



DOKUMENTATION ISG-kernel

SPS-Bibliothek **SPS Beispiele**

Kurzbezeichnung:
MCP-APEX

© Copyright
ISG Industrielle Steuerungstechnik GmbH
STEP, Gropiusplatz 10
D-70563 Stuttgart
Alle Rechte vorbehalten
www.isg-stuttgart.de
support@isg-stuttgart.de

Dokumentation Version: 1.03
13.11.2024

Vorwort

Rechtliche Hinweise

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

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

Qualifikation des Personals

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

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

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

Weiterführende Informationen

Unter den Links (DE)

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

bzw. (EN)

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

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

Haftungsausschluss

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

Marken und Patente

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

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

Copyright

© ISG Industrielle Steuerungstechnik GmbH, Stuttgart, Deutschland.

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

Allgemeine- und Sicherheitshinweise

Verwendete Symbole und ihre Bedeutung

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

Symbole im Erklärtext

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



GEFAHR

Akute Verletzungsgefahr!

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



VORSICHT

Schädigung von Personen und Maschinen!

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



Achtung

Einschränkung oder Fehler

Dieses Symbol beschreibt Einschränkungen oder warnt vor Fehlern.



Hinweis

Tipps und weitere Hinweise

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



Beispiel

Allgemeines Beispiel

Beispiel zu einem erklärten Sachverhalt.



Programmierbeispiel

NC-Programmierbeispiel

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



Versionshinweis

Spezifischer Versionshinweis

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

Inhaltsverzeichnis

Vorwort	2
Allgemeine- und Sicherheitshinweise	3
1 Definitionen	7
1.1 Abkürzungen	7
1.2 Begriffserklärungen	7
2 SPS-Bibliotheken	9
3 Frame_PLCopenP1.pro: Applikationsrahmen für den Einsatz von PLCopen- Funktionsblöcken	10
3.1 Erforderliche Bibliotheken	10
3.2 Hauptprogramm MAIN()	10
4 PLCopen.pro: Testapplikation für PLCopen-Funktionsblöcke	11
4.1 Anforderungen an die Konfiguration	11
4.2 Erforderliche Bibliotheken	12
4.3 Hauptprogramm MAIN()	12
4.4 Allgemeines zur Implementierung	12
4.4.1 Globale Variablen	13
4.4.2 Standardwerte in PLCopen	13
4.5 Programm Part1_PLCopen	14
4.6 Programm Part1_MCV_Fbs	15
4.7 Programm Part1_TouchProbe	15
4.8 Programm Part4_PLCopen	15
4.9 Programm MotionOfAxes	16
4.10 Programm MCV_Table_access	16
4.11 Programm MCV_TriggerCamIn	16
4.12 Programm Trace	16
4.13 Programm SERCOS_PRG	16
4.14 Visualisierungen für Funktionsblöcke	16
4.15 Visualisierungen der Achsenpositionen	18
4.16 Visualisierung für Kurvenscheibendaten	19
4.17 Visualisierung zur Änderung von Kurvenscheibentabellen	19
5 HLI-BlockSearch – Testapplikation für die unterschiedlichen Satzvorlauf-Varianten	20
5.1 Einleitung	20
5.2 Visualisierung BlockSearch_Common	21
5.2.1 Beschreibung der Elemente der Visualisierung BlockSearch_Common	21
5.2.2 Visualisierungen zur Belegung der spezifischen Satzvorlaufparameter	22
5.3 Ablauf bei aktivem "Block search by visu input" und aktivem "auto return"	24
5.3.1 Ablauf bei aktivem "Block search by visu input" und inaktivem "auto return"	25
5.3.2 Ablauf bei aktivem "PLC code handles block search"	26
5.3.3 Weitere Visualisierungen	26
5.3.3.1 Visualisierung AddFunctionality	26
5.3.3.2 Visualisierung Ax_AxisCoupling	27

5.3.4	Tabelle der Zustände des Satzvorlauf	28
6	Literaturverzeichnis	30
7	Anhang	31
7.1	Anregungen, Korrekturen und neueste Dokumentation.....	31
	Stichwortverzeichnis.....	32

Abbildungsverzeichnis

Abb. 1:	Zuordnung Achs- bzw. Achsgruppenreferenzen zu Funktionsblöcken	11
Abb. 2:	Zusammenhang von Funktionsblöcken und Visualisierung	13
Abb. 3:	Elemente einer Visualisierung	17
Abb. 4:	Visualisierung MotionOfAxes.....	18
Abb. 5:	Visualisierung BlockSearch_Common	21
Abb. 6:	Taste: PLC code handles block search" / "Block search by visu input	21
Abb. 7:	Visualisierung zur Parametrierung des Satzvorlauftyps 1	22
Abb. 8:	Visualisierung zur Parametrierung des Satzvorlauftyps 3	22
Abb. 9:	Visualisierung zur Parametrierung des Satzvorlauftyps 4	23
Abb. 10:	Visualisierung zur Beauftragung weiterer Funktionen	27
Abb. 11:	Visualisierung Ax_AxisCoupling	28

1 Definitionen

1.1 Abkürzungen

AXHLI	Achsspezifisches High-Level-Interface
CM	Continuous Motion (Endlosdrehen)
DM	Discrete Motion (Positionieren)
FB	Function Block (Funktionsbaustein)
FBSD	FB-State Diagram
HLI	High-Level-Interface zwischen MC und PLC
MC	Motion Controller
MCP	Motion Control Platform
MCE	Motion Control Engine
MC-FB	Motion Controller Function Block
NL-Slope	Nicht-Linearer Slope
PCS	Part program coordinate system; Teileprogrammkoordinatensystem
PLC	Programmable Logic Control
POE	Programmorganisationseinheit
SAI	Single Axis Interpolator

1.2 Begriffserklärungen

Achsgruppe	Ein Verbund von Achsen, die durch einen Kanal eine Bewegung auf einer Raumkurve koordiniert durchführen können unter Einhaltung vorgegebener Werte für die Geschwindigkeit, Beschleunigung und Ruck auf dieser Raumkurve.
CoDeSys	SPS-Programmiersystem der Fa. 3S Smart Software Solutions
Funktionssatz	Internes Beauftragungsformat des ISG Motion-Controllers.
HLI-Bibliothek	Zugriff auf die Speicherschnittstelle zur ISG-MCE.
ISG-MCE	Damit ist der ISG NC-Kern gemeint, der im Zusammenhang mit dieser Dokumentation auch als „Motion Control Engine“ bezeichnet wird.
Kanal	Einheit, die Achsbewegungen einer Achsgruppe koordiniert.
MC-FB	Bezeichnet die SPS-Funktionsbausteine, die zur Beauftragung des ISG-MC verwendet werden.
Multiprog	SPS-Programmiersystem der Fa. KW-Software
Motion-Bibliothek	SPS-Softwareapplikation, die Funktionsbausteine zur Bewegung von Achsen entsprechend der PLCopen-Spezifikation, sowie weitere FB, die Aufgaben der Bewegungserzeugung übernehmen, enthält.

