder ein anderes Technology Objekt sein, wie z.B. externer Geber oder Addierobjekt. Die Achse kann auf den Master absolut oder relativ aufsynchronisieren. Das elektronische Sollwert-Getriebe zwischen Achse und ihrem Master wird über die STDcIO Schnittstelle vorgegeben („i32GearNumSetpoint“ / „i32GearDenSetpoint“).

Bei negativem Getriebeverhältnis, also „i32GearNumSetpoint“ / „i32GearDenSetpoint“ < 0, dreht die Gleichlaufachse negativ. Dies kann genutzt werden, wenn Gleichlaufachsen optional in beide Richtungen drehen sollen.

Der Leitwert (Master) ist frei wählbar und kann auch während dem Betrieb gewechselt werden.

Der Print Standard beinhaltet einen „default Master“ und eine Liste möglicher Master („possible masters array“). Dieses Array ist in „AxisTOConfigData.sUser.atoPossibleMasters“ definiert. Die möglichen Master können Achsen, Externe Geber, Addier- oder Formelobjekte sein.

Der aktive Master wird in der STDc Schnittstelle über den Parameter „u8MasterNumber“ aus der „possible Masters“-Liste ausgewählt. Bei „u8MasterNumber“ = 0, wird der default Master verwendet.

Wenn die SPS im Betrieb einen neuen Master vorgibt und dieser gültig ist beginnt die Achse sofort auf den neuen Master aufzusynchronisieren. Falls die angewählte Master Nummer im Array ungültig/falsch ist wird ein Applikationsfehler erzeugt.

Mit „boStartMotion“ synchronisiert die Achse zunächst relativ zum Master auf d.h. erreicht die gleiche Geschwindigkeit. Dabei entsteht ein zufälliger Registerwert/Phasenlage zum Master. In der Schnittstelle wird das Erreichen der Master Geschwindigkeit mit der Variable „STDcIO.OUT.boSpeedSynchronous“ angezeigt.

Hinweis

Zum Auf- und Absynchronisieren (Ein- /Auskuppeln) werden die Dynamiken aus „AxisConfigData.sClutchDyn“ verwendet.

Für die meisten Anwendungen, z.B. Druckzylinder, muss die Achse absoluten Lagebezug zum Master haben mit einstellbarem Register/Versatz. Die absolute Synchronisierung wird über das Bit „boSynchronizeAngular“ in der Schnittstelle

angestoßen. Damit berechnet die Achse eine überlagerte Korrekturbewegung (superimposed Motion) um den gewünschten Versatz zwischen Master und Slave Achse anzufahren. Der gewünschte Versatz wird mit den folgenden Variablen eingestellt:

- `STDclO.In.boActivateAbsRegister` = TRUE
- `“STDclO.IN.r64RegisterSetpoint”` = x [°]

Hinweis Wird über das Bit „boSynchronizeAngular“ Winkelgleichlauf hergestellt während „STDclO.In.boActivateAbsRegister“ = FALSE, wird der Registerversatz der Retain-Variablen „AxisDataRetain.r64ActiveRegisterSetpoint“ verwendet!

Die Synchronisierichtung kann über die Konfigurationsvariable „AxisConfigData.sSynchConfig.eSyncMode“ vorgegeben werden:

- FORWARD
- FORWARD_WITH_WINDOW
- BACKWARD
- BACKWARD_WITH_WINDOW
- SHORTEST_WAY

Für die Synchronisation mit Fenster kann die Fenstergröße über die Variable „AxisConfigData.sSynchConfig.r64SyncWindow“ eingestellt werden.

Befindet sich die Achse innerhalb des Fensters, synchronisiert sie auf kürzestem Weg auf, andernfalls in entsprechender Richtung.

Hinweis Wird ein negatives Getriebeverhältnis eingestellt, wird davon ausgegangen, dass der Zylinder in Bahnaufrichtung dreht. Deshalb verhält sich der eSyncMode entsprechend umgekehrt.
FORWARD = Bahnaufrichtung

Hinweis Wird der absolute Winkelgleichlauf angestoßen „boSynchronizeAngular“, werden die Dynamiken aus „AxisConfigData.sSyncDyn“ für die überlagerte Bewegung verwendet.

Das elektronische Getriebe kann während des Betriebs über die Schnittstelle geändert werden. Die Änderung kann entweder über die Schnittstellenparameter „i32GearNumSetpoint“ / „i32GearDenSetpoint“ oder über den Parameter „r64DrawSetpoint“ erfolgen. Mit „r64DrawSetpoint“ wird das Getriebe prozentual beeinflusst. Nach der Änderung wird das neue Getriebe sofort aktiv.

Hinweis Eine Getriebeanpassung wird mit den Dynamiken aus „AxisConfigData.sDrawDyn“ aktiv.

Geht die absolute Synchronisation verloren (STDcIO.OUT.boAngularSynchronous -> FALSE), z.B. durch eine Getriebeänderung, ist das erneute Aufsynchronisieren abhängig von der gewählten eSyncStartCondition (sAxisConfigData.sSyncConfig).

EDGE_TRIGGERED: Erneute steigende Flanke an
STDcIO.IN.boSynchronizeAngular notwendig.

STATE_TRIGGERED: Synchronisation wird automatisch erneute gestartet wenn STDcIO.IN.boSynchronizeAngular = TRUE und STDcIO.OUT.boAngularSynchronous = FALSE.

4.5.12 Mode 61 – SPEED_GEAR_MODE

Dieser Modus realisiert einen einfachen Geschwindigkeitsgleichlauf ohne Lagebezug. Er kann als vereinfachter „GEAR_MODE“ angesehen werden, für Achsen die lediglich der Sollgeschwindigkeit des Masters folgen sollen. Master können, wie im „GEAR_MODE“, Achsen oder andere Technologie Objekte (ext. Geber, Addierobjekte, etc.) sein, auf dessen Soll- oder Istwert gekoppelt wird.

Hinweis Die Eigenschaften bzgl. der Master-Verschaltungen, sowie die verwendeten Dynamiken entsprechen dem „GEAR_MODE“.

4.5.13 Mode 100 - ERROR

Dieser Modus kann nicht von der SPS angesteuert werden!

Der Modus wird vom SIMOTION Print Standard genutzt um Fehler in der Konfiguration oder der Schnittstelle des Print Standard anzuzeigen. Für jede Achse kann definiert werden wie Sie auf die Fehler reagieren soll. Über den Konfigurationsparameter „AxisConfigData.eErrorReaction“ stehen

- “COAST_STOP” und
- “E_STOP”

zur Verfügung.

Hinweis Virtuelle Achsen dürfen als Fehlerreaktion nur “E_Stop” nutzen.

Der Fehler Modus wird in der Schnittstelle angezeigt. Zum Quittieren muss die jeweilige Achse in die Betriebsart „E_STOP“ gebracht werden. Nachdem alle Fehler quittiert wurden, kann die Achse wieder gestartet werden.

4.5.14 Mode 101 - ERROR_TO_E_STOP

Dieser Modus kann nicht von der SPS angesteuert werden!

Er wird intern als „Zwischen-Mode“ verwendet, um vom Fehler-Fall (ERROR Mode) je nach AxisConfigData.eErrorReaction in den Mode „E_STOP“ zu gelangen.

4.5.15 Mode 102 - ERROR_TO_COAST_STOP

Dieser Modus kann nicht von der SPS angesteuert werden!

Er wird intern als „Zwischen-Mode“ verwendet, um vom Fehler-Fall (ERROR Mode) je nach AxisConfigData.eErrorReaction in den Mode „COAST_STOP“ zu gelangen.

4.6 Schnittstellen zur Maschinenlogik (SPS) und Antrieb

4.6.1 Schnittstelle zu Maschinenlogik

Die Schnittstelle zur Nutzung des Print Standard innerhalb SIMOTION ist die STDc Variablenstruktur.

Die dem Print Standard „überlagerte“ Maschinenlogik kann sowohl SIMOTION intern (internal PLC), als auch extern in einer SPS (external PLC) realisiert sein.

Der Print Standard ist vorbereitet um über den Funktionsbaustein „FBLPrint_AxisCommunication“ im Programm „pComm“ von einer der 3 folgenden Quellen angesteuert zu werden. Die jeweilige Quelle wird über zwei Eingangsparameter des Funktionsbausteins ausgewählt.

Abbildung 4-3 Schnittstellen zu/von SIMOTION Print Standard Schnittstelle STDc

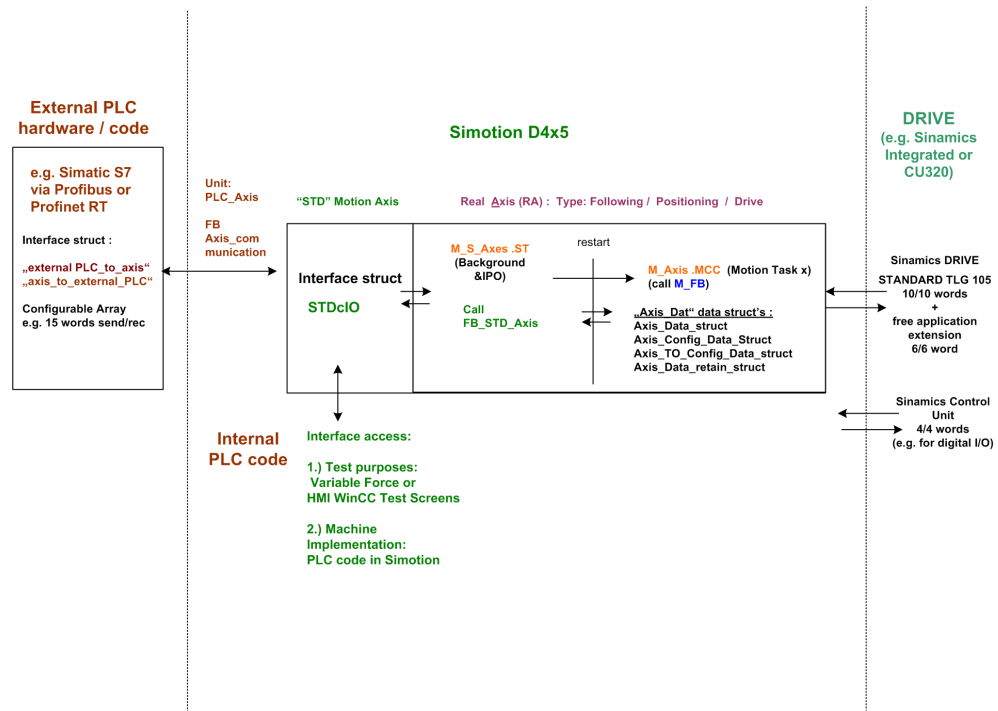


Tabelle 4-18 Ansteuerung der Print Standard STDc Schnittstelle

STDc Zugriff/Steuerung von	Variable „activateTestMode“	Variable „plcMode“
externe SPS z.B. S7	FALSE	EXTERNAL_PLC
SPS Programm in SIMOTION	FALSE	INTERNAL_PLC
IT Diag (HMI)	TRUE	<nicht relevant>

EXTERNAL_PLC

Falls die Ansteuerung des Print Standard von außerhalb der SIMOTION kommt, werden die Daten (z.B. bei PROFIBUS oder PROFINET RT/I-Device) im SIMOTION I/O Adressbereich empfangen und können von dort zur STDc

Schnittstelle übertragen werden. Dazu wird der Kommunikationsbaustein im Mode „EXTERNAL_PLC“ verwendet.

Die Eingangsvariable „activateTestMode“ des Kommunikationsbausteins muss mit FALSE belegt werden.

Der Eingang „plcMode“ muss mit „EXTERNAL_PLC“ beschrieben werden.

Die Bausteinparameter „axisStdclO“, „axisToExternalPLC“ und „extPLCToAxis“ müssen belegt werden. Die beiden letzten sind mit den SIMOTION Adresslisten-Variablen zu belegen.

Baustein-Intern werden die STDC Parameter auf die entsprechenden Bits der Adresslisten-Variablen und umgekehrt kopiert.

Der Aufbau der Schnittstelle ist unten ersichtlich.

INTERNAL_PLC

Ist die dem Print Standard überlagerte Maschinenlogik innerhalb der SIMOTION integriert, kann prinzipiell auf den Kommunikationsbaustein verzichtet werden. Es besteht die Möglichkeit direkt auf die Print Standard STDC Schnittstelle zuzugreifen.

Um jedoch die Diagnose-Funktionalität der IT Diag Seite nutzen zu können, wird empfohlen den Kommunikationsbaustein im Mode „INTERNAL_PLC“ zu verwenden.

Die Eingangsvariable „activateTestMode“ des Kommunikationsbausteins muss mit FALSE belegt werden.

Der Eingang „plcMode“ muss mit „INTERNAL_PLC“ beschrieben werden.

Die Bausteinparameter „axisTOInternalPLC“, „intPLCToAxis“, „AxisToHMI“ und „axisStdclO“ müssen belegt werden.

Baustein-Intern ist folgende Verschaltung aktiv:

```
axisTOInternalPLC      := axisStdclO.OUT;  
axisStdclO.IN          := intPLCTOAxis;  
axisTOHMI              := axisStdclO.OUT;
```

IT Diag (HMI)

Dieser Mode kann verwendet werden, um den Druck Standard über die IT Diag Seite bzw. ein HMI zu bedienen. Der Mode ist als Vorfür-, Test- bzw. Inbetriebnahme-Mode gedacht.

Die Eingangsvariable „activateTestMode“ des Kommunikationsbausteins muss mit TRUE belegt werden.

Der Eingang „plcMode“ ist nicht relevant.

Die Bausteinparameter „HMIToAxis“, „AxisToHMI“ und „axisStdclO“ müssen belegt werden.

Baustein-Intern ist folgende Verschaltung aktiv:

```
axisTOHMI              := axisStdclO.OUT;  
axisStdclO.IN          := HMITOAxis;
```

In der Druck Standard Bibliothek sind zwei Kommunikationsbausteine vorbereitet:

- FBLPrint_AxisCommunicationWord
- FBLPrint_AxisCommunicationByteV30

Der Funktionsbaustein „FBLPrint_AxisCommunicationWord“ kommuniziert Wortweise und ist unverändert zur Print Standard Version V2.2.1.0. Hier sind im Beispielprojekt nicht alle STDcIO Schnittstellen-Variablen eingebunden.

Der Der Funktionsbaustein „FBLPrint_AxisCommunicationByteV30“ kommuniziert in Bytes und wurde mit Version V3.0.0 überarbeitet und auf die neue STDcIO Schnittstelle angepasst.

In den folgenden Tabellen ist die Schnittstellenbelegung für beide Funktionsbausteine aufgeführt:

AxisCommunicationByteV30

Tabelle 4-19 Communication Byte Input (PLC to SIMOTION)

Byte/Bit	Data Type	Parameter
0.0	BOOL	boOff1EStop
0.1	BOOL	boEnableSpeedSetpoint
0.2	BOOL	boSynchronizeAngular
0.3	BOOL	boRelativePositioning
0.4	BOOL	boReferencingOnTheFly
0.5	BOOL	boActivateABSRegister
0.6	BOOL	boModeABSRegister
0.7	BOOL	boActivateRelRegister
1.0	BOOL	boNegativeRelRegister
1.1	BOOL	boInchRegisterAdvance
1.2	BOOL	boInchRegisterRetard
1.3	BOOL	boAcknowledgeFault
1.4	BOOL	boStartMotion
1.5	BOOL	boEnableTorqueLimiting
1.6	BOOL	boEnableReverseMotionInhibit
1.7	BOOL	boEnablePositiveMotionInhibit
2.0	BOOL	boMaskMotorBlockedFault
2.1	BOOL	boEnableAdditiveTorque
2.2	BOOL	boOpenBreak
2.3	BOOL	boSetpointMode
2.4	BOOL	boMopFaster
2.5	BOOL	boMopSlower
2.6	BOOL	boMopHold
4.0	REAL	r64SpeedSetpoint
8.0	REAL	r64PositioningSetpoint
12.0	REAL	r64RegisterSetpoint
16.0	REAL	r64ActualCylinderFormatLength

4 Funktionsbeschreibung

Byte/Bit	Data Type	Parameter
20.0	REAL	r64TorqueLimitPositive
24.0	REAL	r64TorqueLimitNegative
28.0	REAL	r64AdditiveTorque
32.0	DINT	i32GearNumSetpoint
36.0	DINT	i32GearDenSetpoint
40.0	INT	r64DrawSetpoint (16384 \pm 100%)
42.0	UINT	u16ModeNumber
44.0	USINT	u8MasterNumber
45.0	USINT	u8CamNumber

Tabelle 4-20 Communication Byte Output (SIMOTION to PLC)

Byte/Bit	Data Type	Parameter
0.0	BOOL	boReadyToPowerUp
0.1	BOOL	boSwitchOnInhibit
0.2	BOOL	boExternalSwitchOnInhibit
0.3	BOOL	boOperation
0.4	BOOL	boZeroSpeed
0.5	BOOL	boSpeedReached
0.6	BOOL	boSpeedSynchronous
0.7	BOOL	boAngularSynchronous
1.0	BOOL	boRegistrationActive
1.1	BOOL	boRegistrationDone
1.2	BOOL	boPositioningActive
1.3	BOOL	boPositioningDone
1.4	BOOL	boReferencingActive
1.5	BOOL	boReferencingDone
1.6	BOOL	boVMPosIsSet
1.7	BOOL	boEStopActive
2.0	BOOL	boEStopDone
2.1	BOOL	boBreakOpen
2.2	BOOL	boSTOActive
2.3	BOOL	boWarning
2.4	BOOL	boFault
2.5	BOOL	boApplicationWarning
2.6	BOOL	boApplicationError
2.7	BOOL	boTorqueLimitingActive
3.0	BOOL	boPositiveMotionInhibitActive
3.1	BOOL	boNegativeMotionInhibitActive
3.2	BOOL	boPositiveLimitReached

Byte/Bit	Data Type	Parameter
3.3	BOOL	boNegativeLimitReached
4.0	REAL	r64ActualSpeed
8.0	REAL	r64ActualPosition
12.0	REAL	r64ActualRegister
16.0	REAL	r64FollowingError
20.0	REAL	r64ActualTorque
24.0	REAL	r64ActivePositiveLimit
28.0	REAL	r64ActiveNegativeLimit
32.0	DINT	i32ActualGearNum
36.0	DINT	i32ActualGearDen
40.0	UINT	u16ActualModeNumber
42.0	UINT	u16ApplicationWarningNumber
44.0	UINT	u16ApplicationErrorNumber
46.0	USINT	u8ActualMasterNumber
47.0	USINT	u8ActualCamNumber

AxisCommunicationWord

Tabelle 4-21 Communication Word Input (PLC to SIMOTION)

Word/Bit	Data Type	Parameter
0.0	BOOL	boOff1EStop
0.1	BOOL	boEnableSpeedSetpoint
0.2	BOOL	boAcknowledgeFault
0.3	BOOL	boSetpointMode
0.4	BOOL	boMopFaster
0.5	BOOL	boMopSlower
0.6	BOOL	boMopHold
0.7	BOOL	boRelativePositioning
0.8	BOOL	boSynchronizeAngular
0.9	BOOL	boModeABSRegister
0.10	BOOL	boActivateABSRegister
0.11	BOOL	boInchRegisterAdvance
0.12	BOOL	boInchRegisterRetard
0.13	BOOL	boActivateRelRegister
0.14	BOOL	boNegativeRelRegister
1.0	BOOL	boEnableTorqueLimiting
1.1	BOOL	boEnableReverseMotionInhibit
1.2	BOOL	boEnablePositiveMotionInhibit
1.3	BOOL	boMaskMotorBlockedFault
1.4	BOOL	boEnableAdditiveTorque
1.5	BOOL	boReferencingOnTheFly
1.6	BOOL	boStartMotion

4 Funktionsbeschreibung

Word/Bit	Data Type	Parameter
2.0	REAL	r64SpeedSetpoint
4.0	REAL	r64PositioningSetpoint
6.0	REAL	r64RegisterSetpoint
8.0	INT	r64DrawSetpoint (16384 \pm 100%)
9.0	DINT	u16ModeNumber
10.0	USINT	u8MasterNumber
11.0	USINT	u8CamNumber
12.0	REAL	r64ActualCylinderFormatLength
14.0	REAL	r64TorqueLimitPositive
16.0	REAL	r64TorqueLimitNegative
18.0	REAL	r64AdditiveTorque

Tabelle 4-22 Communication Word Output (SIMOTION to PLC)

Word/Bit	Data Type	Parameter
0.0	BOOL	boReadyToPowerUp
0.1	BOOL	boOperation
0.2	BOOL	boFault
0.3	BOOL	boSwitchOnInhibi
0.4	BOOL	boWarning
0.5	BOOL	boZeroSpeed
0.6	BOOL	boSTOActive
0.7	BOOL	boSpeedReached
0.8	BOOL	boSpeedSynchronous
0.9	BOOL	boAngularSynchronous
0.10	BOOL	boPositioningActive
0.11	BOOL	boPositioningDone
0.12	BOOL	boReferencingActive
0.13	BOOL	boReferencingDone
0.14	BOOL	boRegistrationActive
1.0	BOOL	boRegistrationDone
1.1	BOOL	boVMPosIsSet
1.2	BOOL	boApplicationError
1.3	BOOL	boApplicationWarning
1.4	BOOL	boTorqueLimitingActive
1.5	BOOL	boPositiveMotionInhibitActive
1.6	BOOL	boNegativeMotionInhibitActive
1.7	BOOL	boPositiveLimitReached
1.8	BOOL	boNegativeLimitReached
2.0	REAL	r64ActualSpeed

4 Funktionsbeschreibung

Word/Bit	Data Type	Parameter
4.0	REAL	r64ActualPosition
6.0	REAL	r64ActualTorque
8.0	REAL	r64ActualRegister
10.0	DINT	u16ActualModeNumber
11.0	USINT	u8ActualMasterNumber
12.0	USINT	u8ActualCamNumber
13.0	REAL	r64ActivePositiveLimit
15.0	REAL	r64ActiveNegativeLimit

4.6.2 Schnittstelle zum Antrieb (z.B. SINAMICS)

Die Achsen in SIMOTION kommunizieren mit dem (integriertem oder externen) Antrieb (SINAMICS) entweder über die integrierte PROFIBUS Kommunikation bei „SINAMICS Integrated“ der SIMOTION, oder wenn externe Antriebe (SINAMICS) genutzt werden (z.B. CU320), über den externen takt synchronen PROFIBUS oder PROFINET.

In beiden Fällen kann ein Standardtelegramm genutzt werden (z.B. Standard Telegramm 105 mit DSC) um die generelle Ansteuerung des Antriebs von SIMOTION bereitzustellen. Ist die Symbolische Zuordnung im Projekt aktiviert, wird automatisch ein passendes Telegramm eingestellt.

Werden die Drehmoment-technologischen Erweiterungen benötigt, wird beim aktivieren des Technologiedatenblocks das Telegramm automatisch um 1 Wort Input und 3 Worte Output erweitert.

Folgende Antriebsparameter werden dann mit dem TO verdrahtet:

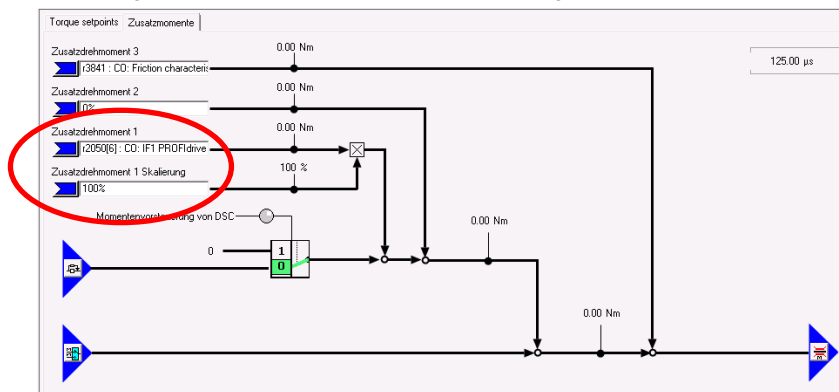
Tabelle 4-23 Verschaltungen der Telegrammerweiterung

Parameter	Beschreibung
p1522	Momentengrenze oben Dieser Parameter wird mit dem Wert für die obere Momentengrenze aus dem Technologiedatenblock verschaltet.
p1523	Momentengrenze unten Dieser Parameter wird mit dem Wert für die untere Momentengrenze aus dem Technologiedatenblock verschaltet.
p1511	Zusatzdrehmoment 1 Dieser Parameter wird mit dem Zusatzmoment aus dem Technologiedatenblock verschaltet. Mit dieser Verknüpfung wird das Vorsteuermoment übertragen.
r80	Drehmomentenistwert Aktueller Drehmomentenistwert, wird zur Steuerung gesendet.

Copyright © Siemens AG 2016 All rights reserved

ACHTUNG Der Antriebsparameter p1512 (Skalierungsfaktor Zusatzdrehmoment 1) muss per Hand auf 100% gestellt werden, damit das Zusatzdrehmoment im Antrieb wirksam wird!

Abbildung 4-4 Zusatzdrehmoment und Skalierungsfaktor



4.7 „Force“-axis Funktionalität

Eine virtuelle Gleichlaufachse (z.B. Local Master) oder eine reale TM41 Achse kann per Default auf einen Master „geforced“ werden. Diese Achse stellt dann eine identische Kopie zu dem entsprechenden Master dar.

Das „forcen“ wird über die Eingangsvariable „forceAxis“ = TRUE des „FBLPrint_StdAxisBckgrnd“ aktiviert.

Hinweis Die zu „forcende“ Achse muss eine virtuelle Gleichlaufachse oder eine reale TM41 Achse sein (axisTOConfigData.sInternal.eAxisType = ENCODER_OUTPUT).

Hinweis Der Konfigurationsparameter „AxisTOConfData.sUser.toDefaultMaster“ muss verschaltet sein!

Folgende STDcIO Schnittstellen-Bits werden durch das „forcen“ automatisch intern geschrieben:

- u16ModeNumber = GEAR_MODE
- boOff1EStop = TRUE
- boSynchronizeAngular = TRUE
- boStartMotion = TRUE
- bolnchRegisterAdvance = FALSE
- bolnchRegisterRetard = FALSE
- r64DrawSetpoint = 0.0
- r64RegisterSetpoint = 0.0
- AxisData.toCurrentMaster = toDefaultMasterInternal

ACHTUNG Es muss darauf geachtet werden, dass die STDcIO.IN Schnittstelle von geforced Achsen nicht zusätzlich von außen beschrieben wird.

Es wird empfohlen den FBAxisCommunication Eingang plcMode auf NO_MODE_SELECTED zu setzen.

ACHTUNG	<p>Wenn der Gleichlauf an einer Folgeachse gestartet werden soll muss applikativ sichergestellt werden, dass der verteilte Gleichlauf zwischen den Controllern dafür bereit ist.</p> <p>Es wird empfohlen, die Slave Achse erst zu „forcen“ wenn die Systemvariable „distributedmotion.stateofdelayvalue“ = valid ist.</p> <p>Dies gilt bei masterseitiger Verzögerung der Ausgangswerte („VA_GM_Web“ Konfigurationsdatum TypeOfAxis.distributedmotion.enableDelayOfCommandValueOutput = YES)</p>
----------------	--

4.7.1 Automatisches Fehler quittieren

Über den Konfigurationsparameter AxisConfigData.eFaultAckForcedAxis = YES wird eine geforcete Achse bei einem Fehler intern automatisch quittiert und wieder auf ihren Master aufsynchronisiert.

Bei einer geforceten TM41 Achse kann über den Konfigurationsparameter AxisConfigData.tFaultAckForcedEncOutputTime das Zeitintervall in dem versucht wird anstehende Fehler automatisch zu quittieren parametrieren werden.

4.8 MOP - Motorisiertes Potentiometer

Das „Motorisierte Potentiometer“ kann genutzt werden um den Drehzahlsollwert in den Betriebsarten „SPEED_POS_CTRL“ und „SPEED_SPD_CTRL“ zu verstellen.

Die maximale und minimale Motorpotentiometer Geschwindigkeit kann in „AxisConfigData.sMop.r64Max“ und „AxisConfigData.sMop.r64Min“ konfiguriert werden.

Die Art der Vorgabe sowie die Einheit dieser Parameter ist abhängig von dem Konfigurationsparameter „AxisConfigData.eVelocityFormat“ (DIRECT / NORMALIZED_WITH_FORMATADAPTION / NORMALIZED_WITHOUT_FORMATADAPTION).

Über die STDc Variablen „boMOPFaster“ / „boMOPSlower“ kann die Achse abhängig von der Dauer des „High“-Zustands an den Eingängen beschleunigt bzw. verzögert werden.

Mit „boMOPHold“ bzw. „boMOPFaster“ und „boMOPSlower“ = FALSE wird die aktuellen Geschwindigkeit gehalten.

Folgende Signale müssen gesetzt sein, um das MOP anzusteuern:

- u16ModeNumber = 30 / 31
- boEnableSpeedSetpoint = TRUE
- boSetpointMode = TRUE

4.9 Spezifikation von Mastern und Kurvenscheiben

Im "CAM_MODE" und "GEAR_MODE" muss ein Master bzw. eine Kurvenscheibe als Leitwert für die Achse vorgegeben werden. Sowohl Master als auch die Kurvenscheibe kann von der SPS in der Schnittstelle vorgegeben werden. Ein „fliegender Wechsel“ ist ebenfalls möglich.

Im Print Standard kann eine Liste mit möglichen Mastern bzw. Kurvenscheiben in Form eines Arrays vorkonfiguriert werden. Sie sind definiert in „AxisTOConfigData.sUser.atoPossibleMasters“ bzw. „AxisTOConfigData.sUser.atoPossibleCams“.

Die Arrays können für jede Achse unterschiedlich vorbelegt werden.

Der aktive Master/Kurvenscheibe wird über die SchnittstellenvARIABLE „STDcIO.IN.u8MasterNumber“ bzw. „STDcIO.IN.u8CamNumber“ ausgewählt.

Die Schnittstelle zeigt im Ausgangsbereich den aktiven Master („u8ActualMasterNumber“) und die aktive Kurvenscheibe („u8ActualCamNumber“) an.

Falls nur ein Master/Kurvenscheibe ausreichend ist, können die SchnittstellenvARIABLEN „AxisTOConfigData.sUser.toDefaultMaster“/ „AxisTOConfigData.sUser.toDefaultCam“ genutzt werden. Die Konfiguration der Master/Kurvenscheiben Arrays ist damit nicht notwendig. Das Handling kann dadurch für einfache Anwendungen deutlich vereinfacht werden.

Von der SPS muss als „u8MasterNumber“ / „u8CamNumber“ der Wert Null geschickt werden.

Als mögliche Master stehen folgende SIMOTION Technologie Objekte zur Verfügung:

- Positionier-/ Folge-Achse (real oder virtuell)
- Externe Geber
- Addierobjekt oder Formelobjekt

ACHTUNG	<p>Mögliche Master/Kurvenscheiben werden wie oben beschrieben für die Nutzung im Print Standard konfiguriert.</p> <p>Zusätzlich sind die Master/Kurvenscheiben am Folgeobjekt (TO) zu konfigurieren, ansonsten erzeugt SIMOTION einen Systemfehler.</p>
----------------	---

4.10 Register Funktionalität

4.10.1 Bedienung

Die Register/Phasenlage wird mit einer überlagerten Positionierbewegung realisiert bzw. verstellt.

Es sind 3 verschiedene Arten der Register Verstellung implementiert:

- Register Tippen (vorwärts / rückwärts)
- Relative Register Verstellung
- Absolute Register Verstellung

Die Registerverstellung ist nur im Betriebsmodus "GEAR_MODE" verfügbar. Zusätzlich muss die Achse bereits Geschwindigkeits- und Winkelsynchron sein („boSpeedSynchronous“ = TRUE und „boAngularSynchronous“ = TRUE).

Register Tippen

Das Register Tippen wird über 2 Bits in der Schnittstelle angestoßen:

- boInchRegisterAdvance (positive Produktionsrichtung)
- boInchRegisterRetard (negative Produktionsrichtung)

Wenn eines der Bits TRUE ist und das andere FALSE dann bewegt sich das Register in die jeweilige Richtung. Wenn beide TRUE sind verändert sich das Register nicht.

Die Register Verstellgeschwindigkeit, sowie die Dynamischen Daten sind in „AxisConfigData.sInchRegisterDyn“ definiert.

Hinweis Die Einheiten der dynamischen Daten entsprechen den Einheiten am TO.

Hinweis Das Register Tippen wird nur ausgeführt wenn die Achse Geschwindigkeitssynchron ist („boSpeedSynchronous“ = TRUE) und weder eine absolute noch eine relative Registerverstellung aktiv ist.

VORSICHT	<p>Wenn die aktuelle Maschinengeschwindigkeit kleiner ist als die zugelassene Registerverstellgeschwindigkeit, kann es vorkommen, dass durch die Registerverstellung eine negative Drehrichtung entsteht.</p> <p>Dies ist zu beachten wenn Achsen nicht reversieren ($n < 0$) dürfen.</p>
-----------------	---

Relative Register Verstellung

Die relative Registerverstellung wird mit den folgenden Schnittstellen Variablen bedient:

- boActivateRelRegister = TRUE
- boNegativeRelRegister
- r64RegisterSetpoint [°]

Hinweis Es muss sichergestellt sein, dass die Achse bereits Geschwindigkeitssynchron ist („boSpeedSynchronous“ = TRUE) und keine absolute Registerverstellung aktiv ist.

Die Variable “r64RegisterSetpoint” definiert die gewünschte Verstellung z.B. in Grad. Die Verstellung wird relativ zur derzeitigen Position ausgeführt mithilfe des „superimposed coordinate system“ d.h. zusätzlich zur aktuellen Hauptbewegung der Maschine.

Die Register Verstellgeschwindigkeit, sowie die Dynamischen Daten sind in „AxisConfigData.sRelRegisterDyn“ definiert.

Hinweis Die Einheiten der dynamischen Daten entsprechen den Einheiten am TO.

Die Register Verstellbewegung wird durch das Bit “boActivateRelRegister“ gestartet. Während der Verstellung werden neue Register Sollwerte übernommen.

Über die Variable “boNegativeRelRegister” kann die Bewegungsrichtung gewählt werden.

Um das Register mehrfach um den gleichen Betrag zu verstellen ist das folgende Verfahren anzuwenden, damit das System eine Änderung des Sollwertes erkennt. Währenddessen bleibt „boActivateRelRegister“ = TRUE.

- r64RegisterSetpoint = 10° -> Register Verstellung um 10°
- r64RegisterSetpoint = 0° -> Register Verstellung um 0°
- r64RegisterSetpoint = 10° -> Register Verstellung um 10°

VORSICHT Wenn die aktuelle Maschinengeschwindigkeit kleiner ist als die zugelassene Registerverstellgeschwindigkeit, kann es vorkommen, dass durch die Registerverstellung eine negative Drehrichtung entsteht.
Dies ist zu beachten wenn Achsen nicht reversieren ($n < 0$) dürfen.

Absolute Register Verstellung

Die absolute Registerverstellung wird mit den folgenden Schnittstellen Variablen bedient:

- boActivateAbsRegister = TRUE
- boModeAbsRegister
- r64RegisterSetpoint [°]

Hinweis Es muss sichergestellt sein, dass die Achse bereits Winkelsynchron ist („boAngularSynchronous“ = TRUE) und keine relative Registerverstellung aktiv ist.

Die Variable “r64RegisterSetpoint” definiert die gewünschte Verstellung z.B. in Grad. Der Sollwert wird absolut angefahren in Bezug zur Position Null d.h. in Bezug auf Register Null wenn Master und Slave keinen Offset besitzen.

Die Register Verstellgeschwindigkeit, sowie die Dynamischen Daten sind in „AxisConfigData.sAbsRegisterDyn“ definiert.

Hinweis Die Einheiten der dynamischen Daten entsprechen den Einheiten am TO.

Die Register Verstellbewegung wird gestartet durch das Bit “boActivateAbsRegister“. Während der Verstellung werden neue Register Sollwerte übernommen.

Wenn “boModeAbsRegister” = TRUE ist nimmt die Register Korrektur den kürzesten Weg, ansonsten geht die Verstellung nur in Vorwärtsrichtung.

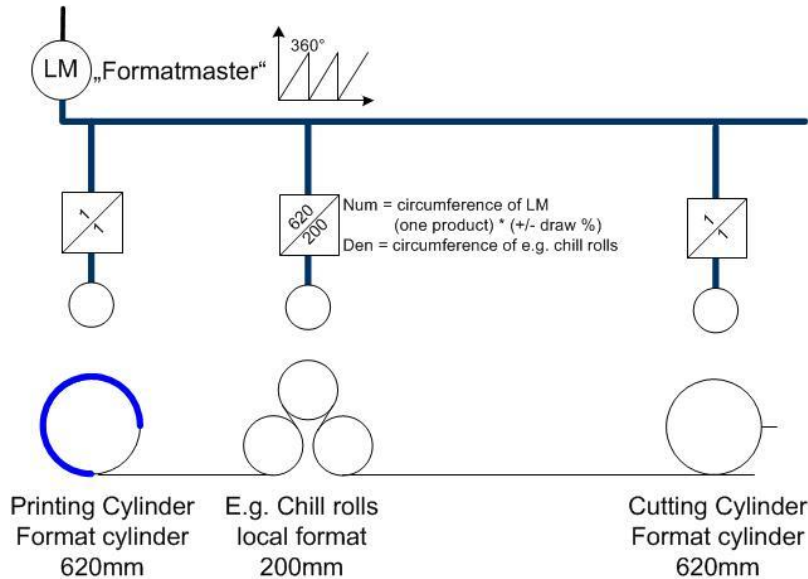
Hinweis Wird ein negatives Getriebeverhältnis eingestellt, wird davon ausgegangen, dass der Zylinder in Bahnaufrichtung dreht. Deshalb verhält sich der boModeAbsRegister umgekehrt, aber für den Anwender wie erwartet.
boModeAbsRegister = FALSE: Verstellung nur in Bahnaufrichtung!

4.10.2 Beispiele für Getriebefaktor ungleich 1

Bei verschiedenen Anwendungsfällen ist es gewünscht, einen Getriebefaktor ungleich 1 zwischen Master und Slave einzustellen.

Zum Beispiel kann bei Offsetmaschinen 1 Produkt oder 2 Produkte pro Druckzylinderumdrehung gedruckt werden.

Abbildung 4-5 Pro Zylinderumdrehung wird 1 Produkt gedruckt



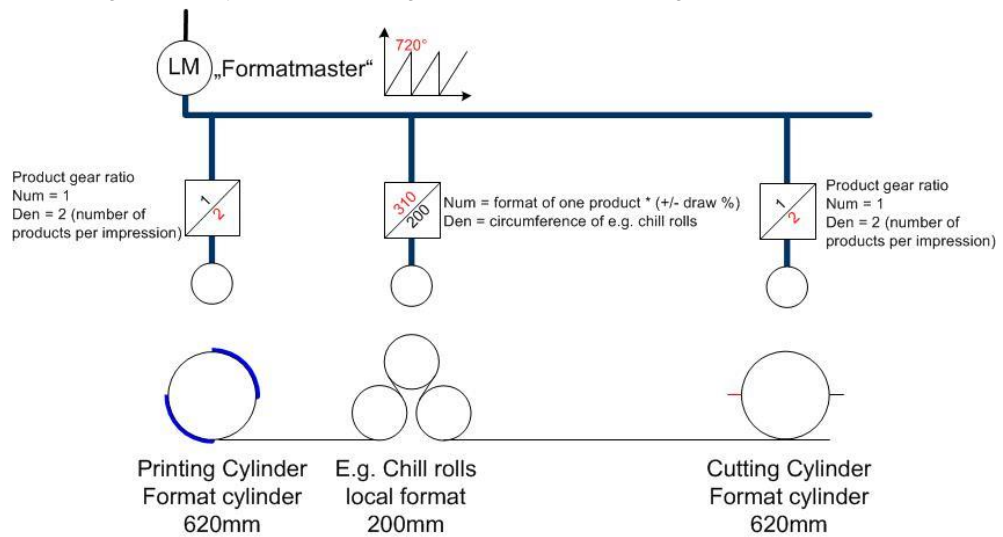
1 Produkt pro Druckzylinderumdrehung (620mm)
1 Produkt entspricht 360° der Leitachse

Möchte man bei 2 Produkten pro Druckzylinderumdrehung, dass pro 360° der Leitachse ein Produkt gedruckt wird, muss ein Getriebe von 1:2 eingestellt werden.

Natürlich muss der Sollwert dann doppelt so hoch vorgegeben werden, um die gleiche Druckgeschwindigkeit zu erreichen. Dadurch entspricht der Sollwert der Produktionsgeschwindigkeit und nicht der Motordrehzahl. (360° entspricht einem Produkt und nicht einer Motorumdrehung)

4 Funktionsbeschreibung

Abbildung 4-6 Pro Zylinderumdrehung wird 1 oder 2 Produkte gedruckt



2 Produkte pro Druckzylinderumdrehung (2 * 310mm)

1 Produkt entspricht 360° der Leitachse, wenn Getriebe 1:2

Der Sollwert der Leitachse ist bei 2 Nutzen doppelt so hoch, da Einheit [copies/h]

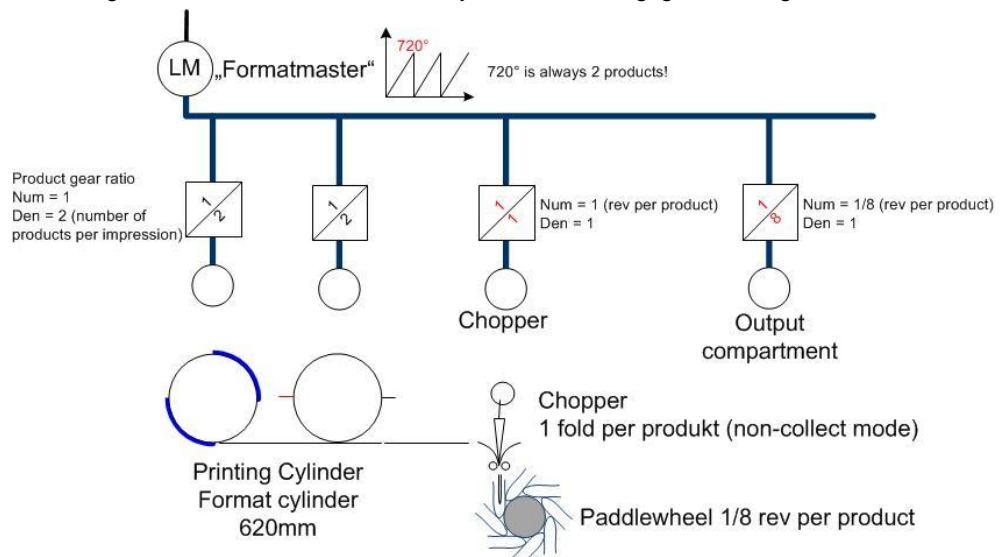
Erweitert man das Beispiel der Offsetmaschine um den Falzapparat, so gibt es hier Antriebe mit einer bestimmten Anzahl von Produkten pro Umdrehung

Diese Achsen haben keine synchrone Umfangsgeschwindigkeit.

Z.B.

- Schwertfalz = Chopper (1 Umdrehung pro Produkt)
- Schaufelrad = Paddlewheel (1/8 Umdrehung pro Produkt)

Abbildung 4-7 Getriebe für Achsen ohne synchrone Umfangsgeschwindigkeit



Abhängig von der Falzproduktion, z.B. Sammeln, kann der Getriebefaktor einfach angepasst werden.

Hinweis Um das Register korrekt verstellen zu können, muss der Parameter `r64SynchAxisCycle` angepasst werden!
Siehe dazu „[Einstellung von r64SynchAxisCycle bei Getriebefaktor ungleich 1](#)“

4.10.3 Einstellung von `r64SynchAxisCycle` bei Getriebefaktor ungleich 1

Der Registersollwert bezieht sich immer auf die Slaveachse.

Register 180°: Bei der Slaveposition = 180° ist die Masterposition = 0°.
Dies gilt auch bei einem Getriebefaktor $\neq 1$.

Hinweis Eine Registerverstellung ist nur sinnvoll und eindeutig bei folgenden Getriebeübersetzungen:
 $i32GearDenSetpoint / i32GearNumSetpoint = [1, 2, 3 \dots]$

Hinweis Der maximale eindeutige Bereich der Registerverstellung errechnet sich wie folgt:
 $r64SynchAxisCycle \leq \text{Modulo Master} * \text{Getriebeübersetzung}$
UND
 $r64SynchAxisCycle \leq \text{Modulo Slave}$

`r64SynchAxisCycle` entsprechend berechnen und einstellen!

Beispiel 1: Das Beispiel bezieht sich auf [Abbildung 4-6](#).

Modulo Master: 720°
Modulo Slave: 360°
Getriebefaktor: 1:2

Berechnung maximaler r64SynchAxisCycle:

$r64SynchAxisCycle \leq \text{Modulo Master} * \text{Getriebeübersetzung} = 720^\circ * 1:2 = 360^\circ$
UND

$r64SynchAxisCycle \leq \text{Modulo Slave} = 360^\circ$

Eine eindeutige Registerverstellung ist innerhalb 360° möglich.