Achsgruppe	Ein Verbund von Achsen, die durch einen Kanal eine Bewegung auf einer Raumkurve koordiniert durchführen können unter Einhaltung vorgegebener Werte für die Geschwindigkeit, Beschleunigung und Ruck auf dieser Raumkurve.
CoDeSys	SPS-Programmiersystem der Fa. 3S Smart Software Solutions
Funktionssatz	Internes Beauftragungsformat des ISG Motion-Controllers.
HLI-Bibliothek	Zugriff auf die Speicherschnittstelle zur ISG-MCE.
ISG-MCE	Damit ist der ISG NC-Kern gemeint, der im Zusammenhang mit dieser Dokumentation auch als „Motion Control Engine“ bezeichnet wird.
Kanal	Einheit, die Achsbewegungen einer Achsgruppe koordiniert.
MC-FB	Bezeichnet die SPS-Funktionsbausteine, die zur Beauftragung des ISG-MC verwendet werden.
Motion-Bibliothek	SPS-Softwareapplikation, die Funktionsbausteine zur Bewegung von Achsen entsprechend der PLCopen-Spezifikation, sowie weitere FB, die Aufgaben der Bewegungserzeugung übernehmen, enthält.

Achsgruppe	Ein Verbund von Achsen, die durch einen Kanal eine Bewegung auf einer Raumkurve koordiniert durchführen können unter Einhaltung vorgegebener Werte für die Geschwindigkeit, Beschleunigung und Ruck auf dieser Raumkurve.
Funktionssatz	Internes Beauftragungsformat des ISG Motion-Controllers.
HLI-Bibliothek	Zugriff auf die Speicherschnittstelle zur ISG-MCE.
ISG-MCE	Damit ist der ISG NC-Kern gemeint, der im Zusammenhang mit dieser Dokumentation auch als „Motion Control Engine“ bezeichnet wird.
Kanal	Einheit, die Achsbewegungen einer Achsgruppe koordiniert.
MC-FB	Bezeichnet die SPS-Funktionsbausteine, die zur Beauftragung des ISG-MC verwendet werden.
Multiprog	SPS-Programmiersystem der Fa. KW-Software
Motion-Bibliothek	SPS-Softwareapplikation, die Funktionsbausteine zur Bewegung von Achsen entsprechend der PLCopen-Spezifikation, sowie weitere FB, die Aufgaben der Bewegungserzeugung übernehmen, enthält.

Obligatorischer Hinweis zu Verweisen auf andere Dokumente

Zwecks Übersichtlichkeit wird eine verkürzte Darstellung der Verweise (Links) auf andere Dokumente bzw. Parameter gewählt, z.B. [PROG] für Programmieranleitung oder P-AXIS-00001 für einen Achsparameter.

Technisch bedingt funktionieren diese Verweise nur in der Online-Hilfe (HTML5, CHM), allerdings nicht in PDF-Dateien, da PDF keine dokumentenübergreifenden Verlinkungen unterstützt.

2 SPS-Bibliotheken

Die ISG liefert eine Reihe von SPS-Bibliotheken aus, die von den beschriebenen Applikationen eingebunden werden. Da die ISG verschiedene SPS-Laufzeitsysteme auf unterschiedlichen Betriebssystemen und in unterschiedlichen Steuerungen unterstützt, werden in diesem Kapitel die einzelnen Bibliotheken aufgeführt und ein einheitlicher Name vereinbart. Dieser wird im Weiteren in der Dokumentation verwendet, um eine einheitliche Bezeichnung für die unterschiedlich bezeichneten Bibliotheken zu benutzen, die denselben Inhalt haben.

Seit Einführung von **TwinCAT 3** existieren weitere Varianten der Bibliotheken. Die Bibliotheken lassen sich durch den Präfix vor dem Bibliotheksnamen unterscheiden:

- CNC build = 28xx: Präfix = **pf_** = TcCnc
- CNC build > 3xxx: Präfix = **pf_** = Tc2_Cnc

Übersicht SPS-Bibliotheken

Einheitlicher Name	3S / CoDeSys	TwinCAT	KW/Multiprog
HLI-Bibliothek	hli.lib bzw. 00_CNCHLIVX_XXXX.lib	CNC build = 28xx pf_HliV3.lib CNC build = 3xxx pf_Hli.lib	hli.mwt
Motionbibliothek Basis	McpBase.lib	pf_Base.lib	McpBase.mwt
Motionbibliothek Part1	McpPlcopenP1.lib	pf_PlcopenP1.lib	McpPlcopenP1.mwt
Motionbibliothek Part4	McpPlcopenP4.lib	pf_PlcopenP4.lib	McpPlcopenP4.mwt
Technobibliothek	McpTechTct.lib	pf_TechTct.lib	McpTechTct.mwt
SERCOS-Bibliothek	McpSercos.lib		
Adaptionsbibliothek	Siehe folgende Tabelle		

Varianten der Adaptionsbibliothek

3S / CoDeSys		
RTX	Win32	Linux/Homag
hli_rts_lib.lib	hli_rts_lib.lib	00_CNCHLIVXINIT_XXXX.lib

3 Frame_PLCCopenP1.pro: Applikationsrahmen für den Einsatz von PLCopen- Funktionsblöcken

Dieses Beispielprogramm zeigt, welche Funktionsblöcke anzulegen sind, damit eine Applikation korrekt funktioniert, die Funktionsblöcke nach PLCopen Part1 und Part4 verwenden soll.

3.1 Erforderliche Bibliotheken

In diese Applikation müssen folgende Bibliotheken eingebunden werden:

- Adaptionbibliothek, bei RTX oder Simulationsbetrieb Win32
- HLI-Bibliothek
- Motionbibliothek Basis
- Motionbibliothek Part1

3.2 Hauptprogramm MAIN()

Im Programm MAIN wird zuallererst eine Instanz des FB MCV_HliInterface aus der HLI-Bibliothek durchgerechnet. Dieser FB prüft, ob die Schnittstelle HLI auf Seite der SPS derjenigen auf Seite des Motion-Controllers entspricht. Dazu muss der FB mehrfach berechnet werden. Erst wenn der Ausgang **Initialized** des FB TRUE und der Ausgang **Error** FALSE zeigt, darf auf das HLI, die Schnittstelle zwischen SPS und Motion-Controller, zugegriffen werden.

Aus diesem Grund darf auch erst danach eine Instanz des FB MCV_PlatformBase aufgerufen werden. Mit diesem FB werden die Achsreferenzen initialisiert. Nachdem der FB mit dem Ausgang **Done** = TRUE signalisiert, dass die Achsreferenzen nutzbar sind, kann eine Instanz des FB MCV_P1_PLATFORM durchgerechnet werden. Dieser FB muss zyklisch in der SPS-Applikation aufgerufen werden. Der FB sorgt dafür, dass Fehlermeldungen vom Motion-Controller entgegengenommen und über die Achsreferenzen jedem FB bekannt gemacht werden. Damit kann jeder FB auf Probleme mit der durch ihn beauftragten Achse reagieren.