Der Trace zeigt folgende Spuren:

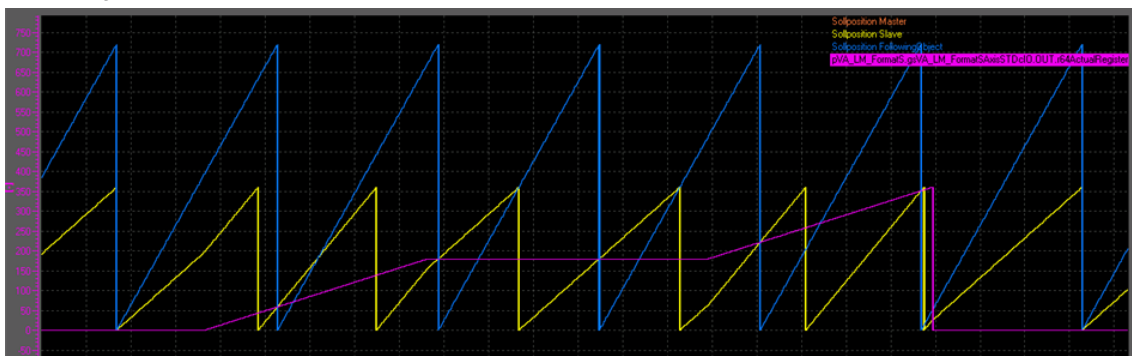
Sollposition Master= Sollposition FollowingObject = blau

Sollposition Slave = gelb

r64ActualRegister = pink

Alle Spuren sind gleich normiert.

Abbildung 4-8



Das Register kann innerhalb 360°, also im kompletten Modulbereich des Slaves, verstellt werden.

Pro Slavezyklus gibt es nur einen Schnittpunkt mit dem Master, wenn keine Registerbewegung aktiv ist!

Beispiel 2: Das Beispiel bezieht sich auf [Abbildung 4-6](#).

Unterschied zu Beispiel 1:

Es werden 4 Produkte pro Druckzylinderumdrehung gedruckt, deshalb wird das Getriebe auf 1:4 gestellt!

Modulo Master: 720°

Modulo Slave: 360°

Getriebefaktor: 1:4

Berechnung maximaler r64SynchAxisCycle:

$r64SynchAxisCycle \leq \text{Modulo Master} * \text{Getriebeübersetzung} = 720^\circ * 1:4 = 180^\circ$

UND

$r64SynchAxisCycle \leq \text{Modulo Slave} = 360^\circ$

Eine eindeutige Registerverstellung ist nur innerhalb **180°** möglich. Ist eine eindeutige Registerverstellung von 360° (Modulo Slave) notwendig, muss der Master Modulwert auf 1440° eingestellt werden.

Der Trace zeigt folgende Spuren:

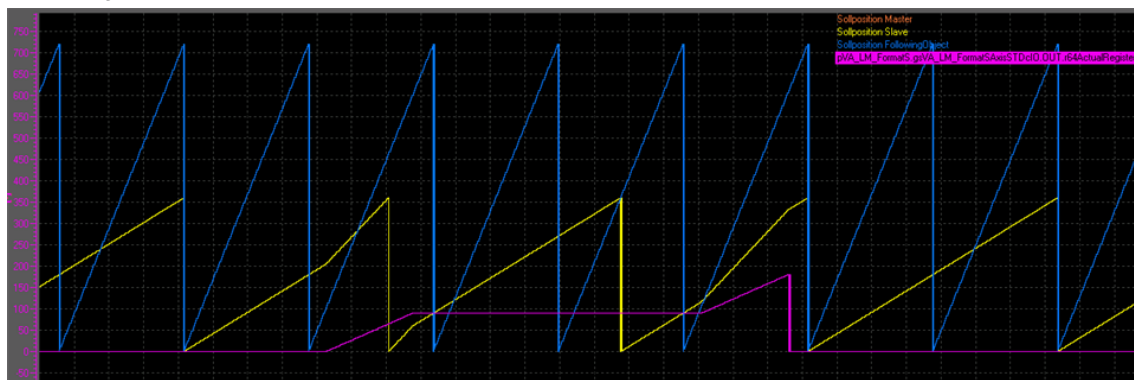
Sollposition Master= Sollposition FollowingObject = blau

Sollposition Slave = gelb

r64ActualRegister = pink

Alle Spuren sind gleich normiert.

Abbildung 4-9



Das Register kann nur innerhalb 180° eindeutig verstellt werden.

Pro Slave-Zyklus gibt es **zwei** Schnittpunkte mit dem Master, wenn keine Registerbewegung aktiv ist!

Beispiel 3: Schaufelrad (Falzapparat) auf [Abbildung 4-7](#).

Modulo Master: 720°

Modulo Slave: 45° (360° / 8 Schaufeln)

Getriebefaktor: 1:8

Berechnung maximaler r64SynchAxisCycle:

$r64SynchAxisCycle \leq \text{Modulo Master} * \text{Getriebeübersetzung} = 720^\circ * 1:8 = 90^\circ$

UND

$r64SynchAxisCycle \leq \text{Modulo Slave} = 45^\circ$

4 Funktionsbeschreibung

Eine eindeutige Registerverstellung ist nur innerhalb 45° möglich und auch gewünscht.

Der Trace zeigt folgende Spuren:

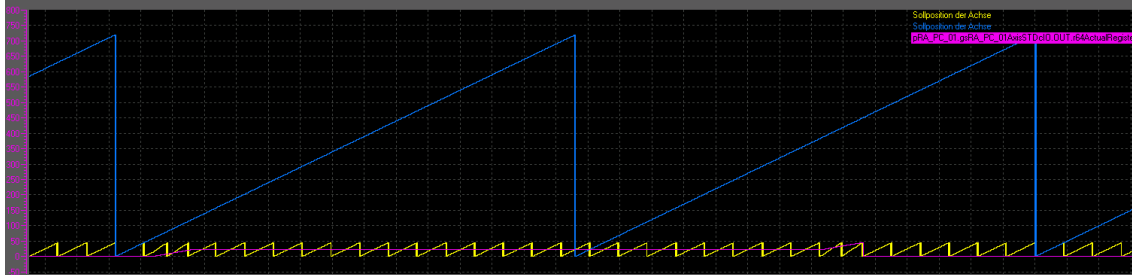
Sollposition Master= Sollposition FollowingObject = blau

Sollposition Slave = gelb

r64ActualRegister = pink

Alle Spuren sind gleich normiert.

Abbildung 4-10 Modulo Slave = 45°



Das Register kann innerhalb 45° eindeutig verstellt werden.

Alternativ kann das Schaufelrad auch mit Modulo 360° arbeiten:

Modulo Master: 720°

Modulo Slave: 360°

Getriebefaktor: 1:8

Berechnung maximaler r64SynchAxisCycle:

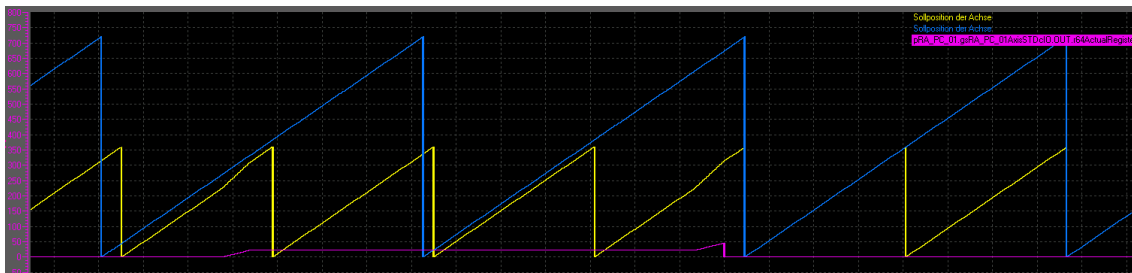
$r64SynchAxisCycle \leq \text{Modulo Master} * \text{Getriebeübersetzung} = 720^\circ * 1:8 = 90^\circ$

UND

$r64SynchAxisCycle \leq \text{Modulo Slave} = 360^\circ$

Eine eindeutige Registerverstellung ist innerhalb 90° möglich. Die gewünschte Registerverstellung ist jedoch 45° (360° / 8 Schaufeln), deshalb wird r64SynchAxisCycle nicht auf den Maximalwert von 90° sondern auf 45° eingestellt.

Abbildung 4-11 Modulo Slave = 360°



Das Register kann innerhalb 45° eindeutig verstellt werden. Übersichtlicher ist jedoch die Einstellung Modulo Slave = 360° wie in [Abbildung 4-10](#) gezeigt.

4.11 Spezifikation von dynamischen Daten

In der Print Standard Applikation können die dynamischen Daten (Beschleunigungen, Verzögerungen und Verrundungen (Ruck)) für die verschiedenen Betriebsarten der Achsen unabhängig eingestellt werden.

Die Datenstrukturen die die dynamischen Daten enthalten sind in der Struktur „AxisConfigData“ definiert und unabhängig vom Typ der Achse. Die Werte sind vorbelegt, können aber durch den Benutzer angepasst werden.

Folgende Dynamik-Strukturen sind vorhanden:

Tabelle 4-24 Dynamik-Strukturen

Struktur	Verwendung
sLocalDyn	für lokalen Betrieb, Mode 30/31
sPosDyn	Positionierdynamiken, Mode 20
sClutchDyn	Einkuppeln in Mode 60/61; Auskuppeln bei „boStartMotion“ = FALSE in Mode 60/61
sDrawDyn	Draw-Setpoint- bzw. Getriebefaktor- Änderung in Mode 60/61
sAbsRegisterDyn	für Registerverstellung absolut in Mode 60
sRelRegisterDyn	für Registerverstellung relativ in Mode 60
sInchRegisterDyn	für Registerverstellung tippen in Mode 60
sSyncDyn	für „boSynchronizeAngular“ in Mode 60
sEStopDyn	für Mode 1

In „dmGlobal“ wird für jede Achsgruppe eine Struktur-Variable der „AxisConfigData“ Struktur erzeugt.

Folgende Gruppen sind im Beispielprojekt angelegt:

- gsGMAxisConfigData (Global Master Achsen)
- gsLMAxisConfigData (Lokal Master Achsen)
- gsRAFAxisConfigData (reale Folge-Achsen)
- gsRAPAxisConfigData (reale Positionier-Achsen)
- gsRADAxisConfigData (reale Drehzahl-Achsen)

Diese Struktur-Variablen können dann einmalig parametrisiert werden und den entsprechenden Achsen zugeordnet werden.

Dies ist hilfreich, da in Maschinen technologisch unterschiedliche Achsen auch häufig dynamisch verschiedenes Verhalten benötigen. Mit diesem Konzept lassen sich verschiedene dynamische Konfigurationen für bestimmte Achsarten erzeugen.

4.11.1 Anpassung der dynamischen Daten bei Format-Variablen Maschinen

In einigen Druckmaschinen Segmenten ist eine Veränderung des Druckzylinderumfangs von einem Produktionslauf zum nächsten möglich. Diese sogenannten „Format-Variablen“ Maschinen kommen z.B. im Flexo- und Tiefdruckbereich vor.

Im Print Standard werden die dynamischen Daten abhängig von der aktuellen Zylinderformatlänge bezogen auf eine Referenzlänge automatisch adaptiert.

Dazu werden die folgenden Parameter benutzt:

- „STDclO.In.r64ActualCylinderFormatlength“ [mm]
- „AxisConfigData.sCylinderFormat.r64ReferenceLength“ [mm]
- „AxisConfigData.sCylinderFormat.r64MinLength“ [mm]
- „AxisConfigData.sCylinderFormat.r64MaxLength“ [mm]

ACHTUNG Alle Parameter müssen in der Einheit mm angegeben werden!

Die dynamischen Daten werden abhängig vom Verhältnis des aktuellen Zylinders zum Referenzzylinder angepasst.

In den Dynamik-Einstellungen aller Bewegungsbefehle im Funktionsbaustein „FBLPrint_MotionAxis“ wird der Dynamik-Konfigurationsparameterwert (z.B. „sLocalDyn.r64Accel“) dazu mit dem Faktor „AxisData.r64CylinderFormatlengthRatio“ verrechnet.

Der Geschwindigkeitssollwert wird bei $eVelocityFormat = NORMALIZED_WITH_FORMATADAPTION$ (ab V3.1) mit „AxisData.r64CylinderFormatlengthRatio“ multipliziert.

Dieser Faktor wird aus dem aktuellem Zylinderformat und dem Referenzzylinderformat gebildet:

$$\text{„AxisData.r64CylinderFormatlengthRatio“} := \frac{\text{„AxisConfigData.sCylinderFormat.r64ReferenceLength“}}{\text{„STDclO.IN.r64ActualCylinderFormatlength“}}$$

Der Parameter „STDclO.IN.r64ActualCylinderFormatlength“ wird zusätzlich durch die Grenzen „AxisConfigData.sCylinderFormat.r64MinLength“ und „AxisConfigData.sCylinderFormat.r64MaxLength“ limitiert.

Werden diese Grenzen verletzt, wird ein Applikationsfehler ausgegeben und der maximale bzw. minimale Wert des Zylinders verwendet.

Bei einer fehlerhaften Parametrierung der Grenzen wird ein Applikationsfehler ausgegeben und der Faktor „AxisData.r64CylinderFormatlengthRatio“ auf 1 gesetzt.

ACHTUNG Wird keine Formatanpassung benötigt, muss „STDclO.In.r64ActualCylinderFormatlength“ == „AxisConfigData.sCylinderFormat.r64ReferenceLength“ sein!
Beide Werte sind im Beispielprojekt im StartUp-Programm mit der Formatlänge 1000 mm vorbelegt.

4.12 Drehmoment-Funktionen

Die Print Standard STDC Schnittstelle beinhaltet die Funktionen/den Zugriff auf folgende Drehmoment Funktionen.

- Drehmoment Begrenzung
- Vorwärts- bzw. Rücklaufsperr
- Zusatzdrehmoment

Wichtige Anmerkungen:

Hinweis Zur Verwendung der Drehmoment-Funktionen muss der Technologiedatenblock am Achs-TO aktiviert sein. Die entsprechende Rückmeldung gibt das Bit „axisTOConfigData.sInternal.boTechnologyData“.

Außerdem muss der Skalierungsfaktor für das Zusatzdrehmoment 1 (Antrieb p1512 ungleich 0% (empfohlen 100%) betragen!

Hinweis Um Systemlast zu sparen sind die Drehmoment-Funktionen über das Bit „boEnableTorqueFunctions“ in der „axisConfigData“ Struktur per Default deaktiviert.

Zur Verwendung muss „boEnableTorqueFunctions“ = TRUE gesetzt werden.

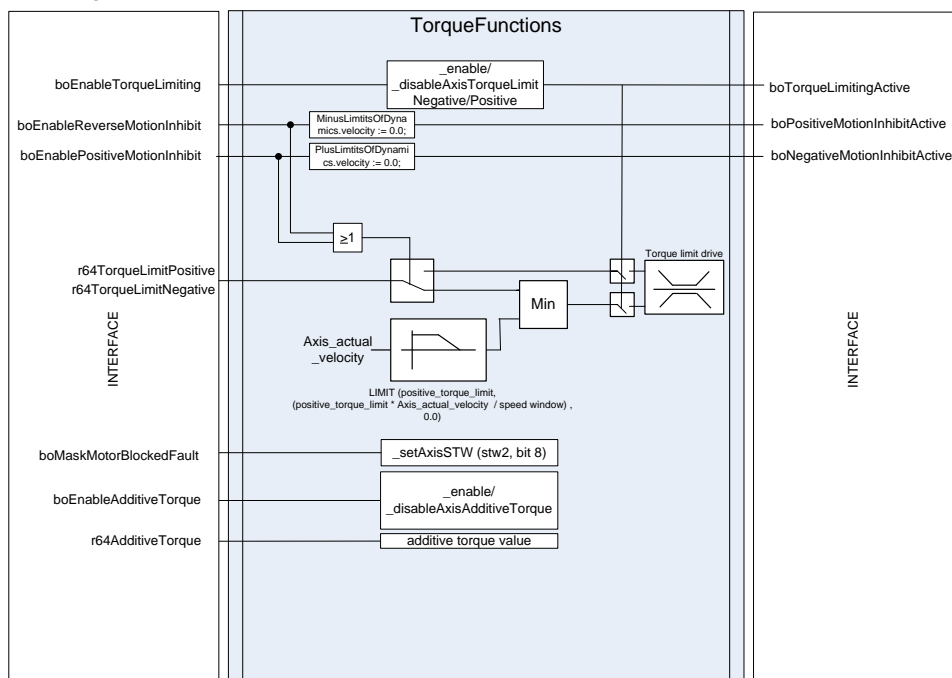
Hinweis Die STDCIO Struktur erwartet die Drehmomentwerte in [%]. Diese werden intern in [Nm] umgerechnet - abhängig von der Einstellung in „axisTOConfigData.sInternal.eTorqueReference“, die wiederum vom dem Achs-Konfigdatum „activeConfigData.TypeOfAxis.SetpointDriverInfo.DriveData.TorqueReference“ abhängt.

Hinweis Die Bezugswerte für das Drehmoment müssen in SIMOTION und SINAMICS gleich sein. d.h., dass der TO Achse Konfigurationswert max. torque (TypeOfaxis.SetPointDriverInfo.DriveData.maxTorque) und p2003 in SINAMICS gleich sein müssen.

Bei der automatischen Adaption ist das gewährleistet.

Die folgende Abbildung zeigt ein Funktionsbild der Drehmoment Funktionen.

Abbildung 4-12 Drehmoment Funktionen des Print Standard / der STDc Schnittstelle



4.12.1 Drehmoment Begrenzung

Die Variable „STDcIO.IN.boEnableTorqueLimiting“ aktiviert die positive und negative Drehmomentgrenze. Beide Grenzwerte sind an der STDc Schnittstelle parametrierbar („r64TorqueLimitNegative“ / „r64TorqueLimitPositive“). Werte von 0 bis zum maximal in SINAMICS parametrierten Drehmoment sind einstellbar.

Die Applikation rechnet mit absoluten Werten und sendet diese vorzeichenbehaftet zur Achse. Damit ist ausgeschlossen, dass der negativen Grenze ein positiver und der positiven ein negativer Wert zugewiesen wird.

Wird die Drehmomentgrenze erreicht wird dies über die Schnittstellenparameter „STDcIO.OUT.boNegativeLimitReached“ / „STDcIO.OUT.boPositiveLimitReached“ angezeigt.

Hinweis Erreicht die Achse/Antrieb eine Drehmomentgrenze und das Bit „STDcIO.IN.boMaskMotorBlockedFault“ ist nicht gesetzt (= FALSE), erzeugt SINAMICS den Motorfehler „Motor blockiert“ und stoppt den Antrieb.

Mit „boMaskMotorBlockedFault“ = TRUE, kann der Fehler maskiert werden.

4.12.2 Rücklauf / Vorlauf Bewegungssperre

Die Bewegungssperre verhindert, dass die Achse vorwärts oder rückwärts drehen kann. Dies ist über zwei Wege realisiert:

- Begrenzung der positiven bzw. negativen Dynamik über die Achs-Systemvariablen „axis.plus(minus)LimitsOfDynamics.velocity = 0.0
- Begrenzung des Drehmoments innerhalb eines Drehzahlfensters

Wenn die Achse während der Verzögerung in das, über die Variable „AxisConfigData.r64SpeedWindow“ definierte Drehzahl Fenster einfährt, wird das Drehmoment – mit sinkender Geschwindigkeit - vom aktuellen Wert auf null reduziert.

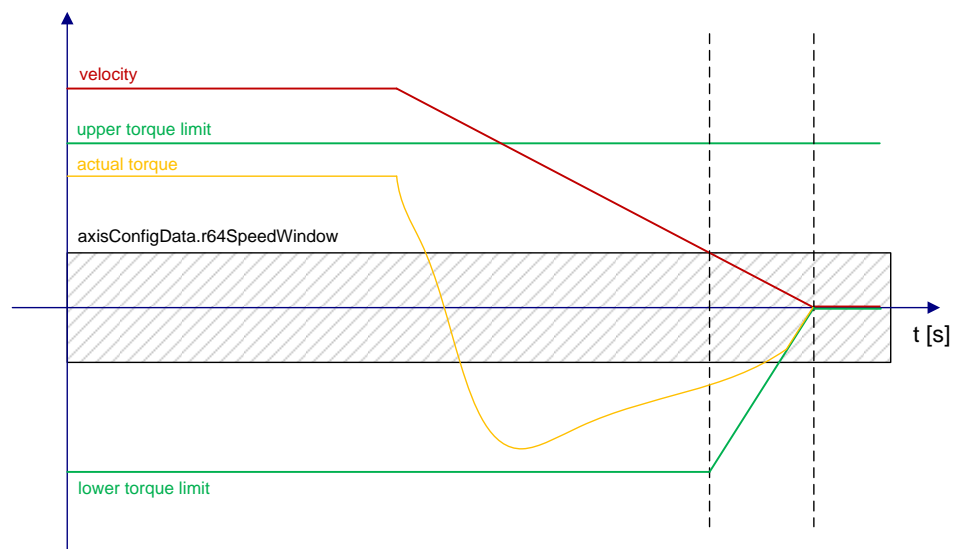
ACHTUNG Die Variable „AxisConfigData.r64SpeedWindow“ erwartet die Geschwindigkeit in der Einheit der MotionStateData.commandVelocity am TO.

Die Funktionalität wird über die Parameter „STDCIO.IN.boEnableReverseMotionInhibit“ bzw. „STDCIO.IN.boEnablePositiveMotionInhibit“ ein-/ausgeschaltet.

Hinweis Zur Verwendung der Rücklauf / Vorlauf Bewegungssperre müssen die Drehmomentfunktionen über den Parameter „axisConfigData.boEnableTorqueFunctions“ eingeschaltet sein!
Außerdem muss an der Achse der Technologiedatenblock aktiviert sein!

Das folgende Bild zeigt das Verhalten: Wenn die Drehzahl in das Drehzahlfenster eintaucht, zieht der FB die Drehmomentgrenzen bis zum Stillstand der Achse zu.