Nach diesem FB wird der Applikationscode eingefügt.

Dieses Programm MAIN() wird dann einer Task zugewiesen.

4 PLCopen.pro: Testapplikation für PLCopen-Funktionsblöcke

Mit dieser Applikation wird dem Anwender die Möglichkeit gegeben, sich mit dem Verhalten von Funktionsblöcken nach den PLCopen-Standards Part1 und Part4 vertraut zu machen. Die Anwendung ist so angelegt, dass zyklisch sämtliche Instanzen dieser Funktionsblöcke durchgerechnet werden. Es existieren keinerlei Abhängigkeiten zwischen den Funktionsblöcken. Als Beauftragungsschnittstelle dienen dem Benutzer die angelegten Visualisierungen, wobei für jeden Funktionsblocktyp genau eine Visualisierung angelegt wurde.

Das Konzept für dieses Beispiel sieht vor, dass jede Achse oder Achsgruppe immer derselben Funktionsblockinstanz zugeordnet ist. Dies dient dem leichteren Verständnis des Beispiels. Es sind deshalb mindestens so viele Funktionsblockinstanzen angelegt, wie Achsen oder Achsgruppen beauftragt werden sollen.

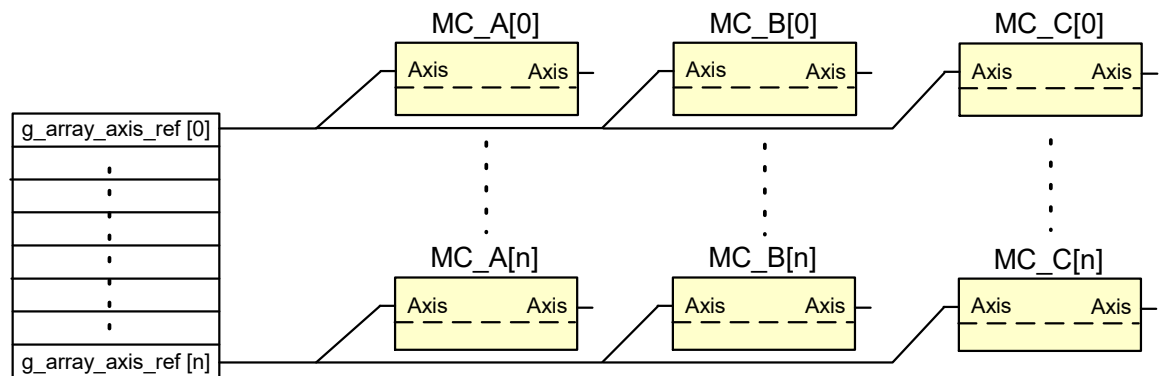


Abb. 1: Zuordnung Achs- bzw. Achsgruppenreferenzen zu Funktionsblöcken

Damit der Nachweis möglich ist, dass die Funktionalität der Funktionsblöcke auch beim Austausch von Achsen und Achsgruppen korrekt erfüllt wird, kann man über 2 globale Variablen (gP1AxisRefChgDisable, gP4AxesGrpRefChgDisable) diesen Austausch zulassen. Diese Betriebsart ist allerdings eher für Demonstrationszwecke gedacht.

Das Beispiel enthält eine einzige Task, die das Hauptprogramm MAIN() aufruft.

4.1 Anforderungen an die Konfiguration

Diese Applikation setzt voraus, dass die konfigurierten Achsen als Achsen mit einem separaten Interpolator (SAI) angelegt werden. Dazu genügt es die Achsen in den Achsparameterlisten als

- Spindeln anzulegen: kenngr.achs_typ 4

oder

- bei anderen Achstypen den Einzelachsinterpolator zu aktivieren durch kenngr.configure_sai 1

Alle die Achsen, die über Funktionsblöcke nach Part 4 der PLCopen-Spezifikation beauftragt werden sollen, müssen bereits bei der Konfiguration den Achsgruppen bekannt gemacht werden und sind deshalb in den Kanalparameterlisten einzutragen.

4.2 Erforderliche Bibliotheken

In diese Applikation müssen folgende Bibliotheken eingebunden werden:

- Adaptionsbibliothek, bei RTX oder Simulationsbetrieb Win32
- HLI-Bibliothek
- Motionbibliothek Basis
- Motionbibliothek Part1
- Motionbibliothek Part4
- Technobibliothek
- SERCOS-Bibliothek

4.3 Hauptprogramm MAIN()

Das Hauptprogramm MAIN basiert auf dem Beispiel Frame_PlcopenP1.pro.

Da in dieser Applikation auch Funktionsblöcke des Part4 der PLCopen-Spezifikation instanziiert werden, muss der Plattformbaustein für die Part4-Funktionsblöcke instanziiert und durchgerechnet werden. Er besitzt den Typ MCV_P4_Platform, erhält den Bezeichner P4_Platform und wird unmittelbar nach dem Plattform-Funktionsblock für die Part1-Funktionsblöcke aufgerufen.

Die FB-Instanzen TechFctChIn und TechFctAxIn dienen dazu, Technologieinformationen, die über M-, H-, S- oder T-Befehle aus dem NC-Programm an die SPS gemeldet werden, aus der Schnittstelle HLI zwischen Motion-Kernel und SPS zu entnehmen und in der SPS bereitzustellen. Die korrespondierenden Funktionsblöcke TechFctChOut und TechFctAxOut am Ende des Programms MAIN quittieren automatisch alle die Technologiefunktionen, die nicht durch das SPS-Projekt behandelt wurden. Zur Behandlung einer Technologiefunktion wird im SPS-Projekt ein für den Technologiefunktionsstyp spezifischer FB instanziiert und mit der Ordnungsnummer der Technologiefunktion versorgt. In diesem Beispielprojekt wird auf diese Möglichkeit kein Augenmerk gelegt, sondern es sollen nur alle eintreffenden Technologiefunktionen quittiert werden, sodass ein NC-Programm ohne Unterbrechung abgearbeitet werden kann.

Nach dem Einlesen der Technologiefunktionen werden nacheinander die Programme aufgerufen, die letztendlich die Funktionsblöcke enthalten, die die Lösung von Bewegungsaufgaben von Einzelachsen oder Achsgruppen ermöglichen.

4.4 Allgemeines zur Implementierung

Den meisten Programmen, die nachfolgend beschrieben sind, liegt dasselbe Prinzip zu Grunde. In der Initialisierungsphase wird jeder Funktionsblockinstanz die für den Betrieb erforderliche Achse oder Achsgruppe über eine Variable des Typs AXIS_REF bzw. AXES_GROUP_REF zugewiesen. Diese Variablen sind bereits als Feld in der Motion-Basisbibliothek angelegt (g_array_axis_ref und gAxesGroupRef). Nach dieser Phase werden in jedem SPS-Zyklus in einer FOR-Schleife alle Funktionsblockinstanzen durchgerechnet.

Da für jeden Funktionsblocktyp nur eine Visualisierung instanziiert angelegt ist, ist es über die Visualisierung möglich diese auf die verschiedenen Funktionsblockinstanzen umzuschalten (s. Visualisierungen für Funktionsblöcke). Um dieses Verfahren abwickeln zu können, ist der restliche Code im Programm erforderlich, der Zeiger auf Funktionsblockinstanzen anwendet.

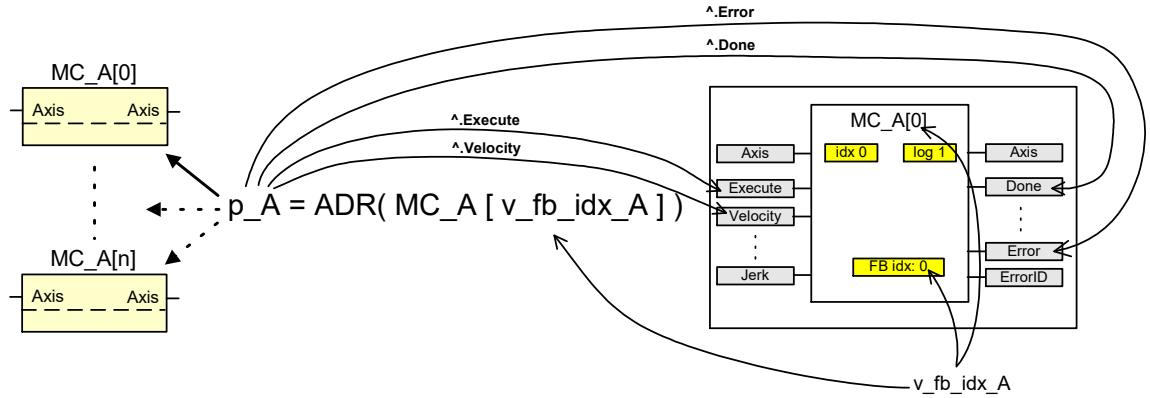


Abb. 2: Zusammenhang von Funktionsblöcken und Visualisierung

4.4.1 Globale Variablen

Die Funktionsblöcke, die von den verschiedenen Programmen aufgerufen werden, sind als globale Variable angelegt und als Felder von Funktionsblöcken organisiert. An dieser Stelle erfolgt auch die Initialisierung der einzelnen Funktionsblockinstanzen.

Auch die für die Versorgung der Visualisierungen erforderlichen Variablen sind als globale Variable angelegt.

4.4.2 Standardwerte in PLCopen

Im Wesentlichen sind Standardwerte für die Beschleunigung, die Verzögerung, die Geschwindigkeit und den Ruck als globale Konstanten angelegt und werden dazu benutzt die Eingangsvariablen der Funktionsblöcke mit geeigneten Standardwerten zu versorgen. Weitere Erläuterungen sind der nachfolgenden Tabelle zu entnehmen. Die restlichen Konstanten skalieren die Applikation hinsichtlich Anzahl Funktionsblöcke je Funktionsblocktyp die angelegt und durchgerechnet werden.

Standardwerte in PLCopen.pro

Wert	Beschreibung
2000.0	Standardwert für die Beschleunigung
2000.0	Standardwert für die Verzögerung
10000.0	Standardwert für den Ruck
25000.0 50000.0 75000.0 100000.0	Standardwerte für die Geschwindigkeit
15	Anzahl der Funktionsblockinstanzen die für Funktionsblocktypen für die Einzelachsbeauftragung angelegt werden. Es können damit gleich viele Achsen beauftragt werden, wenn jeder Achse genau eine Funktionsblockinstanz zugewiesen wird.
8	Anzahl der Funktionsblockinstanzen die für Funktionsblocktypen für die Beauftragung von Master/Salve-Achspaarungen angelegt werden. Es können damit gleich viele Slaveachsen beauftragt werden, wenn jeder Achse genau eine Funktionsblockinstanz zugewiesen wird.
9	Anzahl der Funktionsblockinstanzen die für Funktionsblocktypen für die Achsgruppenbeauftragung angelegt werden. Es können damit gleich viele Achsgruppen beauftragt werden, wenn jeder Achsgruppe genau eine Funktionsblockinstanz zugewiesen wird.

4.5 Programm Part1_PLCopen

Dieses Programm dient dazu Funktionsblockinstanzen, die nach dem Part 1 der PLCopen-Spezifikation implementiert wurden, zu berechnen.

Im Initialisierungsteil wird für die Funktionsblöcke, die sich auf eine einzelne Achse beziehen, wieder die in der Abbildung unter "PLCopen.pro: Testapplikation für PLCopen-Funktionsblöcke [► 11]" gezeigte Beziehung zwischen Achsreferenzen und Funktionsblöcken hergestellt.

```
FOR Idx := 0 TO PLC_AX_MAXIDX DO
  IdxSingleAxFbs := Idx + g_axis_idx_offset;

  IF (IdxSingleAxFbs >= 0) AND
    (IdxSingleAxFbs <= PLC_AX_MAXIDX) THEN
    HomeAxRefIdx[IdxSingleAxFbs] := IdxSingleAxFbs;
    ...
    (* Calculate the administrative Fbs *)
    ReadStatusAxRefIdx[IdxSingleAxFbs] := IdxSingleAxFbs;
    ...
  END_IF;
END_FOR;
```

Für alle Funktionsblöcke die sich auf eine Master- und eine Slaveachse beziehen, wird als Masterachse die erste im System verfügbare SAI-Achse festgelegt. Die einzelnen Funktionsblockinstanzen erhalten dann aber unterschiedliche Referenzen für die Slaveachse.

```
FOR IdxMultAxFbs := 0 TO PLC_MULTIAX_IDX DO
  ...
  GearInMstIdx[IdxMultAxFbs] := IdxMasterAx;
  GearInSlvIdx[IdxMultAxFbs] := IdxSlaveAx;
  GearOutSlvIdx[IdxMultAxFbs] := IdxSlaveAx;
  ...
  IF IdxSlaveAx < GC_MCP_AXREF_MAXIDX THEN
    IdxSlaveAx := IdxSlaveAx + 1;
  ELSE
    IdxMasterAx := IdxMasterAx + 1;
    IdxSlaveAx := 0;
  END_IF;
END_FOR;
```

Nach der Initialisierungsphase werden nur noch die Funktionsblockinstanzen zyklisch berechnet.