Abbildung 4-13 Drehmoment Reduzierung innerhalb Drehzahl Fenster



ACHTUNG	Durch die Verzögerung/Trägheit ist noch eine minimale Bewegung in die begrenzte Richtung möglich bevor die Achse zum Stillstand kommt,
----------------	--

ACHTUNG	Zur Benutzung der Rücklaufsperrung muss an der Achse das richtungsabhängige Geschwindigkeitsprofil aktiviert sein. (Achse -> Begrenzungen -> Haken bei „Direction dependent dyn. response“ setzen) (bzw. <code>TypeOfAxis.DecodingConfig.DirectionDynamic = YES</code>)
----------------	--

4.12.3 Drehmoment Zusatzsollwert

Für einige technologische Funktionen ist ein Drehmoment Zusatzsollwert notwendig. Die Variable „STDcIO.IN.boEnableAdditiveTorque“ gibt dazu den Wert „STDcIO.IN.r64AdditiveTorque“ frei und verschaltet diesen mit der Achs Systemvariable „defaultAdditiveTorque“.

Hinweis	Die STDcIO Struktur erwartet den Drehmomentzusatzsollwert in [%]. Dieser wird intern in [Nm] umgerechnet - abhängig von der Einstellung in „axisTOConfigData.sInternal.eTorqueReference“, die wiederum vom dem Achs-Konfigdatum „activeConfigData.TypeOfAxis.SetpointDriverInfo.DriveData.TorqueReference“ abhängt.
----------------	---

Hinweis	Bei der Benutzung des Zusatzsollwertes ist zu beachten, dass der Parameter p1512 (Skalierung des Zusatzsollwertes) auf 100% gesetzt werden muss.
----------------	--

4.13 Technologie-Erweiterungen: Wickler, Zugregelung und Register Regelung

Die bisher beschriebene (Kern-)Funktionalität des Print Standard kann durch zusätzliche Programme („Tech-Task“) und Technologieobjekte (TO's) erweitert werden.

Es sind z.B. die folgenden zusätzlichen Funktionalitäten realisierbar:

- Wickler
- Zugregelung
- Registerregelung

Für die technologischen Erweiterungen sind zusätzliche FB-Instanzen und deren Aufrufe, Variablenstrukturen sowie StartUp-Task Vorbelegungen in der jeweiligen Achs-Unit notwendig. Diese sind mit den Basis-Programmen im Print Standard nicht verknüpft, so dass sie je nach Bedarf ohne großen Aufwand hinzugefügt oder entfernt werden können.

Die Erweiterung stellt unabhängig von der zu realisierenden Applikation, eine technologische Schnittstelle bereit, um in dem Standard Sollwert-Kanal bezüglich Drehzahl (lagegeregelt Drehzahl) und Drehmoment Einfluss zu nehmen.

Dazu stellt der „FBLPrint_TechAxis“ der Print Standard Bibliothek „LPrint“ eine generische und standardisierte technologische Schnittstelle bereit unter Verwendung folgender SIMOTION System-Funktionalitäten/Eigenschaften:

- Addierobjekt
- Formelobjekt
- „motionIn“ (Systemvariablen Eingang des Achs TO)

4.13.1 Das „Tech-FB“ – Konzept (FBLPrint_TechAxis)

Der Funktionsbaustein „FBLPrint_TechAxis“ standardisiert den notwendigen technologischen Zugriff auf den Sollwert Kanal einer Achse für viele Anwendungen in Druckmaschinen. Dieser Funktionsbaustein liefert verschiedene TO-Schnittstellen um damit Drehzahl/Drehmoment der Achse zusätzlich zur Hauptbewegung zu beeinflussen.

Soll beispielsweise ein Wickler, eine Registerregelung oder eine Zugregelung implementieren werden, ruft der Nutzer die jeweiligen Funktionsbausteine der zu realisierenden Applikation und den „FBLPrint_TechAxis“ auf.

Folgende Punkte müssen bei der Implementierung beachtet werden:

- Aufruf des „FBLPrint_TechAxis“ in der IPO synchronen Task
- Verbinden der technologischen Funktion (z.B. Wickler) mit den Eingängen des „FBLPrint_TechAxis“

4 Funktionsbeschreibung

- Anlegen der notwendigen zusätzlichen TO's (z.B. Addier- oder Formelobjekte), korrekte Verschaltung mit den Achs-TO's sowie die Parametrierung (z.B. Achszyklus)
- Der Technologiedatenblock am Achs-TO muss aktiviert sein (z.B. für Verwendung der Drehmoment-Werte)

Abbildung 4-14 Tech-FB - Konzept

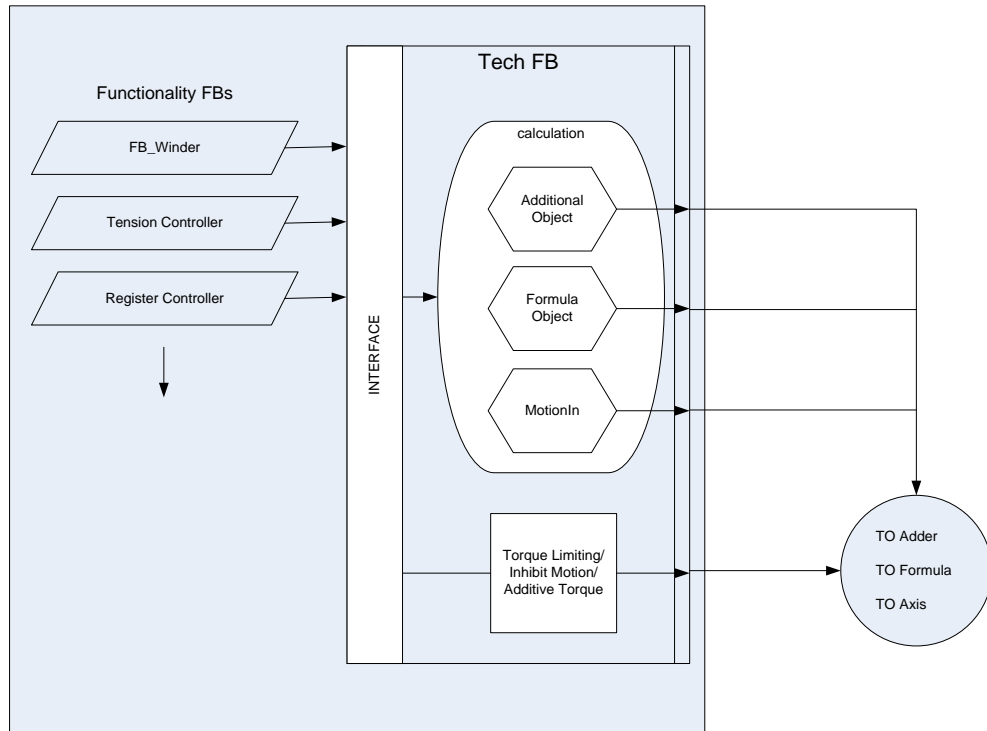
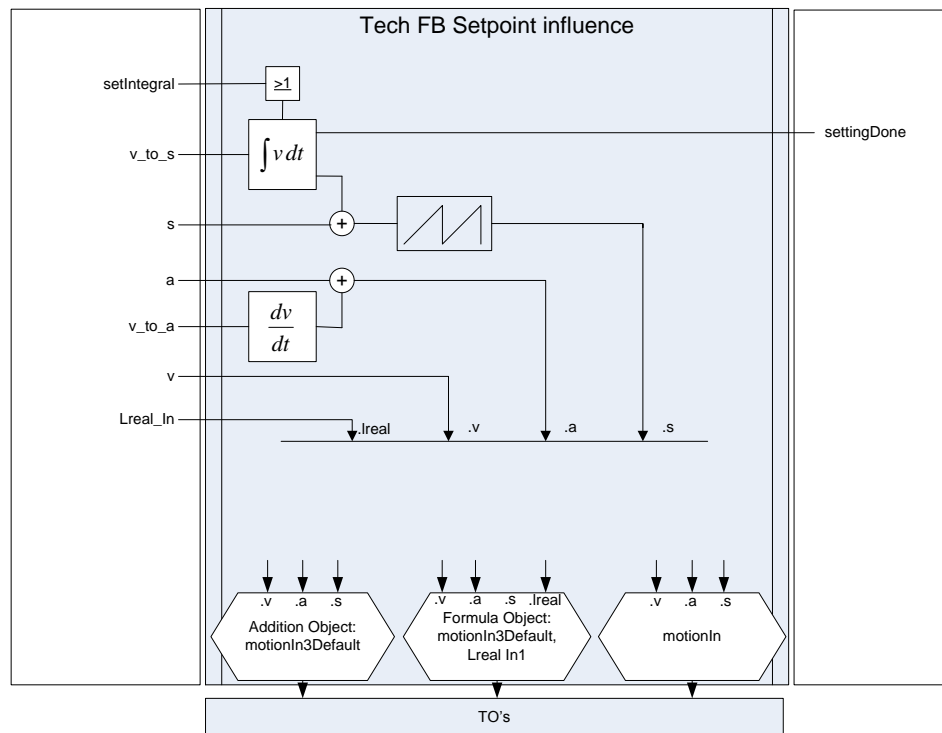


Abbildung 4-15 Bewegungsvektor „FBLPrint_TechAxis“



Die vorhergehenden Bilder zeigen eine Prinzip Darstellung zur Schnittstellenfunktion des „FBLPrint_TechAxis“ zwischen Technologie Programmen und den TO's des Sollwert-Kanals.

Abhängig von der zu realisierenden Funktion werden im „FBLPrint_TechAxis“ die Teile oder der komplette Motion Vektor (s, v, a) berechnet, die zur Verknüpfung an das Addierobjekt, Formelobjekt oder dem Achs-„MotionIn“ Eingang notwendig sind.

Der Baustein wird nur berechnet wenn die Auswertung der Eingangsvariable „axisTOConfigData.sInternal.boValid“ signalisiert, dass die Technologie Objekte gültig sind.

Fehler des Bausteins werden in der Print Standard STDclO Schnittstelle angezeigt und in den Fehlerpuffer „AxisData.sAppMsgLog“ eingetragen.

Da bei Addier- und Formelobjekten mehrere motionInDefault-Eingänge (Addierobjekt 4, Formelobjekt 3) zur Verfügung stehen, muss der zu beschreibende Eingang über die Input-Variable „miDefaultChannel“ vorgegeben werden.

Hinweis Der „MotionIn1“ ist in der Regel durch die feste Kopplung an den Master belegt.

Hinweis Addier- und Formelobjekte haben sowohl „MotionIn...“ als auch „MotionIn...Default“ Eingänge.
 „MotionIn...“ Eingänge werden über die Verschaltung am TO beschrieben.
 „MotionIn...Default“ Eingänge können aus der Applikation heraus beschrieben werden. Sie müssen nicht extra aktiviert werden!

Hinweis Der „FBLPrint_TechAxis“ hat Eingänge für einen kompletten Motion Vektor (Lage s, Drehzahl v, Beschleunigung a).
 Falls nur ein Drehzahlwert „v“ verfügbar ist (z.B. Ausgang eines Zugreglers), dann muss die Drehzahl „v“ auch an die Eingänge „v_to_s „ und „v_to_a“ angeschlossen werden, damit im Baustein der komplette Motion Vektor erzeugt werden kann.

Hinweis Es ist möglich beide Eingangsquellen d.h. den kompletten Motion Vektor „s, v, a“ und „v_to_s, v_to_a“ gleichzeitig zu benutzen.
 Dies führt im „FBLPrint_TechAxis“ zu einer Überlagerung und wird z.B. verwendet wenn zur gleichen Zeit via Sollwertkanal Eingriffe von einem Zugregler und Registerregler implementiert werden sollen.

Tabelle 4-25 IO Schnittstelle des „FBLPrint_TechAxis“

Name	Datentyp	Type	Beschreibung
enable	BOOL	IN	TRUE: Freigabe des FB
v	LREAL	IN	Motion Vector Eingang: Drehzahl [°/s]
a	LREAL	IN	Motion Vector Eingang: Beschleunigung [°/s ²]
s	LREAL	IN	Motion Vector Eingang: Position [°]
v_to_a	LREAL	IN	Motion Vector Berechnung: vom zusätzlichen Drehzahlsollwert [°/s], wird die zusätzliche Beschleunigung differenziert
v_to_s	LREAL	IN	Motion Vector Berechnung: vom zusätzlichen Drehzahlsollwert [°/s], wird die zusätzliche Position integriert
lreal_In	LREAL	IN	LREAL Eingang verbunden mit dem Formel Objekt
setIntegral	BOOL	IN	FALSE -> TRUE: Positionswert (set_value) wird gesetzt
setValue	LREAL	IN	Setzwert zum Position setzen
miDefaultChannel	MotionInDefault	IN	Auswahl des MotionInDefault-Eingangs am Addier- oder Formelobjekt
taskMode	eTaskNameType	IN	Angabe der Task in der der FB aufgerufen wird
axisData	sAxisDataType	IN/OUT	Print Standard axisData Struktur
axisTOConfigData	sAxisTOConfigData Type	IN/OUT	Print Standard axisTOConfigData Struktur
valid	BOOL	OUT	Tech-FB ist in Betrieb nach erfolgreicher Konsistenzüberprüfung

4 Funktionsbeschreibung

Name	Datentyp	Type	Beschreibung
integralValue	LREAL	OUT	Aktueller Integral-Wert [°]
settingDone	BOOL	OUT	Rückmeldung, dass das Position setzen ausgeführt wurde

4.13.2 Beispiel-Anwendungen mit dem Tech-FB Konzept

Im Folgenden werden einige Beispiel-Applikationen zur Verwendung des „Tech-FB-Konzepts“ gezeigt:

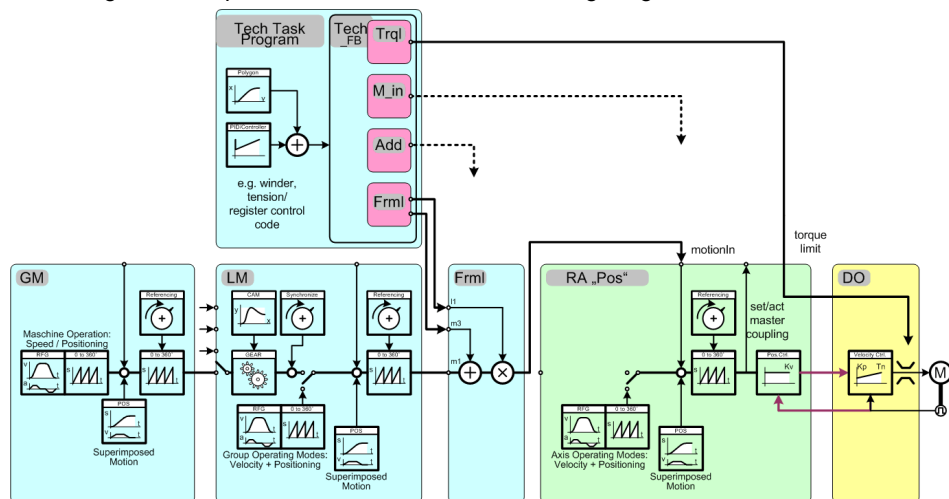
Beispiel Wickler: Nutzung des Formelobjekts und Achs-MotionIn

Die Wickler Applikation beeinflusst den Drehzahlollwert mit dem TO Formelobjekt. Der Ausgang des Formelobjekts wird dabei direkt auf den „motionIn“ Eingang der entsprechenden Achse verschalten.

Der „FBLPrint_TechAxis“ versorgt die zusätzliche Drehzahlkomponente unter Nutzung des „motionin3default“ des Formelobjekts. Zusätzlich wird der Faktor aus der Durchmesserberechnung über den „FBLPrint_TechAxis“ Eingang „lreal_In“ an den Formel Objekt Eingang „LrealIn1“ verbunden.

Die notwendige Formel Berechnung zwischen der Hauptbewegung (basic motion, „motionIn1“), Zusatzdrehzahl („motionIn3default“) und dem Durchmesserfaktor („lreal1default“) ist im Applikationsprogramm definiert. Für den Wickler im Funktionsbaustein „FBLPrint_MotionAxis“ in der Betriebsart „WINDING“.

Abbildung 4-16 Beispiel der Sollwert Kanal Erweiterung/Eingriff beim Wickler



Beispiel Seiten Register: Nutzung des Achs-MotionIn

Für die Realisierung einer Seitenregister-Regelung ist z.B. der Einsatz einer Positionier-Achse und der direkte Eingriff via „FBLPrint_TechAxis“ an den „MotionIn“ Eingang dieser Achse möglich.

Der Achs TO Eingang „defaultmotionin.velocity“ kommt dabei direkt vom „FBLPrint_TechAxis“ Eingang „v“.

Hinweis

Die „motionIn“ Eingangswerte werden vom „FBLPrint_TechAxis“ nur dann berechnet, wenn ein „_runvelocitybasedmotionin“ Befehl abgegeben wurde.

„AxisTOConfigData.sInternal.toDriveAxis.velocitybasedmotionincommand.motion state <> INACTIVE!

Beispiel Zug/Register Regelung: Nutzung des Addierobjekts

Bei der Register- und Zugregelung wird ein Addierobjekt verwendet. Dieses TO bildet aus verschiedenen Bewegungsvektoren an seinen Eingängen einen resultierenden Ausgangsvektor (s, v, a).

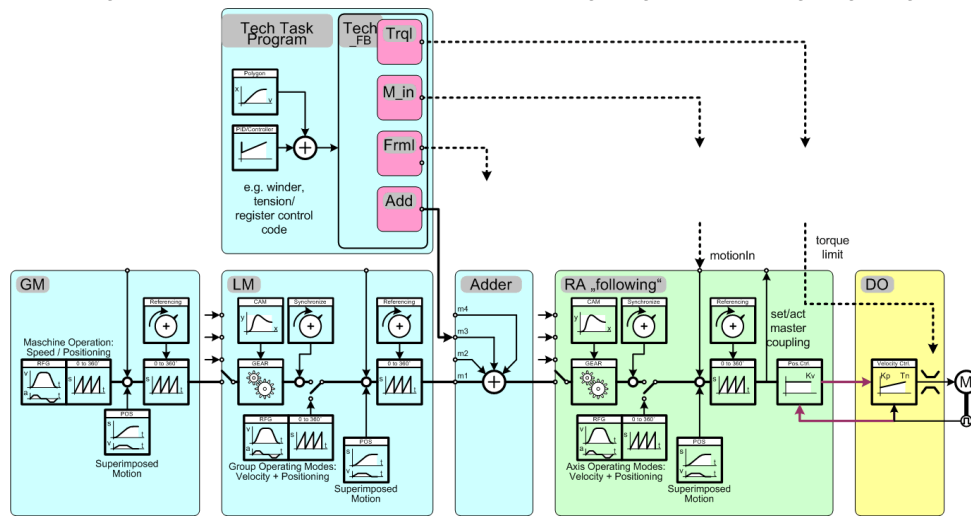
Ein Gleichlaufobjekt kann auf das Addierobjekt als Master aufsynchronisieren, wenn die Verbindung im Scout konfiguriert wurde. Das Addierobjekt hat die gleiche Leitwertfunktionalität wie ein Achs-TO als Leitwert (Master).

Mit dem Addierobjekt wird eine Addierstelle in den Sollwertkanal der Achse bereitgestellt. Der „FBLPrint_TechAxis“ bereitet dafür einen kompletten Bewegungsvektor auf. Dieser wird mit dem Addierobjekt-Eingang „motionIn3default“ verbunden.

Die Lage/Position des Bewegungsvektors kann mit den „FBLPrint_TechAxis“ Eingängen „setIntegral“ und „setValue“ gesetzt werden. Das Setzen ist nur möglich wenn die Folgeachse noch nicht eingekuppelt ist oder gerade eingekuppelt/aufsynchronisiert.

Eine detaillierte Beschreibung zur Integration der Zugregelung in den Print Standard kann dem Kapitel [4.13.3 Integration der Zugregelung](#) entnommen werden.

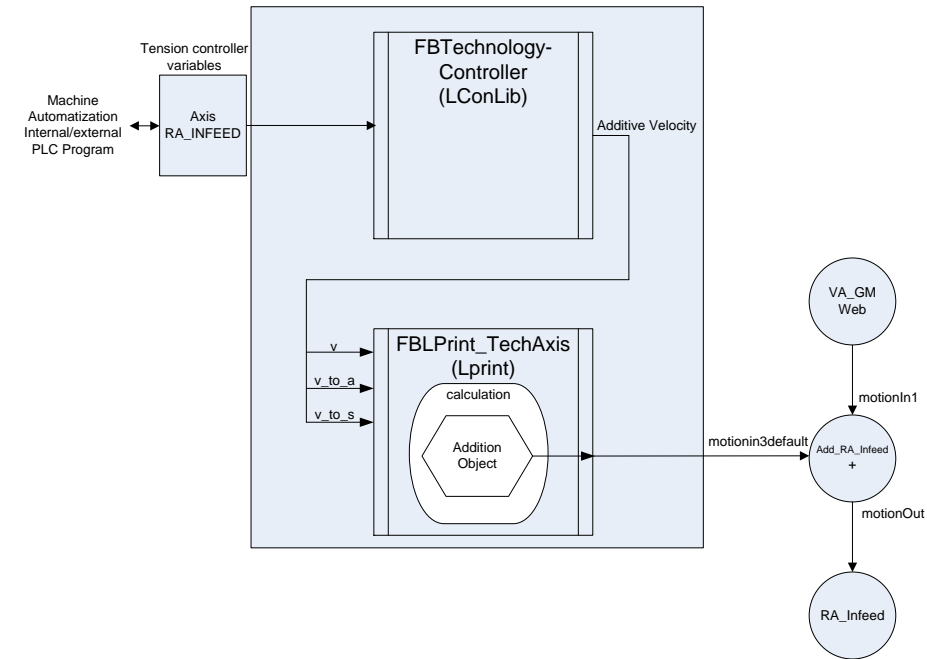
Abbildung 4-17 Beispiel der Sollwert Kanal Erweiterung/Eingriff bei der Zug Regelung



4.13.3 Integration der Zugregelung

Im folgenden Abschnitt wird die Integration der Zugregelung in den Print Standard, wie sie im Beispielprojekt exemplarisch für die Achse „RA_Infeed“ gemacht wurde, näher erläutert. Das folgende Bild zeigt das Konzept zur Integration der Zugregelung mittel Zugreglerbaustein und „FBLPrint_TechAxis“.