Sämtlicher Programmcode nach dem Kommentar

```
(*=====*)
(* The following commmands are only for visualisation. *)
(*=====*)
```

dient lediglich der Versorgung der zugeordneten Visualisierung mit den Daten der angewählten Funktionsblockinstanz.

4.6 Programm Part1_MCV_Fbs

In diesem Programm sind keine Initialisierungen erforderlich und so werden in der FOR-Schleife lediglich die von ISG definierten und zusätzlich in der Motionbibliothek Teil1 gelieferten FB durchgerechnet. Anschließend wird der Code für die Versorgung und Anbindung der Visualisierung ausgeführt.

4.7 Programm Part1_TouchProbe

Dieses Programm dient dazu, die Instanzen der Funktionsblöcke vom Typ MC_AbortTrigger und MC_TouchProbe zu berechnen, welche die Messfunktionalität bereitstellen. Diese Funktionsblöcke besitzen als VAR_IN_OUT-Pin eine Referenz auf die Signalquelle vom Typ TRIGGER_REF. Dieser Datentyp ist in der Bibliothek Motionbibliothek Teil1 definiert. Im Initialisierungsteil des Programms wird für jede FB-Instanz eine solche Referenz angelegt und mit geeigneten Standardwerten belegt.

Nach der Initialisierungsphase werden in der FOR-Schleife alle FB-Instanzen der oben genannten Typen berechnet.

Der restliche Code wird wie schon bei den anderen Programmen ausgeführt lediglich dazu benutzt um die Visualisierung mit den Daten der ausgewählten FB-Instanz zu versorgen.

4.8 Programm Part4_PLCopen

In diesem Programm sind Variablen, die die ..._REF Datentypen (IDENT_IN_GROUP_REF, MC_PATH_DATA_REF, MC_PATH_REF) aus der Spezifikation repräsentieren, als lokale Variablen angelegt und als Felder organisiert.

Im Initialisierungsteil wird wiederum die 1:1-Beziehung der Indizes der Funktionsblockinstanz und der Achsgruppe hergestellt. Außerdem werden einige Eingangsvariablen von Funktionsblöcken initialisiert, damit der Benutzer sinnvolle Werte vorfindet.

Es folgt die bereits bekannte FOR-Schleife, in der sämtliche Instanzen der Funktionsblöcke durchgerechnet werden, und anschließend der Code für die Aktualisierung der notwendigen Daten zur Anzeige in den Visualisierungen.

4.9 Programm MotionOfAxes

Dieses Programm definiert lokale Variablen, in denen die aktuelle Position und ausgewählte Zustandsdaten einzelner Achsen gespeichert wird, nachdem diese aus der Schnittstelle HLI zwischen SPS und Motion-Kernel entnommen worden sind. Die Visualisierungen MotionOfAxes bzw. MotionOfAxes_XY zeigen dann die Werte dieser lokalen Variablen an.

4.10 Programm MCV_Table_access

Dieses Programm ist in der Motionbibliothek Teil1 definiert und erlaubt den Zugriff auf Kurvenscheibentabellen. In der Visualisierung Tbm_Display werden die Daten von zur Verfügung stehenden Kurvenscheibentabellen angezeigt. Diese Daten werden von dem Programm MCV_Table_access bereitgestellt.

4.11 Programm MCV_TriggerCamIn

4.12 Programm Trace

Berechnet eine Funktionsblockinstanz, die die Erzeugung verschiedener Protokolle im Motion-Kernel veranlassen kann.

4.13 Programm SERCOS_PRG

Dieses Programm beauftragt je eine Instanz der Funktionsblöcke mit denen SERCOS-Parameter geschrieben und gelesen werden können. Die Instanzen der Funktionsblöcke sind hierbei lokal im Programm angelegt.

4.14 Visualisierungen für Funktionsblöcke

Für jeden der implementierten Funktionsblocktypen ist in der Applikation eine Visualisierung angelegt, mit der bei jeder Instanz eines Funktionsblocktyps die Werte der Eingangsvariablen verändert werden können und die Werte der Ausgangsvariablen angezeigt werden. Außer den Elementen, die die Eingangs- und Ausgangsvariablen entsprechend der Spezifikation repräsentieren, gibt es noch einige Elemente, die für die Bedienung der Visualisierung wichtig sind. Deren Funktion wird nachfolgend kurz beschrieben:

Elemente in Visualisierungen für Funktionsblöcke

Nr.	Bezeichnung	Beschreibung
1	Funktionsblock-Index	<p>Die Instanzen eines Funktionsblocktyps sind in dieser Applikation als Feld von Funktionsblöcken angelegt. Die Nummer ist der Index des Feldalementes das angezeigt wird, also der Index des mit der Visualisierung verbundenen Funktionsblockes.</p> <p>Diese Nummer ist über die Visualisierung änderbar, was dem Anwender die Möglichkeit gibt die unterschiedlichen Funktionsblockinstanzen zu beauftragen und ihre Statusdaten abzufragen.</p>
2	Referenz-Index	<p>In der PLCopen-Spezifikation sind unterschiedliche Datentypen deklariert, die eine Achse oder Achsgruppe usw. repräsentieren. Diese Datentypen dienen als Referenz für die aufgeführten Elemente.</p> <p>Durch die implementierten SPS-Bibliotheken werden diese Referenzen als Feld von Referenzen definiert (<code>g_array_axis_ref</code>, <code>gAxesGroupRef</code>) oder bei anderen Datentypen als Feld innerhalb der Applikation angelegt.</p> <p>Diese Nummer bezeichnet den Index des Elementes aus einem solchen Feld von Referenzen, das beim ausgewählten FB an den entsprechenden Ein-/Ausgangsvariablen anliegt.</p> <p>Dieser Wert ist für Achs- Achsgruppenreferenzen nur dann editierbar, wenn die globalen Variablen <code>gP1AxisRefChgDisable</code>, <code>gP4AxesGrpRefChgDisable</code> [11] auf FALSE gesetzt werden.</p>
3	HLI-Index	<p>Dies ist ein Statusdatum und deshalb nicht editierbar. Die Schnittstelle zwischen Motion-Kernel und SPS ist als Feld von Strukturen organisiert, denen Achsen bzw. Achsgruppen zugeordnet sind. Die Nummer ist der Index der achs- bzw. achsgruppenspezifischen Struktur, die über den angewählten Funktionsblock beauftragt wird.</p>
4	Logische Nummer	<p>Ordnungsnummer für Achsen und Achsgruppen.</p>
5	All-Taste	<p>Das Betätigen dieser Taste führt dazu, dass bei sämtlichen Funktionsblockinstanzen der Execute-Eingangspin auf TRUE gesetzt wird.</p> <p>Solch eine Taste ist nicht in allen Visualisierungen vorhanden.</p>

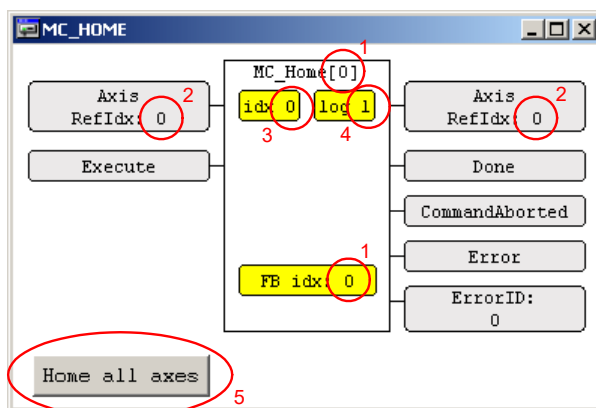


Abb. 3: Elemente einer Visualisierung

4.15 Visualisierungen der Achsenpositionen

Die Visualisierung MotionOfAxes dient dazu die aktuellen Positionen von bis zu 10 Achsen anzuzeigen. Die Positionen werden von dem gleichnamigen Programm aus dem HLI entnommen und in lokalen Variablen gespeichert, die dann von der Visualisierung angezeigt werden. Zusätzlich werden weitere Zustandsdaten für die jeweilige Achse angezeigt, deren Bedeutung nachfolgend beschrieben wird.

Elemente zur Visualisierung von Zustandsdaten einer Achse

Zustandsdatum	Beschreibung
Ready	Wird für Ready Grün angezeigt, so bedeutet dies dass der Antrieb betriebsbereit ist und sich Regelung befindet. Ansonsten wird Rot angezeigt.
Error	Wird für Error Grün angezeigt, so ist kein Fehler aufgetreten. Im Fenster neben der Fehlerstatusanzeige, wird die aktuelle Position der Achse angezeigt.
Homed	Ist die Achse referenziert, wird Grün angezeigt, ansonsten Rot
IsSAI	Wird Grün angezeigt, so wird diese Achse von einem Einzelachsinterpolator versorgt und kann mit Bausteinen nach Part1 der PLCopen-Spezifikation betrieben werden. Die Farbe Rot zeigt an, dass die Achse in eine Achsgruppe aufgenommen wurde und nun über Funktionsblöcke nach Part 4 der PLCopen-Spezifikation beauftragt werden kann.
IsChAx	Die Farbe Grün zeigt an, dass die Achse nun in eine Achsgruppe aufgenommen ist. Im Fenster daneben wird die Achsgruppennummer der entsprechenden Achsgruppe angezeigt.

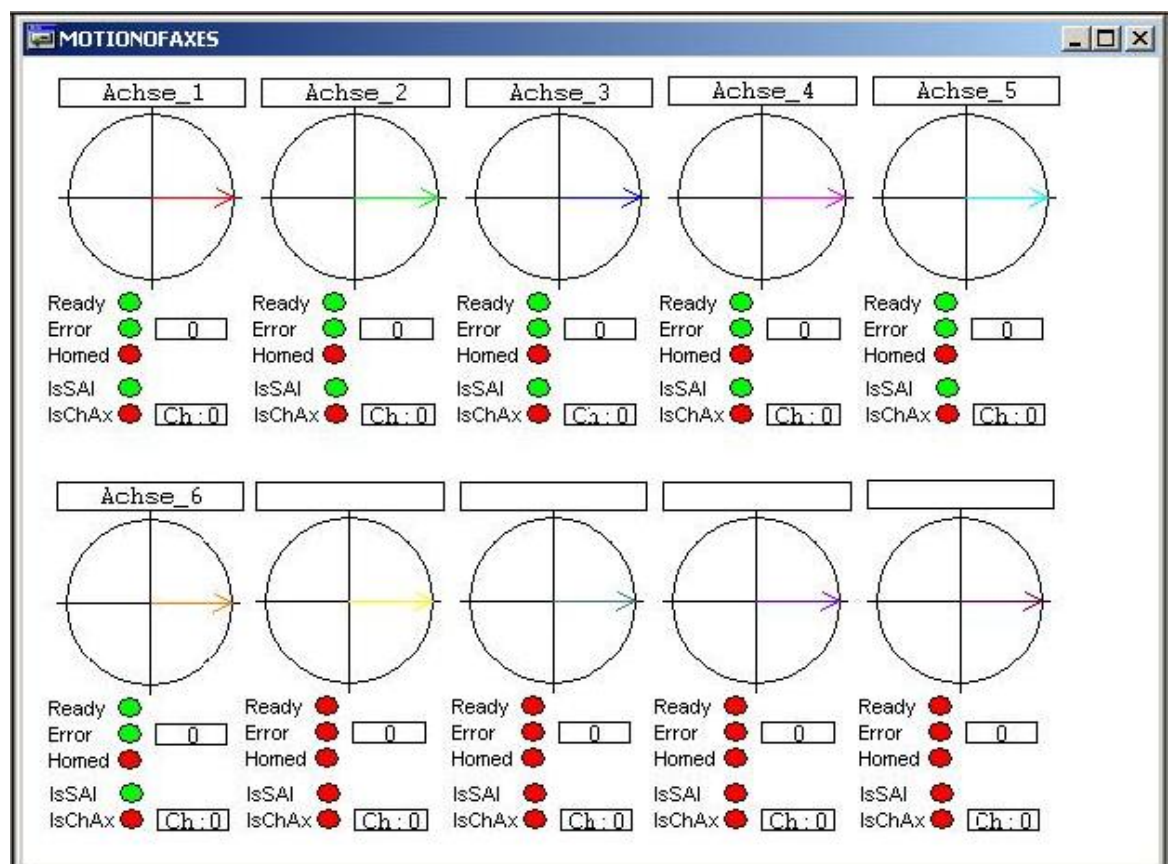


Abb. 4: Visualisierung MotionOfAxes

4.16 Visualisierung für Kurvenscheibendaten

TBM_DISPLAY

Table Descriptions

idx	name	id	type	function	lines	ready	hooked	locked
0	f_Cos_180°	400	3	1	3600	ready	hooked	locked
1	Sinus_360°	401	3	1	36	ready	hooked	locked
2	poly5line_2	402	7	5	12	ready	hooked	locked
3	poly5line_3	403	7	5	4	ready	hooked	locked
4	Smallest_Table	404	7	4	2	ready	hooked	locked
5	Largest_Table	405	7	4	360	ready	hooked	locked
6	poly5line_6	406	7	4	4	ready	hooked	locked
367		0	0	0	0	ready	hooked	locked
368		0	0	0	0	ready	hooked	locked
369		0	0	0	0	ready	hooked	locked

4.17 Visualisierung zur Änderung von Kurvenscheibentabellen

CHANGE_TABLE_LINE

Show and Change Table Line

Tab-Idx.

Fkt.-Type

New Fkt.-Type

Tab-Line Col-Index

Act-Val.

New-Val.