Abbildung 4-18 Integration der Zugregelung über die Tech Task



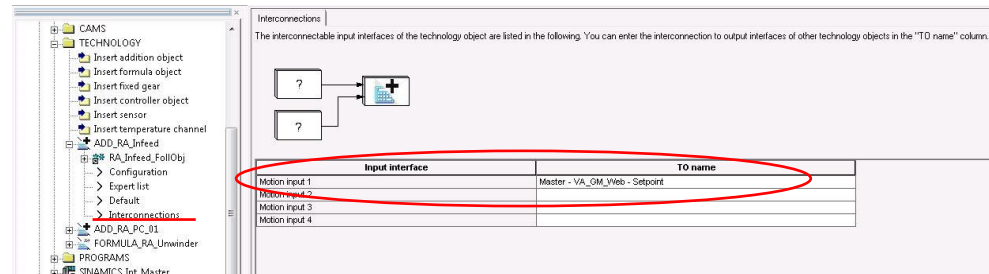
Copyright © Siemens AG 2016 All rights reserved

TO-Verschaltungen & Konfiguration

Grundlegend für die Funktion des Zugreglers sind die richtigen TO-Verschaltungen in den Konfigurationsmasken der TOs. Hierzu müssen die folgenden Verbindungen gesetzt werden:

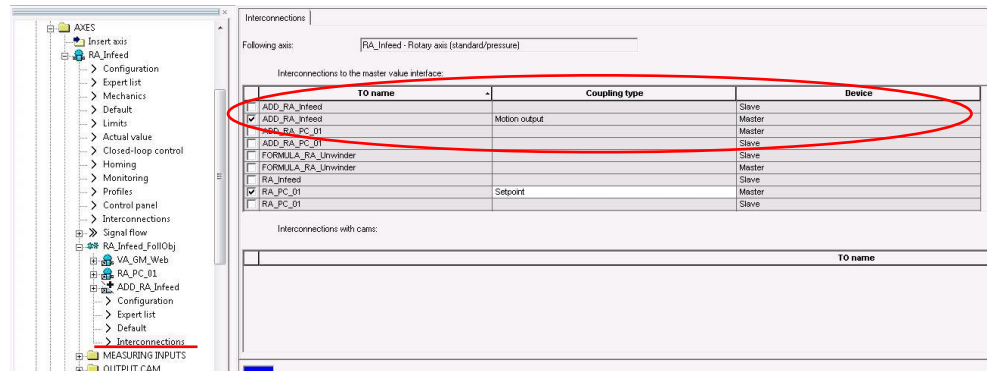
- Leitachse (VA_GM_Web) <-> Addierobjekt (ADD_RA_Infeed)

Abbildung 4-19 Verschaltung Addierobjekt auf die Leitachse



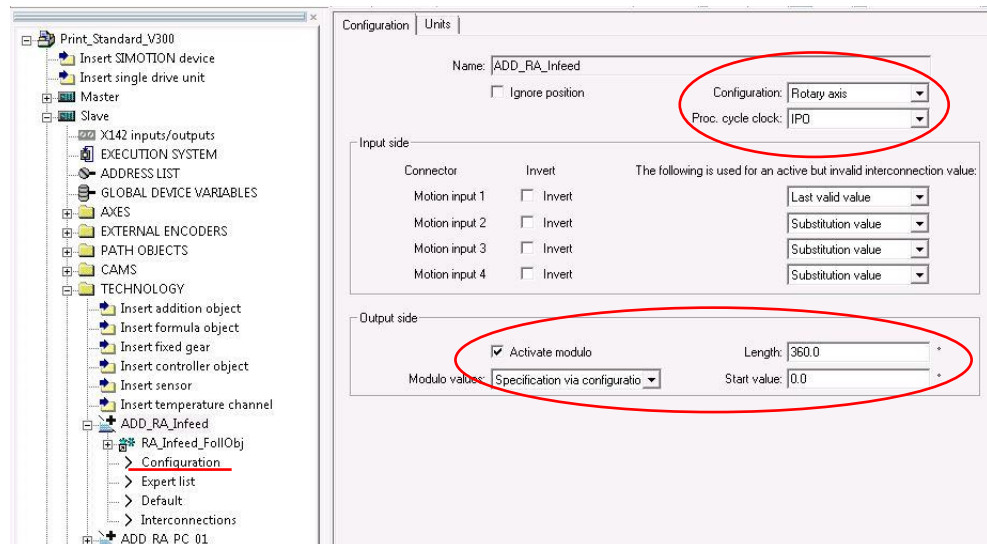
- Addierobjekt (ADD_RA_Infeed) <-> Folgeachse (RA_Infeed_FollObj)

Abbildung 4-20 Verschaltung Folgeachse auf das Addierobjekt



Neben den richtigen Verschaltungen des Addierobjektes ist darauf zu achten, dass das Addierobjekt der grundlegenden Konfiguration der Achse entspricht (z.B. Modulolänge 360°).

Abbildung 4-21 Konfiguration Addierobjekt



Copyright © Siemens AG 2016 All rights reserved

Variablen und Programme

Die globalen Variablen, FB-Instanzen, FB-Aufrufe sowie StartUp-Vorbelegungen werden in die entsprechenden Achs-Unit integriert.

Benötigt wird der Zugreglerbaustein, der „FBLPrint_TechAxis“ und eine Konfig-Variablenstruktur vom Typ „sTensionType“.

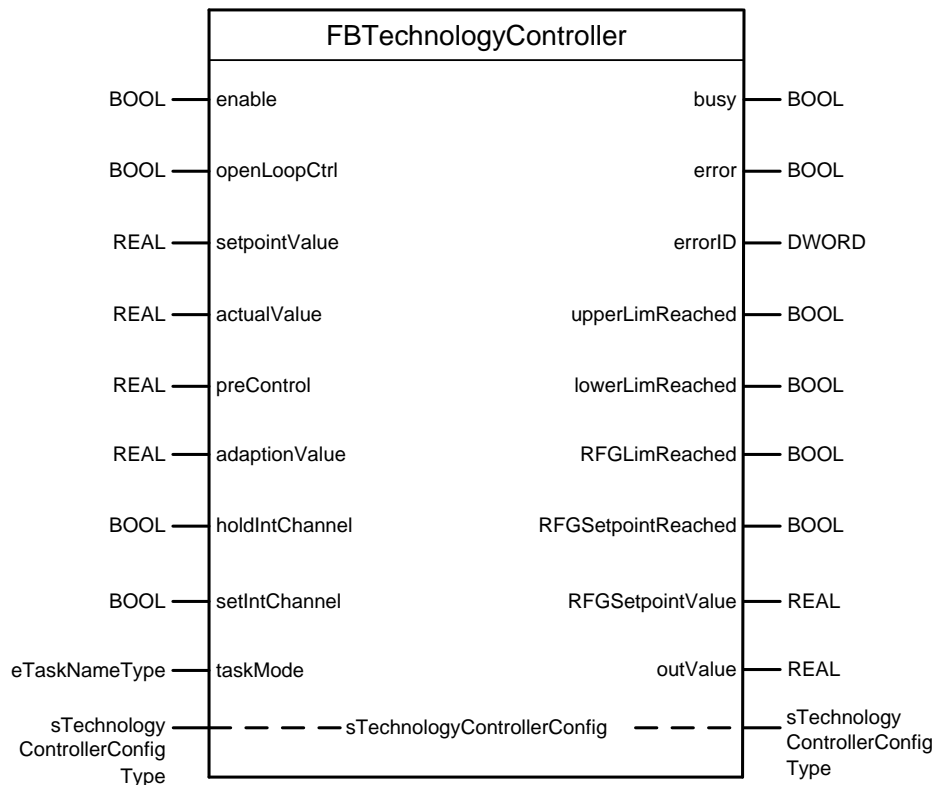
Im Zugreglerbaustein wird die nötige Zusatzgeschwindigkeit zum Erreichen des Zugsollwertes errechnet. Aus der Geschwindigkeit wird im „FBLPrint_TechAxis“ ein Geschwindigkeitsvektor gebildet und über das Addierobjekt mit dem Geschwindigkeitsvektor der Leitachse verrechnet. Die resultierende Geschwindigkeit wird über den Ausgang des Addierobjektes als Sollwert an die Folgeachse weitergegeben.

Technologieregler („FBTechnologyController“)

Die Converting Bibliothek „LConLib“ bietet mit dem „FBTechnologyController“ einen Technologieregler auf Basis eines PID-Reglers zur Verwendung als Zug- oder Tänzerregler. Er beinhaltet folgende Funktionalität:

- Sollwerthochlaufgeber mit Setzfunktion
- PID-Regler
- Rampenfunktion als Reglerausgangsbegrenzung
- Kp-Adaption
- D-Anteil optional zuschaltbar im Istwertzweig

Abbildung 4-22 FBTechnologyController (LConLib)



Hinweis

Eine ausführliche Beschreibung der Funktionalität und der Variablen des „FBTechnologyController“ ist dem Handbuch der Converting Bibliothek zu entnehmen.

Hinweis

Aufgrund unterschiedlicher Typ-Namen muss ausgewählt werden ob die „alte“ oder „neue“ Converting Bibliothek verwendet wird. Über eine Prä-Processor Anweisung in LPrint\dTechData kann gewählt werden:

```
#define LConLib_1xx für LConLib Version <= 1.3.0
```

```
#define LConLib_2xx für LConLib Version >= 2.0.0
```

Die Folgende Tabelle beschreibt die Struktur „sTensionType“ die in der Druck Standard Bibliothek „LPrint“ in der Quelle „dTechData“ definiert ist.

Sie fasst die wichtigsten Ein- und Ausgangsparameter des Bausteins „FBTechnologyController“ inklusive der „sTechnologyControllerConfigType“ Struktur zusammen.

Für jede zuggeregelte Achse muss eine globale Strukturvariable vom Typ „sTensionType“ angelegt werden.

Tabelle 4-26 sTensionType

Name	Datentyp	Type	Beschreibung
boEnable	BOOL	IN	FALSE -> TRUE: Hochlaufgebers werden gestartet, PIDRegler ist aktiv (wenn nich im openLoop Modus) TRUE -> FALSE: Sollwert wird nach Istwert integriert, Reglerausgang wird begrenzt, dann wird der FB deaktiviert.
boOpenLoopControl	BOOL	IN	FALSE: PID-Regler aktiv TRUE: PID-Regler und Begrenzung-HLG sind inaktiv. Es wird nur Sollwert-HLG verwendet
r32SetValue	REAL	IN	Sollwertsignal
r32ActValue	REAL	IN	Istwertsignal
r32PreControl	REAL	IN	Vorsteuersignal
r32AdaptionValue	REAL	IN	Signal für kp-Adaption
r32IntSetValue	REAL	IN	Setzwert Integral-Anteil
boHoldIntChannel	BOOL	IN	TRUE: I-Anteil wird eingefroren
boSetIntChannel	BOOL	IN	TRUE: I-Anteil setzen
sTechnologyControllerConfig	sTechnology ControllerCo nfigType	IN	Konfigurationsdaten des Technologiereglers
boBusy	BOOL	OUT	FALSE: FB ist deaktiviert TRUE: FB ist aktiviert
boError	BOOL	OUT	FALSE: kein Fehler TRUE: Fehler steht an siehe Fehlerkennung
b32ErrorID	DWORD	OUT	Fehlerkennung, siehe Fehlermeldungen FBLPrint_TechnologyController
boUpperLimitReached	BOOL	OUT	TRUE: Ausgang ist begrenzt, obere Grenze
boLowerLimitReached	BOOL	OUT	TRUE: Ausgang ist begrenzt, untere Grenze
r32OutValue	REAL	OUT	Reglerausgang

Test der Zugregelung im Beispielprojekt

Der für die Achse „RA_Infeed“ im Beispielprojekt integrierte Zugregler kann zur Veranschaulichung der Funktion verwendet werden. Um die Zugregelung zu aktivieren müssen folgende Schritte durchgeführt werden:

- Achse in Mode 60 oder 61 schalten
- Achse auf Master auf synchronisieren
- Zugregler einschalten („boEnable“ in der „sTensionType“ Struktur)

Der Analogeingang „AI1“ der im Beispielprojekt integrierte TB30 simuliert den Zug-Istwert. Er ist normiert auf 0...100%, geregelt wird auf 50% (Potentiometer auf Mittelstellung 5).

4.13.4 Integration des Wicklers

Die Integration eines Wicklers in den Print Standard ist vergleichbar mit der Integration der Zugregelung (siehe Kapitel [4.13.3 Integration der Zugregelung](#)). Anstelle des Addierobjekts ist jedoch ein Formelobjekt notwendig, um den sich ändernden Durchmesser und die damit verbundene Geschwindigkeitsänderung der Wicklerachse in den Sollwertkanal einzuspeisen.

Die im Baustein FBLPrint_MotionAxis in der Betriebsart WINDING hinterlegte Formel lautet:

$$mo1.v = (mi1.v + mi3.v) * li1$$

Zusätzlich ist eine Verschaltung des Formelobjekt-Ausgangs „Motion output 1“ mit dem Motion Input der Wicklerachse vorzunehmen.

Abbildung 4-23 Verschaltung Wicklerachse - Formelobjekt

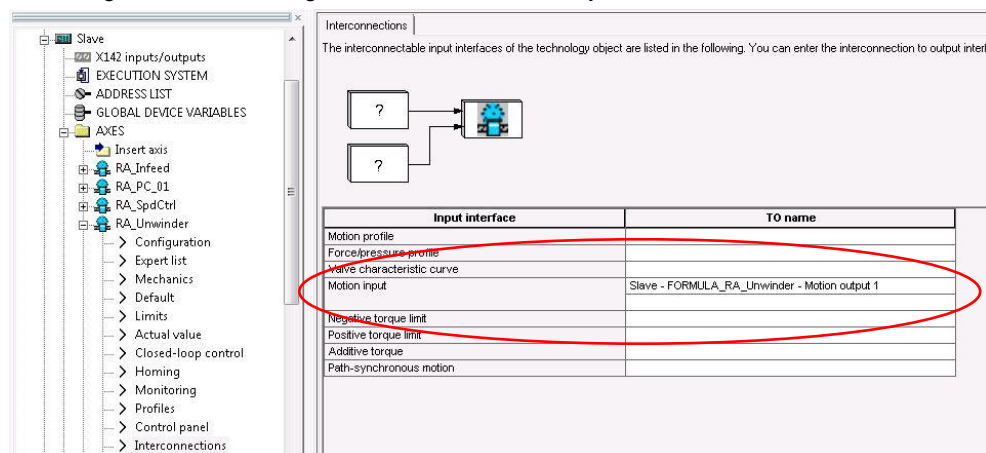
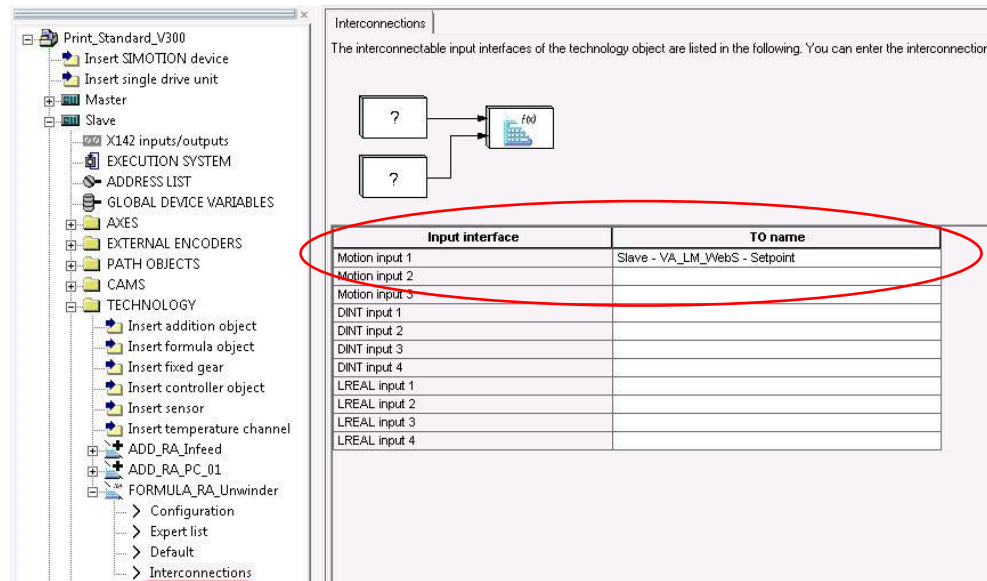


Abbildung 4-24 Verschaltung Formelobjekt – Master-Sollwert



Hinweis

Im Beispielprojekt wird der Wickler Baustein „FBWinder“ der Converting Bibliothek „LConLib“ eingesetzt.

Detaillierte Informationen dazu können den Dokumentationen zum Wickler und zur Converting Bibliothek entnommen werden. Beides kann über SIOS bezogen werden.

Hinweis

Aufgrund unterschiedlicher Typ-Namen muss ausgewählt werden ob die „alte“ oder „neue“ Converting Bibliothek verwendet wird. Über eine Prä-Processor Anweisung in LPrint\dTechData kann gewählt werden:

```
#define LConLib_1xx für LConLib Version <= 1.3.0
#define LConLib_2xx für LConLib Version >= 2.0.0
```

Die Folgende Tabelle beschreibt die Struktur „sWinderType“ die in der Druck Standard Bibliothek „LPrint“ in der Quelle „dTechData“ definiert ist.

Sie fasst die wichtigsten Ein- und Ausgangsparameter des Wickler-Bausteins inklusive der „sWinderConfig“ Struktur zusammen.

Pro Wicklerachse muss eine globale Strukturvariable vom Typ „sWinderType“ angelegt werden.

Tabelle 4-27 sWinderType

Name	Datentyp	Type	Beschreibung
boEnableWinder	BOOL	IN	Wickler Funktionsbaustein aktivieren TRUE: Freigabe der Schnittstellen und Berechnung der relevanten Ausgangsdaten FALSE: Keine Berechnung der Ausgangswerte

4 Funktionsbeschreibung

Name	Datentyp	Type	Beschreibung
boEnableCtrl	BOOL	IN	TRUE: Wickelbetrieb aktivieren FALSE: Wickelbetrieb deaktivieren, Sollwerte werden zurückgeführt
boSetDiamCalc	BOOL	IN	TRUE: Setze Durchmesser auf vorgegebenen Wert (r32DiamSetVal). Nur bei busy = FALSE oder im Stillstand bei boEnableCtrl = FALSE und boCtrlBusy = FALSE möglich
boHoldDiamCalc	BOOL	IN	TRUE: Halte aktuellen Durchmesser
r32PositionSetpoint	REAL	IN	Tänzerlagesollwert, wirkt als Sollwert für Technologieregler bei Tänzerlageregelung. Keine Verwendung bei Zugregelung.
r32DiamSetVal	REAL	IN	Durchmessersetzwert
r32ActValuePID	REAL	IN	Rückführung für die Zug- oder Tänzerlageregelung (Zugistwert oder Tänzerlageistwert).
eTypeOfWinder	'eWindingMode Type'	IN	Wickelmodus (bei positiver Bahngeschwindigkeit): UNWIND_FROM_ABOVE UNWIND_FROM_BELOW REWIND_FROM_ABOVE REWIND_FROM_BELOW Die Änderung des Eingangs ist nur bei boEnable = FALSE oder im Stillstand bei boCtrlEnable = FALSE und boCtrlBusy = FALSE wirksam.
sWinderConfig	'sWinderConfigType'	IN/OUT	Struktur für Wickler Konfigurationsdaten Details zu dieser Struktur befinden sich in der SIMOTION Winder Dokumentation.
r32LatchDiamCalc	REAL	IN/OUT	Speicherung des letzten berechneten Durchmessers (Retain Variable)
boBusy	BOOL	OUT	FALSE: FB nicht aktiv TRUE: FB ist aktiv und arbeitet
boActive	BOOL	OUT	FALSE: Wickler ist nicht aktiv TRUE: Wickler ist aktiv und generiert Sollwerte
boError	BOOL	OUT	FALSE: kein Fehler TRUE: Es steht ein Fehler an, siehe „b32ErrorID“ Die Bearbeitung und Berechnung der Prozesswerte wird im Fehlerfall nicht unterbrochen.
b32ErrorId	DWORD	OUT	Fehlerkennung, siehe Fehlermeldungen in SIMOTION Winder Dokumentation errorID < 16#000_8000: Warnung errorID >= 16#0000_8000: Fehler
r32Diameter	REAL	OUT	Aktueller Rollendurchmesser
r32DiameterFactor	REAL	OUT	Bewertungsfaktor für die Bahngeschwindigkeit bei externer Verschaltung (Umrechnung Bahngeschwindigkeit auf Geschwindigkeit der Wickelachse)

4 Funktionsbeschreibung

Name	Datentyp	Type	Beschreibung
r32AdditiveVelocity	REAL	OUT	Zusatzgeschwindigkeit der Wickelachse in der Einheit der Leitachse (Verschaltung für Formelobjekt im Druckstandard). Vorzeichenbereinigt zur Multiplikation mit dem Ausgang „r32DiameterFactor“
r64AdditiveTorque	LREAL	OUT	Vorsteuermoment
r64TorqueLimitNegative	LREAL	OUT	Untere Momentengrenze
r64TorqueLimitPositive	LREAL	OUT	Obere Momentengrenze

4.14 Verteilter Gleichlauf und I-Device Kommunikation

Hinweis Allgemeine Informationen zum verteilten Gleichlauf via Controller-Controller-Querverkehr, sowie zur I-Device Kommunikation können den Systemhandbüchern „Gleichlauf_Technologie_Funktionen“, „Kommunikation_Funktionen“ sowie dem SIOS Beitrag <https://support.industry.siemens.com/cs/de/de/view/38486079> entnommen werden.

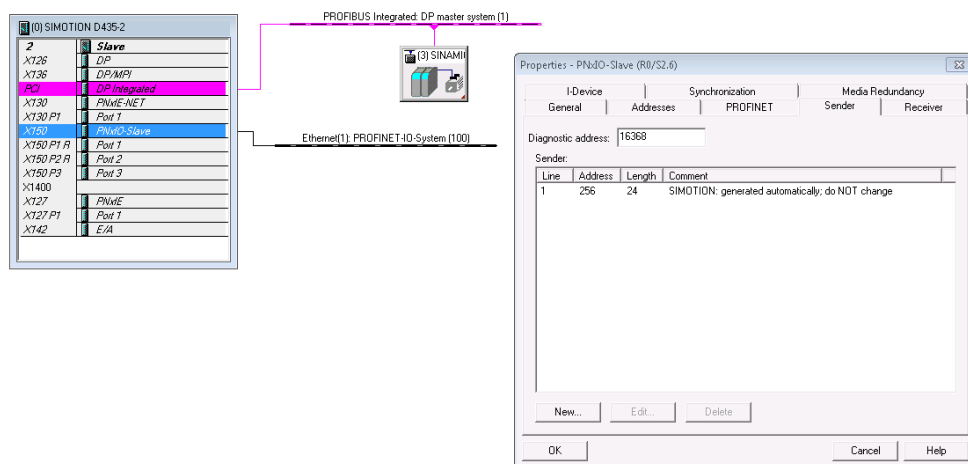
4.14.1 Verteilter Gleichlauf

Das Druck Standard Beispielprojekt zeigt den verteilten Gleichlauf zwischen zwei SIMOTION Controllern innerhalb eines SIMOTION Scout Projekts via Controller-Controller-Querverkehr.

Die Achse „VA_GM_Web“ der Master Steuerung ist der Leitwert der Achse „VA_LM_WebS“ der Slave Steuerung (siehe Verschaltungsübersicht [5.1 Print Standard Beispielprojekt TO-Verschaltung](#)).

Wird am Gleichlaufobjekt eine Verschaltung zu einer Achse eines anderen Controllers projektiert, wird in der HW-Konfiguration beider Controller automatisch die Kommunikationsverbindung projektiert.

Abbildung 4-25



Für eine projektierte Gleichlaufverbindung zwischen zwei Achsen werden 24 Byte benötigt. Insgesamt können 3072 Byte übertragen werden.

Hinweis Wie im Beispielprojekt gezeigt, sollte nach Möglichkeit der Leitwert mit nur einer Verbindung auf eine Folgeachse auf den Slave Controller (VA_LM) übertragen werden. Weitere Folgeachsen können dann diesem lokalen Master folgen. Somit wird die Kommunikationslast so gering wie möglich gehalten.

Um den DP-Takt des Slave-Controllers auf den des Masters zu synchronisieren ist auf dem Slave-Controller der Systemfunktionsaufruf

```
_enableDpInterfaceSynchronizationMode(dpinterfacesyncmode :=  
AUTOMATIC_INTERFACE_SYNCHRONIZATION);
```

notwendig.

ACHTUNG

Während des Systemfunktionsaufrufs sollten keine Achsbewegungen stattfinden, da durch die Synchronisation Sollwertsprünge entstehen können!

Im Beispielprojekt wird die Funktion in der Quelle pMStartup in einem Programm der StartUp-Task aufgerufen.

An der Master-Achse sollte die Leitwertseitige Kompensation über Sollwertausgabeverzögerung aktiviert werden, während an der(n) Folgeachse(n) die Versatzermittlung aktiviert werden muss:

```
(TypeOfAxis.)distributedMotion.enableOffsetCompensation:=YES
```

```
(TypeOfAxis.)distributedMotion.enableDelayOfCommandValueOutput:= YES
```

Der Taktversatz wird nach einem STOP → RUN Übergang automatisch ermittelt und kann über die Systemvariable distributedMotion.delayOfCommandValueOutput an der Leitachse ausgelesen werden.

4.14.2 I-Device Kommunikation

Projektübergreifender Gleichlauf oder die Kommunikation mit einer überlagerten Steuerung (z.B. SIMATIC CPU) muss über ein IRT-I-Device (GSD Datei) projiziert werden.

In Summe können pro SIMOTION Steuerung max. 1440 Bytes Eingangsdaten und max. 1440 Bytes Ausgangsdaten via I-Device übertragen werden. Außerdem kann ein SIMOTION I-Device nur einer überlagerten Steuerung zugeordnet werden.

ACHTUNG

Es ist zu beachten, dass an einer SIMOTION PN-Schnittstelle, welche als IRT I-Device fungiert, keine weiteren IRT IO-Devices betrieben werden können, nur RT IO-Devices. D. h. eine Steuerung kann an einer PN-Schnittstelle nicht gleichzeitig IRT I-Device und IRT IO-Controller sein!

Hinweis

Die Projektierung eines I-Device wird in einer separaten Dokumentation inkl. Beispielprojekt beschreiben:

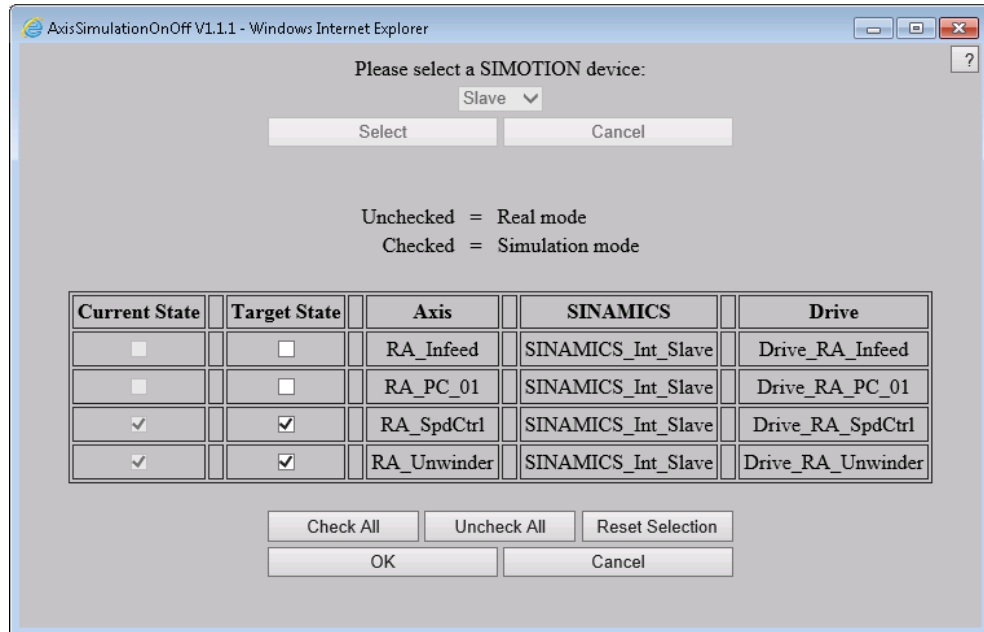
SIOS Beitrag: <https://support.industry.siemens.com/cs/de/de/view/38486079>

4.15 Antrieb/SINAMICS und TO Parametrierung via Scripting

4.15.1 AxisSimulationOnOff

Das Skript kann verwendet werden, um Achsen in den Simulations-Mode bzw. in den realen Mode zu schalten.

Abbildung 4-26 Skript AxisSimulationOnOff



Hinweis

Um Achsen aus dem Simulations-Mode zurück in den real Moden zu schalten, wird mit dem Skript auf eine Backup-Datei zugegriffen, damit die ursprünglichen Einstellungen (z.B. HW-Adressen) wiederhergestellt werden können.

Damit dieser Zugriff funktioniert, muss der Achs-Name im Projekt mit dem Achs-Namen im Backup-File übereinstimmen!

Falls das zurückschalten nicht wie gewünscht funktioniert, kann das Backup-File überprüft werden. Das File wird im Projektordner im Unterordner „BackupFiles“ abgelegt.

4.15.2 ChangeTechAlarmType

Das Skript kann verwendet werden, um den Alarm-Typ von TO Alarmen anzupassen.

Nach dem Ausführen des Skripts muss zunächst die Alarm-Nummer eingegeben werden. Danach kann der Alarm-Typ für diesen Alarm festgelegt werden.

Abbildung 4-27 Alarm-Nummer auswählen

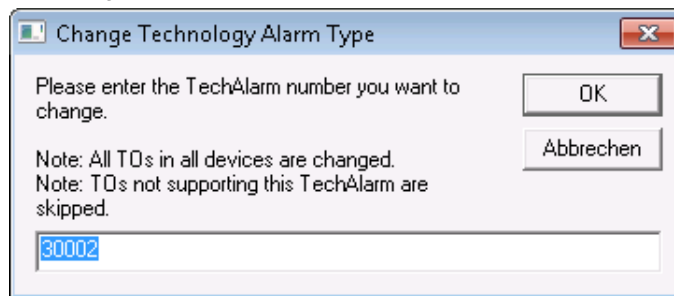
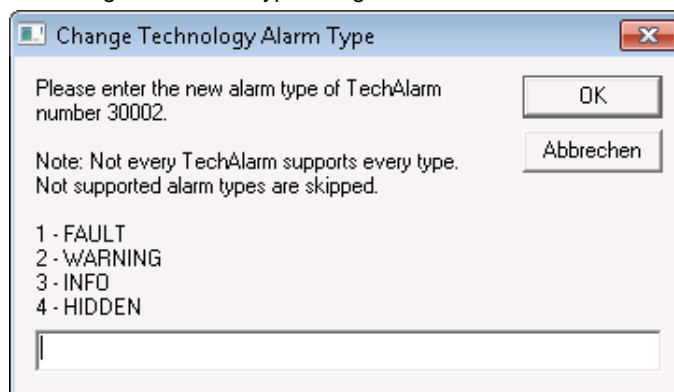


Abbildung 4-28 Alarm-Typ festlegen



4.15.3 LoggingLibrary

Wird für das Skript „ChangeTechAlarmType“ benötigt.
Der Benutzer muss dieses Skript nicht extra ausführen!

4.15.4 SetEncoderInterface

Dieses Skript kann verwendet werden, um nachträglich die SINAMICS-Gebereinstellungen in der SIMOTION anzupassen.
Dies kann notwendig sein, falls bei Projekterstellung die Geberdaten noch nicht zur Verfügung stehen oder nachträglich geändert werden.

4.15.5 SinamicsSettings_OFFLINE

Dieses Skript kann verwendet werden um Antriebsparameter zu parametrieren.

Hinweis Dieses Skript ist nur als Vorlage gedacht und muss für das jeweilige Maschinenprojekt erweitert/angepasst werden!

4.15.6 SimotionSettings_OFFLINE

Dieses Skript kann verwendet werden um TO-Einstellungen vorzunehmen.

Hinweis Dieses Skript ist nur als Vorlage gedacht und muss für das jeweilige Maschinenprojekt erweitert/angepasst werden!

5 Bedienung der Beispielapplikation

Das Print Standard Beispielprojekt ist dafür ausgelegt, auf dem SIMOTION Demokoffer lauffähig zu sein.

Im Projekt befinden sich zwei SIMOTION Geräte, um auch das Konzept des verteilten Gleichlaufs zu demonstrieren.

Die beiden SIMOTION Geräte sind mittel PROFINET IRT synchronisiert. Nur kleine Anpassungen in HWKonfig bzw. NetPro sind nötig um auf PROFIBUS umzustellen.

Sowohl die „Master“ als auch „Slave“ Steuerung kann zur online-Demonstration verwendet werden.

Beide Steuerungen beinhalten die gleichen Achsen (2 virtuelle Achsen und 4 reale Achsen) sowie 4 SINAMICS Antriebe. Pro Steuerung sind jedoch nur 2 der 4 Antriebe aktiviert um das Projekt auf dem SIMOTION Demokoffer zu betreiben.

Hinweis Die deaktivierten Antriebe können bei Bedarf über die Expertenlistenparameter p125 und p145 aktiviert werden.

Hinweis Alle Achsen können bei Bedarf auch im Simulations-Modus, auch ohne SINAMICS Antrieb betrieben werden.

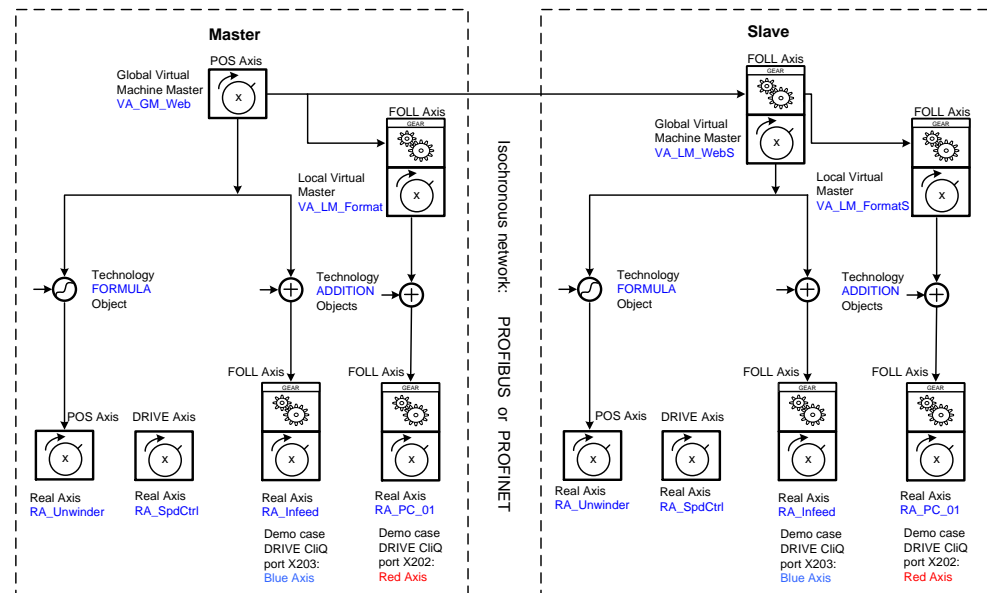
Dazu kann das Skript „AxisSimulationOnOFF“, welches sich bereits im Beispielprojekt befindet, verwendet werden.

5.1 Print Standard Beispielprojekt TO-Verschaltung

Das folgende Bild 5-1 zeigt die vorbelegten Verbindungen der TO's im Print Standard Beispielprojekt. Damit können die folgenden Szenarien demonstriert / getestet werden:

- Globaler Master Produktionsbetrieb (Stationsübergreifend mit verteiltem Gleichlauf)
- Web-basierende und Format-basierende Master Sollwert Konzepte für „format-variable“ Maschinen (z.B. Flexodruck, Tiefdruck)
- Gruppenbetrieb von Achsen mit lokalen virtuellen Mastern oder realen Mastern
- Individuelle Achs-Betriebsarten
- Konzepte für Zugregelung, Wickler und Register Regelung

Abbildung 5-1 Demo Projekt Technologie Objekt (TO) Verschaltung



Hinweis

Die Achsen „RA_Unwinder“ und „RA_SpdCtrl“ sind mit dem Skript „AxisSimulationOnOff“ in Simulation geschaltet. Das Skript ist im Beispielprojekt enthalten.

Der SINAMICS Antrieb ist in der Topologie auf „deaktiviert und nicht vorhanden“ gestellt (p125 und p145).

Sollwert Konzept

Der globale Master „VA_GM_Web“ auf der Master-Steuerung liefert in diesem Beispiel den Bahngeschwindigkeitssollwert für den kompletten Gleichlaufverbund.

Für den „VA_GM_Web“ gilt $360^\circ = 1\text{m}$. Das Referenz-Format entspricht also 1000mm.

Der Geschwindigkeitssollwert des „VA_GM_Web“ wird an der Variablen `STDcIO.IN.r64SpeedSetpoint` vorgegeben.

Achsen auf der Master-Steuerung die lediglich dem Bahnsollwert folgen und nicht winkelsynchron im Format betrieben werden, sind direkt an den „VA_GM_Web“ gekuppelt (Bsp. „RA_Infeed“). Über das Sollwertgetriebe in der `STDcIO` Schnittstelle (`StdCIO.IN.i32GearNumSetpoint/i32GearDenSetpoint`) können die Umfungsverhältnisse jeder Achse an das Referenz-Format angepasst werden.

Der „VA_LM_Format“ auf der Master-Steuerung bildet über sein Sollwertgetriebe der `STDcIO` Schnittstelle das jeweilige Format für alle formatbehafteten Achsen (Bsp. Druckzylinder „RA_PC_01“), die zueinander winkelsynchron laufen sollen.

Für alle formatabhängigen Achsen die auf den „VA_LM_Format“ gekuppelt sind bedeutet das wiederum, dass sie mit einem Sollwertgetriebe von 1:1 oder einem ganzzahligen Vielfachen zur Realisierung mehrerer Nutzen auf den lokalen Formatmaster gekuppelt werden müssen.

Um die Zahl an Controller übergreifenden Gleichlaufverbindungen möglichst gering zu halten, ist die Verteilung des Gleichlaufs auf die Slave-Steuerung mit lediglich einer Verbindung realisiert.

Dazu ist auf der Slave-Steuerung ein virtueller Master „VA_GM_WebS“ mit einer 1:1 Kopplung im Sollwertgetriebe auf den globalen Master der Master-Steuerung verschaltet.

Die übrige Sollwert-Verschaltung entspricht der der Master-Steuerung.

Hinweis

Um nur die Slave-Steuerung zu betreiben, kann die virtuelle Gleichlaufachse `VA_GM_WebS` im lokalen Mode (z.B. Betriebsart 30 – `SPEED_POS_CTRL`) betrieben werden.

ACHTUNG

Wenn der Gleichlauf an einer Folgeachse gestartet werden soll muss applikativ sichergestellt werden, dass der verteilte Gleichlauf zwischen den Controllern dafür bereit ist.

Es wird empfohlen, die Slave Achse erst zu „forcen“ wenn die Systemvariable „`distributedmotion.stateofdelayvalue`“ = valid ist.

Dies gilt bei masterseitiger Verzögerung der Ausgangswerte („VA_GM_Web“ Konfigurationsdatum `TypeOfAxis.distributedmotion.enableDelayOfCommandValueOutput = YES`)

5.2 Schritt für Schritt Demonstration des Print Standard

5.2.1 Verantwortung des Bedieners

Richtige Verwendung

Der korrekte Test dieser Applikation und der verwendeten Produkte ist nur unter Verwendung der hier beschriebenen Hardware / Firmware und Software Tools gewährleistet.

Verantwortung zur Überwachung

Die Firma oder Person, die das System betreibt ist verantwortlich regelmäßig den technischen Zustand der Applikations-Komponenten zu überwachen. Dies gilt für Schäden die von außen erkennbar sind als auch für Veränderungen im Verhalten. Der Zustand der Komponenten muss vor deren Einsatz überprüft werden und jedwede Schäden müssen vor der Nutzung behoben werden.

Qualifikation des Personals

Die Firma oder Person darf nur ausgebildetes und autorisiertes Personal einsetzen. Alle Sicherheitsvorschriften sind genau einzuhalten.

Das einzusetzende Personal muss spezielle Einweisungen in Bezug auf die möglichen Fehler/Gefahren erhalten.

5.2.2 Benötigte Hardware, Software und Projekt-Dateien

Hardware Komponenten

Diese Applikation wurde überarbeitet und getestet mit:

- SIMOTION D435-2 Trainingskoffer von SITRAIN
- TB30 (Optionskarte für zusätzliche IO's)
- SLM 5kW
- Double Motor Module 3A
- 2 Servo Motoren
 - 1FK7022-5AK71-1AG0 (2048 sin/cos incremental mit SMC20)
 - 1FK7022-5AK71-1LG0 (512 sin/cos multiturn absolute mit DRIVE-CLiQ)

Software Komponenten

Diese Applikation wurde überarbeitet und getestet mit:

- SIMOTION SCOUT V4.4 HF 4
- SINAMICS V4.7
- Technologiepaket Cam_Ext V4.4.0.4
- SIMATIC STEP 7 V5.5 SP4 HF1

Projekt-Dateien

Zum Testen / Vorführen dieser Applikation wird benötigt:

- Dieses Dokument
- Das SIMOTION Scout Print Standard Projekt

5.2.3 “Online gehen” in “8 einfachen Schritten”

Tabelle 5-1 „Online gehen” in “8 einfachen Schritten”

Schritt	Beschreibung	Bild
1	<ul style="list-style-type: none"> • Öffne SIMOTION SCOUT und dearchiviere das Projekt Print_Standard_Vx.x.x.zip” 	
2	<ul style="list-style-type: none"> • Öffne NetPro: z.B. Menü „Projekt“ -> Öffnen „NetPro“ • Doppel-Klick auf PG/PC Symbol um die Online Verbindung auszuwählen. Sicherstellen dass die richtige Ethernet Schnittstelle zugeordnet ist. • Speichern und Übersetzen der Änderungen. 	