5 HLI-BlockSearch – Testapplikation für die unterschiedlichen Satzvorlauf-Varianten

5.1 Einleitung

Die grundlegenden Mechanismen, die für den Satzvorlauf implementiert wurden, sind der Funktionsbeschreibung "Satzvorlauf" (FCT-C6) zu entnehmen. Im Weiteren wird auf die Beauftragung der verschiedenen Satzvorlauf-Varianten über die PLC-Applikation HLI-BlockSearch eingegangen.

Die PLC-Applikation dient dazu, den Satzvorlauf zu parametrieren und nach dem Starten des Satzvorlaufs die Reaktion des NC-Kerns zu beobachten. Der Satzvorlauf selber wird dadurch aktiv, dass über die SPS-Applikation ein Satzvorlauftyp angewählt, die erforderlichen Parameter gesetzt wurden und dann ein Start eines NC-Programms beauftragt wird. Dies geschieht im Fall der Beispielapplikation üblicherweise über das Registerblatt **Online** des entsprechenden Kanals im SystemManager (TwinCAT 2) oder die MOTION-Konfiguration (TwinCAT 3), indem die **Start**-Taste dort gedrückt wird. Im weiteren Text wird dieses Registerblatt vereinfacht Bedienung genannt.

Die Parametrierung des Satzvorlaufs erfolgt über die ADS-Kommunikation. Es gibt keine entsprechenden Schnittstellen hierzu auf dem HLI. Über ADS und das HLI können allerdings Zustandsdaten des NC-Kerns abgegriffen werden, die dann zur Parametrierung des Satzvorlaufs über die ADS-Kommunikation verwendet werden können. Der Handshake zwischen PLC und NC-Kern während des Satzvorlaufs wird über die entsprechende Control Unit (CNC build \geq 2800: **gp-Ch[i].bahn_lc_control.block_search**) auf dem HLI abgewickelt (s. a. [HLI//Satzvorlauf], auch Schaubilder zu Handshake).

Durch die PLC-Applikation werden alle vom NC-Kern ausgegebenen Technologiefunktionen (M, H, S, T) automatisch für den Kanal 1 und 2 quittiert (siehe Programm MAIN(), Fkt: QuitChTechFunction()).

Mit der PLC-Applikation kann der Satzvorlauf parametriert und anschließend gewählt werden, ob die zur Durchführung der Funktion erforderlichen Handshakes automatisch von der PLC-Applikation durchgeführt (s. Zustandautomat am Ende von POU BlockSearch) oder der Benutzer diese Schritt-für-Schritt über die Visualisierung beauftragen will.

Die Visualisierungen sind so aufgebaut, dass

- die Daten in der Spalte unter der Bezeichnung "**read command**" die aktuell wirksamen Parameter im NC-Kern für den Satzvorlauf sind und per ADS ausgelesen wurden.
- die Daten in der Spalte unter "**command to set**" diejenigen Parameter sind, die bei der nächsten Beauftragung eines Satzvorlaufs aktiv werden, wenn sie nach dem Eintragen in das Visualisierungsfeld durch Betätigen der **set**-Taste per ADS an den NC-Kern übertragen wurden.
- die in der Spalte "**HLI act.value**" angezeigten Zustandsdaten auf dem HLI mit der **copy**-Taste in das zugehörige "**command to set**"-Feld übernommen werden können. Dies wird genutzt, um die Daten auf dem HLI z. B. nach einem Programmabbruch für die Parametrierung des Satzvorlaufs zu übernehmen.

5.2 Visualisierung BlockSearch_Common

Über diese Visualisierung wird im oberen Bereich festgelegt, ob die erforderlichen Handshakes von der PLC-Applikation automatisch durchgeführt werden oder ob der Benutzer die erforderlichen Handshakes über die entsprechenden Visualisierungselemente beauftragt.

Der grün hinterlegte Bereich dient dazu, die für alle Satzvorlaufvarianten möglichen Parameter an den NC-Kern zu übertragen.