Copyright © Siemens AG 2016 All rights reserved

5 Bedienung der Beispielapplikation

Schritt	Beschreibung	Bild																																																																								
3	<ul style="list-style-type: none"> In NetPro: Überprüfe die notwendige Ethernet IP Adresse des PC (Maus Zeiger über PG/PC Ethernet Interface) PG-Adresse z.B. = 169.254.10.11 	<p>Ethernet IE2 Industrial Ethernet</p> <p>MPI(1) MPI</p> <p>PG/PC(1)</p> <p>Interface Type: Industrial Ethernet IP address: 169.254.10.11 MAC address: 08-00-06-01-00-00</p>																																																																								
4	<ul style="list-style-type: none"> Zuweisung der IP Adresse an die Ethernet Schnittstelle des PC. z.B. "Local Area Connection" Rechtsklick -> Eigenschaften -> Eigenschaften von TCP/IP z.B. IP: 169.254.10.11 Subnet: 255.255.0.0 	<p>Netzwerkverbindungen</p> <p>Netzwerkverbindungen</p> <p>LAN oder Hochgeschwindigkeitsternet</p> <p>1394 Connection LAN oder Hochgeschw... Verbindung hergestellt... 1394 Net Adapter Wireless Network Connection LAN oder Hochgeschw... Nicht verbunden, mit Fir... Intel(R) PRO/Wireless 22... Local Area Connection LAN oder Hochgeschw... Netzwerkabel wurde en... Broadcom NetXtreme Gp...</p> <p>Eigenschaften von Local Area Connection</p> <p>Verbindung herstellen über:</p> <p>Broadcom NetXtreme Gigabit Ethernet</p> <p>Diese Verbindung verwendet folgende Elemente:</p> <p><input checked="" type="checkbox"/> PROFNET IO RT Protocol <input checked="" type="checkbox"/> SMTAC Industrial Ethernet (ISO) <input checked="" type="checkbox"/> Internet Protocol (TCP/IP)</p> <p>IP-Einstellungen können automatisch zugewiesen werden, wenn das Netzwerk diese Funktion unterstützt. Wenden Sie sich andernfalls an den Netzwerkadministrator, um die geeigneten IP-Einstellungen zu besetzen.</p> <p><input type="checkbox"/> IP-Adresse automatisch beziehen Folgende IP-Adresse verwenden:</p> <p>IP-Adresse: 169.254.10.11 Subnetzmaske: 255.255.0.0 Standardgateway:</p> <p><input type="checkbox"/> DNS-Serveradressen automatisch beziehen Bevorzugter DNS-Server: Alternativer DNS-Server:</p>																																																																								
5	<ul style="list-style-type: none"> Vor diesem Schritt muss sichergestellt werden, dass die richtigen Firmware Files verfügbar sind! "KEYS" Verzeichnis des alten Kartenabbildes auf PC Festplatte sichern Alle Dateien auf CF Karte löschen Passende Firmware Files auf die Compact Flash card kopieren "KEYS" Verzeichnis von PC zurück auf Karte kopieren 	<table border="1"> <thead> <tr> <th>Name</th> <th>Größe</th> <th>Typ</th> <th>Geändert am</th> </tr> </thead> <tbody> <tr> <td>ADDON</td> <td></td> <td>File Folder</td> <td>13.10.2008 10:41</td> </tr> <tr> <td>OEM</td> <td></td> <td>File Folder</td> <td>13.10.2008 10:41</td> </tr> <tr> <td>SIEMENS</td> <td></td> <td>File Folder</td> <td>13.10.2008 10:41</td> </tr> <tr> <td>USER</td> <td></td> <td>File Folder</td> <td>13.10.2008 10:41</td> </tr> <tr> <td>base.abb</td> <td>5.760 KB</td> <td>ABB-Datei</td> <td>07.03.2008 09:07</td> </tr> <tr> <td>CODE_0</td> <td>130 KB</td> <td>Datei</td> <td>07.03.2008 09:07</td> </tr> <tr> <td>CONTENT.TXT</td> <td>6 KB</td> <td>Textdokument</td> <td>07.03.2008 09:07</td> </tr> <tr> <td>d435_fw.bin</td> <td>2.706 KB</td> <td>BIN-Datei</td> <td>07.03.2008 09:07</td> </tr> <tr> <td>descr.inf</td> <td>1 KB</td> <td>Setup-Informationen</td> <td>07.03.2008 09:07</td> </tr> <tr> <td>LICENSEA.TXT</td> <td>15 KB</td> <td>Textdokument</td> <td>07.03.2008 09:07</td> </tr> <tr> <td>LICENSEB.TXT</td> <td>15 KB</td> <td>Textdokument</td> <td>07.03.2008 09:07</td> </tr> <tr> <td>LICENSEC.TXT</td> <td>17 KB</td> <td>Textdokument</td> <td>07.03.2008 09:07</td> </tr> <tr> <td>LICENSED.TXT</td> <td>16 KB</td> <td>Textdokument</td> <td>07.03.2008 09:07</td> </tr> <tr> <td>LICENSEE.TXT</td> <td>15 KB</td> <td>Textdokument</td> <td>07.03.2008 09:07</td> </tr> <tr> <td>READ_OSS.PDF</td> <td>697 KB</td> <td>Adobe Acrobat Doc...</td> <td>07.03.2008 09:07</td> </tr> <tr> <td>SYMTAB</td> <td>6 KB</td> <td>Datei</td> <td>07.03.2008 09:07</td> </tr> <tr> <td>toc.txt</td> <td>9 KB</td> <td>Textdokument</td> <td>07.03.2008 09:10</td> </tr> </tbody> </table>	Name	Größe	Typ	Geändert am	ADDON		File Folder	13.10.2008 10:41	OEM		File Folder	13.10.2008 10:41	SIEMENS		File Folder	13.10.2008 10:41	USER		File Folder	13.10.2008 10:41	base.abb	5.760 KB	ABB-Datei	07.03.2008 09:07	CODE_0	130 KB	Datei	07.03.2008 09:07	CONTENT.TXT	6 KB	Textdokument	07.03.2008 09:07	d435_fw.bin	2.706 KB	BIN-Datei	07.03.2008 09:07	descr.inf	1 KB	Setup-Informationen	07.03.2008 09:07	LICENSEA.TXT	15 KB	Textdokument	07.03.2008 09:07	LICENSEB.TXT	15 KB	Textdokument	07.03.2008 09:07	LICENSEC.TXT	17 KB	Textdokument	07.03.2008 09:07	LICENSED.TXT	16 KB	Textdokument	07.03.2008 09:07	LICENSEE.TXT	15 KB	Textdokument	07.03.2008 09:07	READ_OSS.PDF	697 KB	Adobe Acrobat Doc...	07.03.2008 09:07	SYMTAB	6 KB	Datei	07.03.2008 09:07	toc.txt	9 KB	Textdokument	07.03.2008 09:10
Name	Größe	Typ	Geändert am																																																																							
ADDON		File Folder	13.10.2008 10:41																																																																							
OEM		File Folder	13.10.2008 10:41																																																																							
SIEMENS		File Folder	13.10.2008 10:41																																																																							
USER		File Folder	13.10.2008 10:41																																																																							
base.abb	5.760 KB	ABB-Datei	07.03.2008 09:07																																																																							
CODE_0	130 KB	Datei	07.03.2008 09:07																																																																							
CONTENT.TXT	6 KB	Textdokument	07.03.2008 09:07																																																																							
d435_fw.bin	2.706 KB	BIN-Datei	07.03.2008 09:07																																																																							
descr.inf	1 KB	Setup-Informationen	07.03.2008 09:07																																																																							
LICENSEA.TXT	15 KB	Textdokument	07.03.2008 09:07																																																																							
LICENSEB.TXT	15 KB	Textdokument	07.03.2008 09:07																																																																							
LICENSEC.TXT	17 KB	Textdokument	07.03.2008 09:07																																																																							
LICENSED.TXT	16 KB	Textdokument	07.03.2008 09:07																																																																							
LICENSEE.TXT	15 KB	Textdokument	07.03.2008 09:07																																																																							
READ_OSS.PDF	697 KB	Adobe Acrobat Doc...	07.03.2008 09:07																																																																							
SYMTAB	6 KB	Datei	07.03.2008 09:07																																																																							
toc.txt	9 KB	Textdokument	07.03.2008 09:10																																																																							

5 Bedienung der Beispielapplikation

Schritt	Beschreibung	Bild
6	<p>In SCOUT:</p> <ul style="list-style-type: none"> Projekt „Speichern und Übersetzen“ Rechtsklick auf das SIMOTION Gerät im Projektbaum -> „Lade ins Datei System“ „Speichern normal“ und „Ziel wählen“ <p>Ziel: CF Kartenlaufwerk und das Projekt auf die Karte laden.</p>	
7	<p>IT-Diag Dateien auf CF-Karte kopieren:</p> <ul style="list-style-type: none"> IT-Diag Zip-Archiv entpacken Inhalt (Ordner „printStandard“ und Datei „printStandard.mbs“ nach CF-Karte: <i>ISIMOTIONHMIFiles</i> kopieren. 	
8	<ul style="list-style-type: none"> Ethernetkabel zwischen PC und PROFINET-Switch (X150) der SIMOTION verbinden. Die CF Karte in den Demo Koffer stecken und einschalten. (Nach Hochlauf kann evtl durch automatischem Firmware Hochrüstung ein weiterer Spannung Aus/Ein notwendig sein) GO ONLINE mit dem Button in der Menüleiste oder Rechtsklick auf das SIMOTION Gerät im Projektbaum -> „Mit Zielsystem verbinden“ 	

5.2.4 Den Motor zum Drehen bringen in "3 einfachen Schritten"

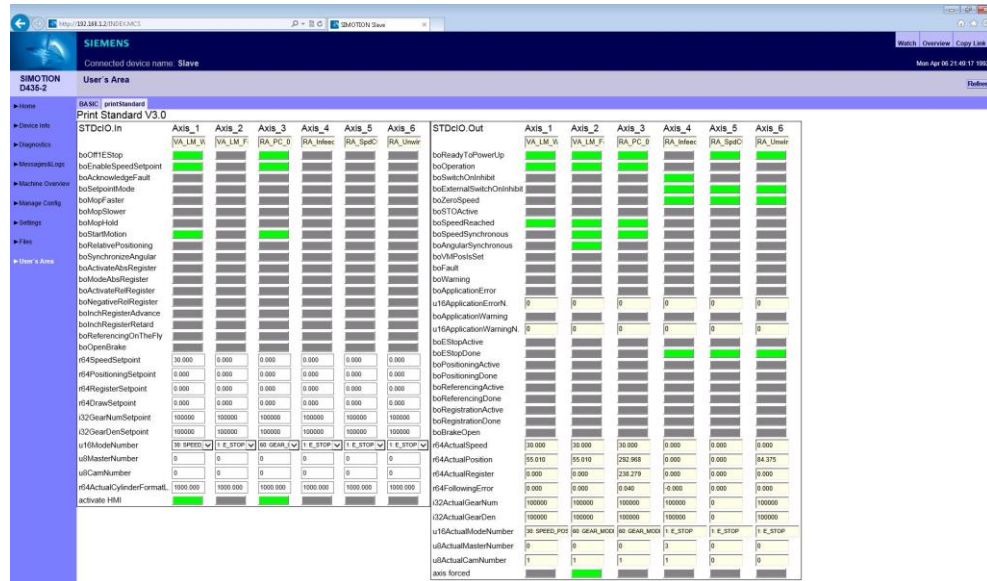
Voraussetzung sind die vollständig durchgeführten Schritte aus Kapitel [5.2.3 "Online gehen" in "8 einfachen Schritten"](#).

Tabelle 5-2 Den Motor zum drehen bringen in "3 einfachen Schritten"

Schritt	Beschreibung	Bild
1	Sicherstellen, dass die Print Standard IT-Diag Dateien auf die CF-Karte kopiert wurden. (siehe Kapitel 5.2.3)	
2	SIMOTION IT-Diag Seite öffnen: <ul style="list-style-type: none"> Dazu im Browser (z.B. Internet Explorer) in der Adresszeile die IP-Adresse des SIMOTION Gerätes eingeben Auf die Print Standard IT-Diag Seite wechseln 	
3	<ul style="list-style-type: none"> Gewünschte Achsen für die Bedienung über IT-Diag/HMI mit dem Button „activate HMI“ freischalten Folgende Eingänge der STDClO.IN Struktur einer beliebigen Achsen ansteuern: <ul style="list-style-type: none"> - boOff1EStop = TRUE - boEnableSpeedSetpoint = TRUE - r64SpeedSetpoint <> 0.0 - u16ModeNumber = 30 - boStartMotion = TRUE => Der Motor dreht !	

5.2.5 Hinweise zum Verwenden der IT-Diag Seite

Abbildung 5-2 Print Standard IT-Diag Seite



Die IT-Diag Seite verbindet sich zur STDCIO Schnittstelle von 6 oder mehr generischen Achsen (diese sind frei konfigurierbar). Dadurch wird eine komfortablere (graphische) und übersichtlichere Möglichkeit geschaffen die komplette STD Schnittstelle der Achsen zu testen.

Die folgenden Programme geben IT-Diag Zugriff auf die STDCIO Schnittstelle:

- dHMI:**

Erzeugt Kopien der STDCIO Schnittstellen Variablen Struktur nur für den HMI Zugriff

Die Anzahl der Achsen die über IT-Diag gesteuert werden sollen, kann über die globale Variable „gu8NumberOfAxis“ festgelegt werden.

Ebenso kann ein String zum Anzeigen eines Achs-Namens auf IT-Diag festgelegt werden.
- pMcomm:**

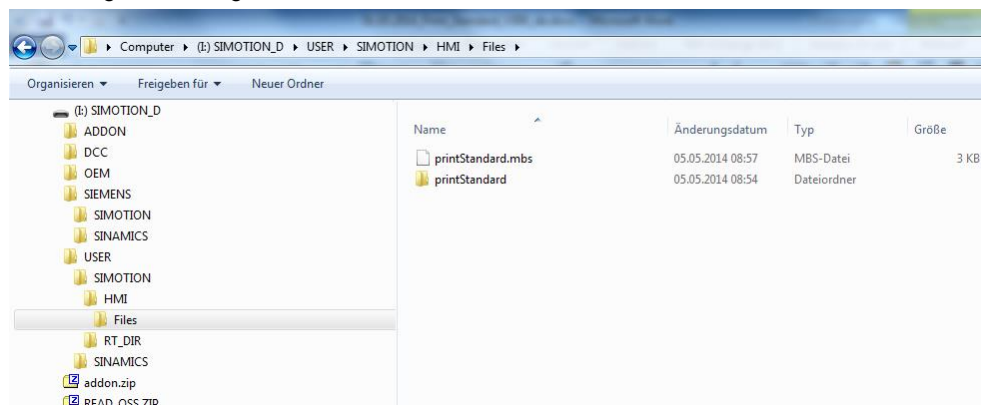
Der Funktionsbaustein “ FBLPrint_AxisCommunication” stellt die Verbindung zwischen HMI STDCIO und Print Standard STDCIO Schnittstelle her.

Über die globale Variable „HMI_axis_<number>_stdcIOActivateTestIO“ wird am „FBLPrint_AxisCommunication“ der Test-Mode aktiviert.

Kopieren der IT-Diag Dateien auf die CF-Karte

- IT-Diag Zip-Archiv entpacken
- Den Ordner „printStandard“ und die Datei „printStandard.mbs“ auf die SIMOTION CF-Karte nach x:\User\SIMOTION\HMI\Files kopieren.

Abbildung 5-3 IT-Diag Dateien auf SIMOTION CF-Karte



Handling der IT-Diag Seite

Um eine Achse über die STDcIO.IN Schnittstelle zu bedienen, muss der Button „activate HMI“ aktiviert werden.

Die Ausgangswerte der STDcIO.OUT Schnittstelle werden immer angezeigt (unabhängig ob „activate HMI“ gesetzt ist oder nicht).

Die Ausgabe „axis forced“ ist nicht Teil der STDc Schnittstelle. Sie zeigt an, ob die Achse „geforced“ (intern gekoppelt) ist oder nicht.

Im Beispielprojekt wird die virtuelle Format-Achse („VA_LM_Format“ bzw. „VA_LM_FormatS“) automatisch „geforced“, wenn „activate HMI“ nicht gesetzt ist.

Die Ausgaben „u16ApplicationWarningN.“ und „u16ApplicationErrorN.“ zeigen die aktuelle Warnungs- bzw. Fehlernummer an.

Bewegt man den Mauszeiger auf das Ausgabefeld, wird der entsprechende Fehlertext als Tooltip angezeigt. (Die Fehlertexte sind in den IT-Diag Dateien auf der CF-Karte hinterlegt.)

5.2.6 Demonstration einiger Betriebsarten

Mit den Schritten der vorherigen Kapitel ist der Print Standard funktionsfähig d.h. online Verbindung ist hergestellt und eine Achse/Antrieb kann gedreht werden. Im nächsten Schritt können einige typische Achs Betriebsarten getestet werden.

Mode 60 – GEAR_MODE: Beispiel “Gruppierung von realen Achsen”

Demonstration eines Gleichlaufbetriebs zu einem realen Master.

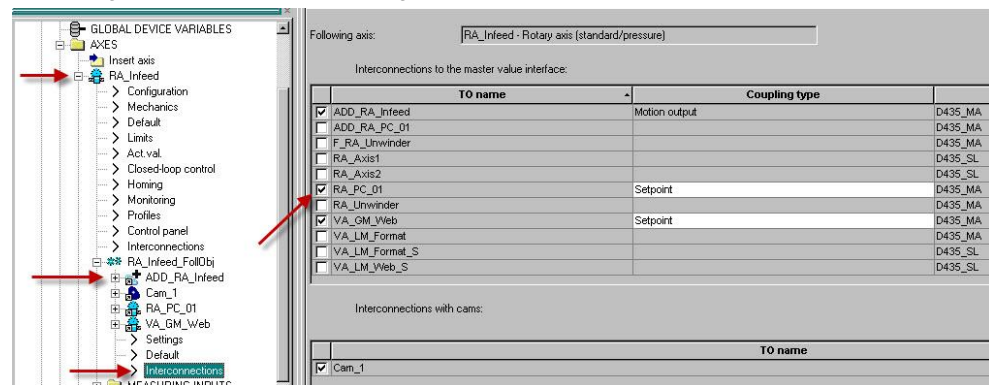
Rote Achse (Gruppen Master) im lokal Betrieb (Drehzahlvorgabe) im Mode 30.

Blaue Achse (Gruppen Slave) folgt / Synchronisiert der roten Achse im Mode 60

„Mögliche Master“ Einstellungen der Achse „RA Infeed“ (blaue Achse):

- Sicherstellen, dass am “RA_Infeed_Following Object” die Achse “RA_PC_01” als möglicher Master angewählt ist.

Abbildung 5-4 Gleichlaufverschaltungen



- Überprüfen, dass in “AxisTOConfigData.sUser.atoPossibleMasters” die Achse „RA_PC_01“ als möglicher Master Nummer 2 konfiguriert und aktiviert ist.

Abbildung 5-5 Masterzuordnung

D435_MA.M_Var_GI:				
	Name	Datentyp	Statuswert	Anzeigeformat
1	web_axes_possible_masters	Array		
2	-web_axes_possible_masters[1]	ANYOBJECT	_to.VA_GM_Web	<input type="checkbox"/>
3	-web_axes_possible_masters[2]	ANYOBJECT	_to.RA_PC_01	<input checked="" type="checkbox"/>
4	-web_axes_possible_masters[3]	ANYOBJECT	TO#NIL	<input type="checkbox"/>
5	-web_axes_possible_masters[4]	ANYOBJECT	TO#NIL	<input type="checkbox"/>
6	-web_axes_possible_masters[5]	ANYOBJECT	TO#NIL	<input type="checkbox"/>

Einstellungen in der STDcIO Schnittstelle der Achse „RA Infeed“ (blaue Achse):

- u8MasterNumber := 2
- u16ModeNumber := 60
- boOff1Estop := TRUE
- boStartMotion := TRUE

Einstellungen in der STDclIO Schnittstelle der Achse „RA PC 01“ (rote Achse):

- u16ModeNumber := 30
- boEnableSpeedSetpoint := TRUE
- r64SpeedSetpoint := (ungleich 0)
- boOff1Estop := TRUE
- boStartMotion := TRUE

➔ Beide Achsen drehen zusammen.

Wenn die Drehzahl der Roten Achse verändert wird (da Realer Master mit Sollwertkopplung) ändert sich auch die Drehzahl der Blauen Achse.

Mode 60 – GEAR_MODE: Beispiel “Folgebetrieb am globalen Maschinen Master”

Demonstration eines Gleichlaufbetriebs zu einem virtuellen Master.

VA_GM_Web (globaler Master) im lokal Betrieb (Drehzahlvorgabe) im Mode 30.

Rote Achse folgt / synchronisiert (über andere TO's) zum Globalen Master im Mode 60.

Synchronisation und Registerverstellung können im Anschluss ebenfalls getestet werden.

Einstellungen in der STDclIO Schnittstelle der Achse „RA PC 01“ (rote Achse):

ACHTUNG	Stellen Sie sicher, dass ein Masterobjekt verschaltet ist! u8MasterNumber = 0 = default master = Addierobjekt von “RA_PC_01” Der aktuelle Master des Addierobjekts muss die Achse “VA_LM_Format” sein! Überprüfen Sie wie im vorherigen Beispiel in AxisTOConfigData, dass“ADD_RA_PC_01” und “VA_LM_Format” als „current master“ aktiviert sind.
----------------	--

- u16ModeNumber := 60
- boOff1Estop := TRUE
- boStartMotion := TRUE

➔ Die Achse ist jetzt im Gleichlaufbetrieb.

Einstellungen in der STDcIO Schnittstelle der virtuellen Achse „VA GM Web“:

- u16ModeNumber := 30
- boEnableSpeedSetpoint := TRUE
- r64SpeedSetpoint := (ungleich 0)
- boOff1Estop := TRUE
- boStartMotion := TRUE

➔ Die virtuelle Achse “bewegt” sich. Die rote Achse folgt der Bewegung.

Nun können die verschiedenen Synchronisation und Register Methoden getestet werden:

- Master Drehzahl Null setzen (zur besseren visuellen Beobachtung des Verhaltens an der roten Achse)
- STDcIO Bit “boSynchronizeAngular” aktivieren
- 90° in “r64RegisterSetpoint” eingeben
- Register Bewegung mit “boActivateAbsRegister” starten

➔ Verhalten in “STDcIO.OUT.r64ActualRegister” beobachten.

Mode 20 - POSITIONING: Beispiel des lokalen Positionier Betriebs

Demonstration/Test von Positionier Bewegungen in der Betriebsart „POSITIONING“.

Einstellungen in der STDcIO Schnittstelle der Achse „RA PC 01“ (rote Achse):

- u16ModeNumber := 20
- boEnableSpeedSetpoint := TRUE
- r64SpeedSetpoint := (ungleich 0)
- boOff1Estop := TRUE

➔ Die Achse ist jetzt im Positionier-Modus.

Test Absolute Positionierung:

- boRelativePositioning := FALSE
- Positionssollwert “r64PositioningSetpoint” ändern (z.B. 90° und dann 180°)
- boStartMotion := TRUE

➔ Die Achse positioniert auf die eingegebene Position.

Test Relative Positionierung:

- boStartMotion := FALSE
- boRelativePositioning := TRUE
- Positionssollwert "r64PositioningSetpoint" ändern (z.B. 90° und dann 180°)
- boStartMotion := TRUE

➔ Die Achse positioniert um die eingegebene Position (Distanz)

6 Applikations-Fehler- und Warnungen

Generell wird zwischen Applikationsfehler und Applikationswarnungen unterschieden.

Diese werden in der STDcIO Schnittstelle über die Variablen...

- boApplicationError
- u16ApplicationErrorNumber
- boApplicationWarning
- u16ApplicationWarningNumber

...angezeigt.

Die STDcIO.OUT Variablen boFault und boWarning zeigen Antriebs- bzw. TO Fehler und Warnungen des Systems.

Applikationsfehler

Ein Fehler führt zu einer Reaktion an der Achse. Das jeweilige Kommando wird nicht ausgeführt und die Achse wird gestoppt um Schaden zu verhindern.

Abhängig von der Fehler-Reaktion, die über die Konfigurations-Variable AxisConfigData.eErrorReaction für jede Achse separat eingestellt werden kann, unterscheidet sich die Art des Stopps:

ERROR_E_STOP:

Die Achse verzögert mit der E-Stop Rampe, die in „axisConfigData.sEStopDyn“ parametrisiert werden kann.

ERROR_COAST_STOP:

The axis will coast to stop (disable axis, AUS2).

Applikationswarnungen

Eine Warnung führt nicht zu einer Fehler-Reaktion an einer Achse. Der jeweilige Befehl wird einfach nicht ausgeführt. Die aktive Bewegung wird fortgesetzt.

Quittieren von Fehlern und Alarmen

Applikations-Fehler müssen nach beheben der Ursache quittiert werden. Das Quittieren von Fehlern muss über das STDcIO.IN Bit „boAcknowledgeFault“ erfolgen. Quittieren von Print Standard Fehlern ist nicht durch Fehler quittieren im SCOUT Alarmfenster möglich!

Hinweis

Zum Quittieren von Fehler muss sich die entsprechende Achse im Mode E_STOP befinden.

Applikations-Warnungen müssen nicht quittiert werden. Wurde die Ursache der Warnung eliminiert, werden die Bits „boApplicationWarning“ und „u16ApplicationWarningNumber“ zurückgesetzt.

Meldungshandling (Fehler-Historie)

Im Fehlerfall ist die Struktur „axisData“ als Diagnose-Struktur geeignet.

Hier sind die wichtigsten Aktual-Parameter zusammengefasst, die Ursächlich für Fehler oder Warnungen sein können.

Außerdem sind die Steuer- und Zustandswörter 1 und 2 des entsprechenden Antriebs sowie das Safety-Zustandswort (wenn Safety im Antrieb aktiv) vorhanden.

Tieferegehende Fehlerdiagnose bieten die zwei Unterstrukturen „sStateLog“ und „sAppMsgLog“:

sAxisSafetyState1:

Die Struktur wird nur ab Simotion V4.4 befüllt.

Dazu muss die Anweisung „#define LPRINT_USE_SAFETY_CHANNEL“ in der Bibliotheks-Unit „cVarConf“ einkommentiert sein. Weiterhin dürfen unter „Bibliothek Eigenschaften / TPs/TOs“ der LPrint nur Geräte mit V4.4 ausgewählt sein.

Für die Auswertung des Safety-Zustandsworts (Safety Funktionalität über DSDB) muss am TO unter Konfiguration / Funktionen / SINAMICS SAFETY INTEGRATED „Standard“ ausgewählt sein.

sStateLog:

Hier wird mitgeschrieben, wann (Zeitstempel) von einem in einen anderen Mode (Betriebsart) gewechselt wird. Mode-intern werden ebenfalls State-Wechsel erfasst.

Auf diese Weise kann nachvollzogen werden, von welchem Mode bzw. State in den Mode ERROR gewechselt wurde.

Der Beiwert „i32CmdState“ gibt Auskunft über Rückgabewerte von Systemfunktionen bzw. abortID's von Befehlen, die mit commandID's überwacht werden.

Der Beiwert „i32AddValue“ kann vom Benutzer verwendet werden, um „Marker“ zur genauen Bestimmung der Fehler-Stelle in den Code einzubringen

sAppMsgLog:

Diese Unterstruktur dient als Fehler-Historie. Hier werden alle Print Standard Applikations-Fehler und –Warnungen mit Zeitstempel eingetragen.

Hinweis

Ein Meldehandling zur Anzeige von SIMOTION und SINAMICS Fehlern in der STDcIO Schnittstelle ist im Print Standard nicht integriert.

Eine Lösung um sowohl System-Fehler als auch anwenderdefinierte Fehler (z.B. Print Standard Fehler) zentral zu sammeln und anzuzeigen bietet die Applikation SIMOTION Meldungs-Handling, welche über SIOS erhältlich ist.

Um die Handhabung zu vereinfachen sind Warnungen und Fehler in getrennte Zahlenbereiche eingeteilt:

- Warnungen: 0001 – 499
- Initialisierungsfehler 0500 – 999
- Fehler: 1000 – 2000

Diese sind in den folgenden Tabellen aufgeführt.

6.1 Print Standard Applikations-Warnungen

Tabelle 6-1 Applikationswarnungen

Nr.	Name	Beschreibung	Abhilfe
100	LPRINT_WAR_INVALID_GEAR_FACTOR	Getriebe Zähler oder Nenner ist Null	Zähler und Nenner ungleich Null vorgeben. STDcIO.IN.r32GerNumSetpoint STDcIO.IN.r32GerDenSetpoint
101	LPRINT_WAR_POS_NO_SPEED_SETPOINT	Positionieren angewählt und kein Geschwindigkeits-sollwert vorgegeben	STDcIO.IN.r64SpeedSetpoint <> 0 vorgeben
102	LPRINT_WAR_REF_ON_THE_FLY_NO_MI	Fliegendes referenzieren angewählt aber kein Messtaster angelegt oder verschaltet	Messtaster für die Achse konfigurieren und in der Struktur „AxisTOConfigData.sUser.toMeasuringInput“ zuordnen
103	LPRINT_WAR_INVALID_MODE_SELECTED	Gewählte Betriebsart für diesen Achstyp nicht verfügbar oder Betriebsart nicht vorhanden.	Gültige Betriebsart vorgeben.
104	LPRINT_WAR_REF_ON_THE_FLY_AXIS_NOT_REFERENCED	Die Achse muss referenziert sein, bevor das Speichern der Referenzposition angewählt werden kann	Passives Referenzieren durchführen Siehe auch unter Mode 10 - REFERENCING (Speichern der Inkrementalgeberposition)
105	LPRINT_WAR_NO_CYCLIC_INTERFACE	Zyklische Antriebschnittstelle nicht aktiv	Es liegt ein Antriebsfehler vor. U.u. ist SINAMICS noch nicht vollständig hochgelaufen.
106	LPRINT_WAR_HW_ESTOP_ACTIVE	Kein 24V Signal (TRUE Signal) am Eingang "HWOFF1Estop" des Funktionsbausteins „FBLPrint_StdAxisBackgrnd“	24V Signal an der Hardware-Klemme überprüfen. Verbindung von SIMOTION Eingang zum Eingang "HWOFF1Estop" des Funktionsbausteins "FBLPrint_StdAxisBackgrnd" überprüfen.
107	LPRINT_MAX_MEAS_REV_REACHED	Warnung der Funktion Homing active: Die Anzahl der maximalen Umdrehungen axisConfigData.sHoming.u16MeasInMaxNumOfRev wurde erreicht.	Verwendeten Messtaster kontrollieren.
108	LPRINT_WAR_REF_INVALID_ENCODER_TYPE	Ausfallsicheres Speichern der Position nach passivem Referenzieren nicht möglich mit Absolutwertgeber.	Funktionalität mit Absolutwertgeber nicht starten. Speichern der Position bei Verwendung eines Absolutwertgebers nicht notwendig.
109	LPRINT_WAR_REL_ABS_REGS_ACTIVE	STDcIO.In.boActivateAbsRegister und STDcIO.In.boActivateRelRegister sind gleichzeitig angewählt	Nur eines der beiden bits anwählen.

6 Applikations-Fehler- und Warnungen

Nr.	Name	Beschreibung	Abhilfe
110	LPRINT_WAR_ABS_REGISTER_CYLINDER_NOT_ANGULAR_SYNCHRONOUS	Zum Zeitpunkt des Startens einer Absoluten Registerverstellung (boActivateAbsRegister) muss die Achse Winkelsynchron sein.	Mit einer steigenden Flanke am Bit boSynchronizeAngular kann der Zylinder absolut synchronisiert werden.
111	LPRINT_WAR_OFF1_ESTOP_CYCLIC_INTERFACE_WAS_INACTIVE	Es wurde versucht die Achse über das Bit boOff1Estop einzuschalten während die zyklische Kommunikation zum Antrieb (cyclicInterface) INAKTIV war.	Die Warnung verschwindet nach einer fallenden Flanke am bit boOff1Estop. Mit der nächsten steigenden Flanke (während cyclicInterface = ACTIVE) kann die Achse eingeschaltet werden.
112	LPRINT_WAR_REF_ON_THE_FLY_ENC_OUTPUT_NOT_POSSIBLE	Referenziermode REF_ON_THE_FLY nur mit TM41 Achsen (Achstyp ENCODER_OUTPUT) nicht möglich.	/
113	LPRINT_WAR_TORQUE_FUNCTIONS_NOT_POSSIBLE	Drehmoment-Funktionen nur mit realen Achsen möglich.	/
114	LPRINT_WAR_SYNCH_AXIS_CYCLE_IS_ZERO	Der Konfigurationsparameter axisConfigData.sSyncConfig.r64SynchAxisCycle ist 0.0. Als Ersatzwert wird die Modulolänge der Achse verwendet.	Konfigurationsparameter axisConfigData.sSyncConfig.r64SynchAxisCycle überprüfen.
115	LPRINT_WAR_SYNCH_AXIS_CYCLE_WAS_SET_TO_ZERO	Der intern verwendete Parameter axisData.r64ActiveSynchAxisCycle wurde automatisch auf den Wert 0.0 gesetzt, da die Modulolänge 0.0 ist.	Modulolängen-Einstellung der Achse überprüfen. Die Modulolänge wird aus dem Achs TO ausgelesen und wird im Parameter axisTOConfigData.sInternal.r64ModuloLength gespeichert.
150	LPRINT_WAR_COMMAND_EXECUTION_ERROR	Die Warnung wird ausgegeben, wenn Mode-intern (z.B. REFERENCING, ...) aufgrund eines Fehlers während der Befehlsausführung in den Error-Case (interner Fehler-Case) gewechselt wurde.	Für detaillierte Informationen siehe Struktur „axisData.sStateLog“. Dort wird im Beiwert „i32CmdState“ z.B. der Rückgabewert von Systemfunktionen, oder die abortID von Befehlen die mit command ID's überwacht werden, ausgegeben.

6.2 Print Standard Initialisierungsfehler

Diese Fehler können nur durch einen Neustart oder das erneute Durchlaufen der Initialisierung (axisConfigData.boRestartInitialization = TRUE) behoben werden.

Tabelle 6-2 Initialisierungsfehler

Nr.	Name	Beschreibung	Abhilfe
500	LPRINT_INITERR_INVALID_TO_REFERENCE	Es wurde keine Achs-Referenz zugeordnet.	„AxisTOConfigData.sUser.toAxis“ zuordnen
501	LPRINT_INITERR_NO_AXIS_ASSIGNED	Kein Achs-Objekt konfiguriert.	Achs-Konfiguration in AxisTOConfigData.sUser.toAxis überprüfen
502	LPRINT_INITERR_NO_FOLL_OBJ_ASSIGNED	axisToConfigData.sInternal.toFollowingObject wurde nicht korrekt zugewiesen.	Axis.synchMonitoring.activeFollowingObject (Expertenliste der Achse) überprüfen
503	LPRINT_INITERR_DEFAULT_VELOCITY_IS_ZERO	Die Defaultvelocity ist Null.	Wert größer Null eintragen in Axis.userDefaultDynamics.velocity
504	LPRINT_INITERR_UNSUPPORTED_ENCODER_TYPE	Es wurde kein gültiger Geber erkannt.	Gültige Geber sind: Inkrementalgeber, Absolutwertgeber; Axis.activeconfigdata.TypeOfAxis.NumberOfEncoders.Encoder_1.encoderType überprüfen
505	LPRINT_INITERR_NOMINAL_TORQUE_IS_ZERO	Im Konfigurationsdatum der Achse (nominalTorque) wurde der Wert 0.0 ausgelesen.	Achs-Konfigurationsdatum TypeOfAxis.setpointDriverInfo.DriveData.nominalTorque überprüfen.
506	LPRINT_INITERR_MAX_TORQUE_IS_ZERO	Im Konfigurationsdatum der Achse (maxTorque) wurde der Wert 0.0 ausgelesen.	Achs-Konfigurationsdatum TypeOfAxis.setpointDriverInfo.DriveData.maxTorqueüberprüfen.

6.3 Print Standard Applikations-Fehler

Tabelle 6-3 Applikationsfehler

Nr.	Name	Beschreibung	Abhilfe
1000	LPRINT_ERR_DRIVE_AXIS_INVALID_MODE	Für eine Drehzahlachse wurde eine Betriebsart angefordert, die mindestens Positionier-Funktionalität benötigt (z.B. POSITIONING)	Geeignete Betriebsart anwählen (siehe DRIVE_AXIS_MODE_ARRAY); Achse mit ausreichender Technologie wählen; Ggf. DRIVE_AXIS_MODE_ARRAY anpassen (nicht empfohlen)

6 Applikations-Fehler- und Warnungen

Nr.	Name	Beschreibung	Abhilfe
1001	LPRINT_ERR_POS_AXIS_INVALID_MODE	Für eine Positionierachse wurde eine Betriebsart angefordert, die mindestens Gleichlauf-Funktionalität benötigt (z.B. GEAR_MODE)	Geeignete Betriebsart anwählen (siehe POSITIONING_AXIS_MODE_ARRAY); Andere Achse mit ausreichender Technologie wählen; Ggf. POSITIONING_AXIS_MODE_ARRAY anpassen (nicht empfohlen)
1002	LPRINT_ERR_FOL_AXIS_INVALID_MODE	Für eine Gleichlaufachse wurde eine Betriebsart angefordert, die nicht im FOLLOWING_AXIS_MODE_ARRAY hinterlegt ist.	Geeignete Betriebsart anwählen (siehe FOLLOWING_AXIS_MODE_ARRAY); Ggf. FOLLOWING_AXIS_MODE_ARRAY anpassen
1004	LPRINT_ERR_FORCE_AXIS_INVALID_AXIS	Die Achse die geforced werden soll ist keine virtuelle Folge-Achse.	Es kann nur eine virtuelle Gleichlaufachse geforced werden.
1011	LPRINT_ERR_ASSIGNED_MASTER	Folgeachse hat sich selbst als Master-Objekt	Master Array Vorbelegung in pMStarup überprüfen; Angewählte Master Nummer in STDcIO.IN.u8MasterNumber prüfen
1012	LPRINT_ERR_NO_DEFAULT_MASTER_ASSIGNED	Einer Folgeachse wurde kein default Master zugewiesen	Default Master in axisTOConfigData.sUser.toDefaultMaster prüfen
1013	LPRINT_ERR_SELECTED_MASTER_NO	Master Nummer grosser als Matser Array	Master Nummer (STDcIO.IN.u8MasterNumber) <= SIZE_OF_MASTER_ARRAY (cVarConf) vorgeben
1015	LPRINT_ERR_NO_MASTER_ASSIGNED	Einer Folgeachse wurde kein Master zugewiesen	Master Array Vorbelegung in pMStarup überprüfen; Master Nummer (STDcIO.IN.u8MasterNumber) mit gültigem Master in Master Array wählen
1016	LPRINT_ERR_ASSIGNED_DEFAULT_MASTER	Folgeachse hat sich selbst als default Master	Default Master in axisTOConfigData.sUser.toDefaultMaster ändern
1022	LPRINT_ERR_REF_NO_SET_AXIS_ASSIGNED	Referenzier-Modus „SET_AXIS_POSITION“ gewählt, aber keine set axis angegeben.	axisTOConfigData.sUser.toSetAxis angeben.
1023	LPRINT_ERR_REF_NO_SET_FOLL_OBJ_ASSIGNED	Folgeobjekt für Referenzier-Modus „SET_AXIS_POSITION“ wurde nicht korrekt gesetzt.	Axis.synchMonitoring.activeFollowingObject (Expertenliste der Achse) überprüfen und Config-Data Initialisierung neu starten: axisConfigData.boRestartInitialization
1025	LPRINT_ERR_REF_NO_VIRTUAL_AXIS	Referenzier-Modus für virtuelle Achsen („SET_AXIS_POSITION“) für eine reale Achse angewählt.	Referenziermodus für reale Achsen wählen.

6 Applikations-Fehler- und Warnungen

Nr.	Name	Beschreibung	Abhilfe
1026	LPRINT_ERR_REF_NO_MI_ASSIGNED	Kein Messtaster für Referenziermodi „ACTIVE“ oder „REF_ON_THE_FLY“ mit Absolutwertgeber verschaltet.	Messtaster in axisTOConfigData.sUser.toMeasuringInput verschalten.
1027	LPRINT_ERR_REF_NO_OR_INVALID_MODE_SELECTED	Kein bzw. ungültiger Referenzier-Modus angewählt.	Gültigen Referenzier-Modus in axisConfigData.sHoming.eReferencingMode vorgeben
1028	LPRINT_ERR_REF_NO_REAL_AXIS	Referenzier-Modus für reale Achsen für eine virtuelle Achse angewählt.	Referenziermodus für virtuelle Achsen wählen.
1032	LPRINT_ERR_NO_DEFAULT_CAM_ASSIGNED	Einer Folgeachse wurde keine default Cam zugewiesen	Default Cam in axisTOConfigData.sUser.toDefaultCam prüfen
1033	LPRINT_ERR_SELECTED_CAM_NO	Cam Nummer größer als Cam Array	Cam Nummer (STDcIO.IN.u8CamNumber) <= SIZE_OF_CAM_ARRAY (cVarConf) vorgeben
1035	LPRINT_ERR_NO_CAM_ASSIGNED	Einer Folgeachse wurde kein Kurvenscheibe zugewiesen	Cam Array Vorbelegung in pMStarup überprüfen; Cam Nummer mit gültiger Kurvenscheibe in Cam Array wählen
1040	LPRINT_ERR_CYLINDER_LENGTH_IS_ZERO	Aktuelles Zylinderformat ist 0; Ersatzwert für axisData.r64CylinderFormatLengthRatio ist 1.0	STDcIO.IN.r64ActualCylinderFormatLength überprüfen; Wert ungleich Null schicken
1041	LPRINT_ERR_CYLINDER_LENGTH_LOWER_LIMIT	Aktuelles Zylinderformat ist kleiner als Grenze; axisData.r64CylinderFormatLengthRatio wird mit minimaler Zylinderformatlänge berechnet	Grenze prüfen in axisConfigData.sCylinderFormat.r64MinLength; Wert innerhalb Grenzen schicken
1042	LPRINT_ERR_CYLINDER_LENGTH_UPPER_LIMIT	Aktuelles Zylinderformat ist größer als Grenze; axisData.r64CylinderFormatLengthRatio wird mit maximaler Zylinderformatlänge berechnet	Grenze prüfen in axisConfigData.sCylinderFormat.r64MaxLength; Wert innerhalb Grenzen schicken
1043	LPRINT_ERR_CYLINDER_LENGTH_LIMIT_IS_ZERO	Zylinderformat Grenzen (Min oder Max) <= 0; Ersatzwert für axisData.r64CylinderFormatLengthRatio ist 1.0	Grenze prüfen in axisConfigData.sCylinderFormat.r64MaxLength; und axisConfigData.sCylinderFormat.r64MinLength;
1044	LPRINT_ERR_CYLINDER_LENGTH_LIMITS	Zylinderformat Grenze Min ist größer als Max; Ersatzwert für axisData.r64CylinderFormatLengthRatio ist 1.0	Grenze prüfen in axisConfigData.sCylinderFormat.r64MaxLength; und axisConfigData.sCylinderFormat.r64MinLength;
1045	LPRINT_ERR_CYLINDER_REF_LENGTH_IS_ZERO	Zylinderformat Referenzwert ist 0.0 Ersatzwert für axisData.r64CylinderFormatLengthRatio ist 1.0	Konfiguration in: axisConfigData.sCylinderFormat.r64ReferenceLength überprüfen

6 Applikations-Fehler- und Warnungen

Nr.	Name	Beschreibung	Abhilfe
1203	LPRINT_ERR_WINDING_NO_FORMULAR_OBJ	Master Objekt muss ein Formel-Objekt sein für folgende Betriebsarten: WINDING SPLICE_TO_WINDING SPLICE_TO_STOP	Formelobjekt als Master vorgeben. z.B. an axisTOConfigData.sUser.toDefaultMaster
1221	LPRINT_ERR_DEFAULT_CHANNEL_FORMULAR_OBJ	Es wurde ein ungültiger MotionIn Eingang des Formelobjekts gewählt.	Überprüfen Sie den Eingang „miDefaultChannel“ am „FBLPrint_TechAxis“: Verwenden Sie einen anderen MotionIn Eingang (1-3) am Formelobjekt.
1222	LPRINT_ERR_FBTECH_NOLOGY_FUNCTIONS_WRONG_TASK	Die aufrufende Task des Funktionsbausteins „FBLPrint_TechAxis“ stimmt nicht mit dem gewählten Taskmode überein.	Überprüfen Sie den Funktionsbaustein-Eingang „taskMode“ des „FBLPrint_TechAxis“ und die Task im Ablaufsystem in der der Baustein aufgerufen wird.

7 Abkürzungen

Tabelle 7-1 Abkürzungen

TIA	Totally Integrated Automation Konzept der Siemens Hardware/Software Produkte
TO	SIMOTION Technologie Objekte: Modulare Motion Control Firmware Software Objekte mit verschiedenen Funtionalitäten
RA	Real Axis : Motion control Achse mit Verbindung zu einem Antrieb.
VA	Virtual Axis : Motion control Achse ohne physikalischen Antrieb.
LM	Local Master: Virtuelle Achse für eine Gruppe Realer Achsen.
GM	Global Master: Virtuelle Achse für alle Achsen in der Maschine
PROFINET IRT	Motion Control Synchronisation implementiert mit PROFINET Kommunikation
MCC	Motion Control Chart : Eine der (graphischen) Programmiersprachen von SIMOTION
ST	Structured Text: Die Basis Programmiersprache von SIMOTION.
MOP	Motorized Potentiometer: (Software)-Funktion zum Verändern eines (Drehzahl) Sollwertes mittels schneller/langsamer/halt Bit Kommandos

8 Literaturhinweise

Tabelle 8-1

	Themengebiet
\1\	Siemens Industry Online Support http://support.automation.siemens.com
\2\	Downloadseite des Beitrages https://support.industry.siemens.com/cs/ww/de/view/38195516
\3\	Converting Bibliothek http://support.automation.siemens.com/WW/view/de/48805235
\4\	SIMOTION, Motion Control, TO Achse elektrisch/hydraulisch, Externer Geber http://support.automation.siemens.com/WW/view/de/61055938
\5\	SIMOTION Runtime Basisfunktionen http://support.automation.siemens.com/WW/view/de/60744985
\6\	SIMOTION SCOUT Ergänzende Technologieobjekte http://support.automation.siemens.com/WW/view/de/61056839
\7\	SIMOTION. Motion Control, TO Gleichlauf, Kurvenscheibe http://support.automation.siemens.com/WW/view/de/61055960
\8\	SIMOTION, Motion Control, TO Nocken und Messtaster http://support.automation.siemens.com/WW/view/de/61055977
\9\	SINAMICS S120 Funktionshandbuch http://support.automation.siemens.com/WW/view/de/68042590
\10\	SINAMICS S120 Funktionshandbuch Safety http://support.automation.siemens.com/WW/view/de/64915128
\11\	SINAMICS S120/S150 Listenhandbuch http://support.automation.siemens.com/WW/view/de/68041075

9 Ansprechpartner

Siemens AG
 Digital Factory
 DF FA PMA APC
 Frauenaauracher Straße 80
 D - 91056 Erlangen
 mailto: tech.team.motioncontrol@siemens.com

10 Historie

Tabelle 10-1 Historie

Version	Datum	Änderung
V2.2.0.0	01/2009	Erstmalige deutsche Übersetzung der englischen Version Print Standard_Docu_V22e.pdf
V2.2.1.0	03/2010	- Konvertierung auf neue Formatvorlage - Änderungen für neue Softwareversion eingepflegt
V3.0.0	08/2014	- Komplettüberarbeitung (siehe Changelog)
V3.1.0	05/2015	Ergänzungen (siehe Changelog)
V3.1.2	09/2015	Ergänzungen (siehe Changelog)
V3.1.3	09/2015	Ergänzungen (siehe Changelog)
V3.2.0	01/2016	Fehlerbehebungen und Erweiterungen; Neues Kapitel 4.14 - Verteilter Gleichlauf und I-Device Kommunikation
V3.2.1	10/2016	Neue Warnungsmeldung; Neues Kapitel, Aufruf Motion FB in IPO Task; Beschreibung der „alten“ Programm-Struktur entfernt