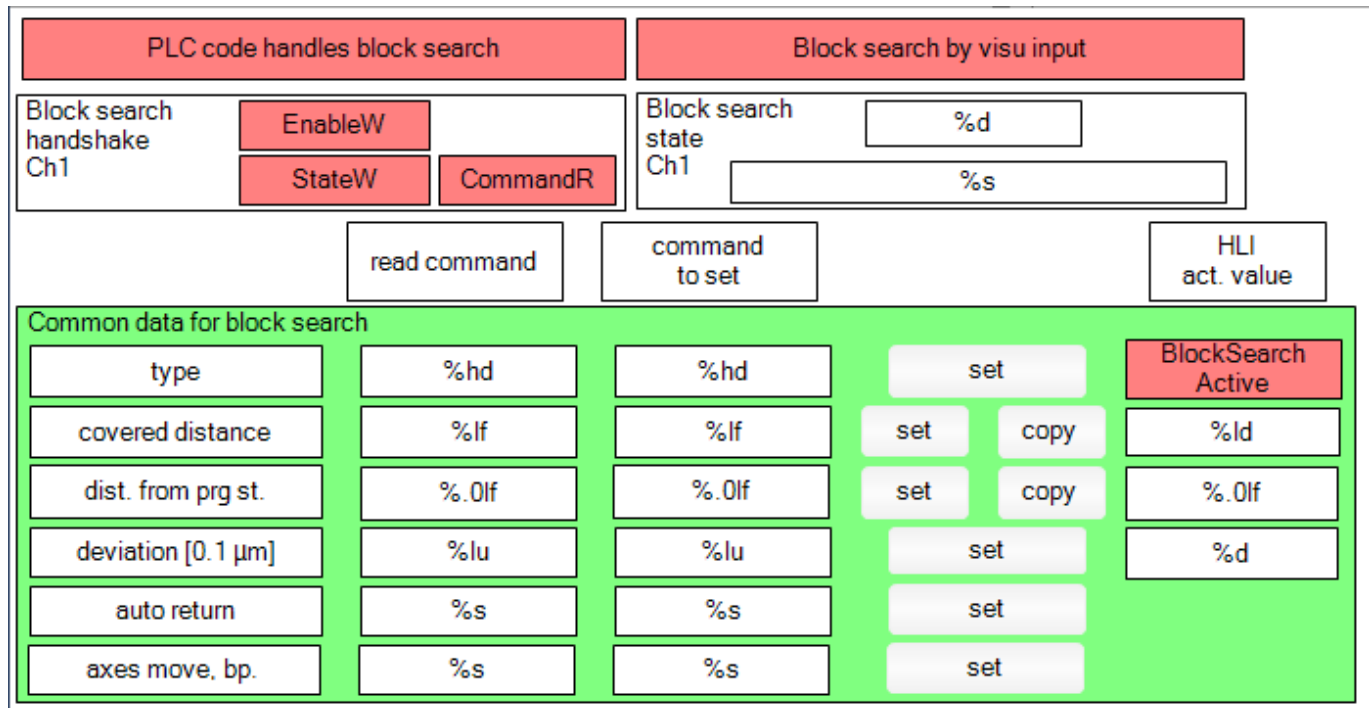


Abb. 5: Visualisierung BlockSearch_Common

5.2.1 Beschreibung der Elemente der Visualisierung BlockSearch_Common

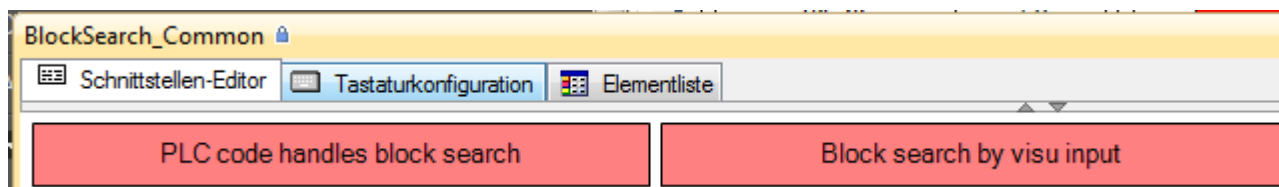


Abb. 6: Taste: "PLC code handles block search" / "Block search by visu input"

Nur eine der beiden Tasten kann aktiv sein.

Nach dem Starten der PLC-Applikation ist durch Vorbelegung die Taste **"Block search by visu input"** sowie die Control Unit für den Satzvorlauf auf dem HLI ist aktiviert (enable_w der Control Unit ist TRUE). Dies zeigt die grüne Anzeige "EnableW" im Bereich **"Block search handshake CH1"** im linken Bereich unterhalb der beiden Tasten.

Wird **"PLC code handles block search"** aktiviert, werden bei parametrimtem Satzvorlauf und gestartetem NC-Programm alle Handshakes die erforderlich sind durch die Zustandsmaschine abgehandelt, die in der POU BlockSearch (ein Programm) am Ende implementiert ist.

Allgemeine Satzvorlaufparameter

Parameter	Beschreibung	Einheit
Type	Satzvorlauftyp	
covered distance	Angabe der in einem NC-Satz zurückgelegten Distanz	‰
dist. from prg st.	Zurückgelegte Distanz seit Programmstart oder seit dem letzten Auftreten des NC-Befehls #DISTANCE PROG START CLEAR und Aktivierung der Berechnung durch #DISTANCE PROG START ON.	0.1 µm
Deviation	maximale Abweichung der Achsen zwischen Istposition und Fortsetzposition bei Wiederaufnahme der Bearbeitung nach Satzvorlauf	0.1 µm
axes move bp	Gibt eine automatische Unterbrechungsstelle über die Angabe des Abstands ab Programmanfang an	0.1 µm

5.2.2 Visualisierungen zur Belegung der spezifischen Satzvorlaufparameter

Die Visualisierungen BlockSearch_Type_1, BlockSearch_Type_3 und BlockSearch_Type_4 dienen dazu, die Satzvorlaufparameter für den entsprechenden Satzvorlauftyp an den NC-Kern zu übertragen.

Beschreibung der jeweiligen Satzvorlauftypen siehe Funktionsbeschreibung "Satzvorlauf" (FCT-C6).

Block search type: 1

Start:

file offset	%d	%d	set	copy	%d
filename	%s	%s	set	copy	%s
program path type	%d	%d	set		
pass count	%d	%d	set		

End:

file offset	%d	%d	set		
filename	%s	%s	set	copy	%s
program path type	%d	%d	set		
pass count	%d	%d	set		

Abb. 7: Visualisierung zur Parametrierung des Satzvorlauftyps 1

Block search type: 3

block count	%ld	%ld	set	copy	%ld
-------------	-----	-----	-----	------	-----

Abb. 8: Visualisierung zur Parametrierung des Satzvorlauftyps 3

Block search type: 4

block number	%ld	%ld	set	copy	%ld
pass count	%ld	%ld	set		

Abb. 9: Visualisierung zur Parametrierung des Satzvorlaufstyps 4

5.3 Ablauf bei aktivem "Block search by visu input" und aktivem "auto return"



Hinweis

Start X bedeutet in der weiteren Beschreibung, dass die Starttaste des SystemManager (Twin-CAT 2) oder der MOTION-Konfiguration (TwinCAT 3) im Online-Registerblatt des entsprechenden Kanals betätigt wird.

Start 1: NC-Programm starten nach Parametrierung des Satzvorlaufs

Ist eine Satzvorlauf-Variante durch Parametrierung angewählt, zeigt der NC-Kern nach dem Start des NC-Programms durch Setzen von **command_r** der Control Unit an, dass der Satzvorlauf aktiv ist. Dies zeigt die Schaltfläche "**CommandR**" der Visualisierung **BlockSearch_Common** an, indem sie nun grün wird. Die Achsen bewegen sich während des Satzvorlaufs nicht.

Rechts davon im Bereich "**Block search state Ch1**" wird der Zustand des Satzvorlaufs angezeigt und durch einen Text beschrieben. Der Zustand stammt vom Element **gp-Ch[i].bahn_state.block_search_state_r** (CNC build >= 2800) auf dem HLI. Zu diesem Zeitpunkt ist der Zustand 1.

Handshake 1

Die PLC muss nun bestätigen, dass sie informiert darüber ist, dass der Satzvorlauf aktiv ist und ihre vorbereitenden Aktionen abgeschlossen sind. Sie tut dies indem sie **state_w** der Control Unit auf TRUE setzt. Für die PLC übernimmt dies der Benutzer indem er auf die Schaltfläche "**StateW**" klickt und diese darauf mit grün zeigt, das **state_w** der Control Unit auf TRUE gesetzt wurde.

Nun läuft der Satzvorlauf. Die Achsen bewegen sich immer noch nicht. Solange der Satzvorlauf läuft wird der Zustand 2 angezeigt. Am Ende des Satzvorlaufs wird der Zustand 3 angezeigt und **command_r** der Control Unit auf FALSE gesetzt. Das spiegelt sich in der Visualisierung **BlockSearch_Common** wieder, so dass nun die Schaltfläche "**CommandR**" wieder den Ursprungszustand zeigt – helles Rot.

Handshake 2

Die PLC muss nun bestätigen, dass sie das Ende des Satzvorlaufs mitbekommen hat. Der Benutzer muss deshalb auf die Schaltfläche "**StateW**" klicken. Damit wird **state_w** der Control Unit auf FALSE gesetzt.

Danach zeigt der Satzvorlauf den Zustand 4.

Start 2: Anfahren an die Kontur

Unter der Voraussetzung, dass der Satzvorlaufparameter "**auto return**" mit 1 belegt wurde, fahren nun die Achsen auf **direktem** Weg an die durch die Satzvorlaufparametrierung vorgegebene Position (Fortsetzposition) und halten dann an.

Solange die Achsen zur Fortsetzposition fahren wird der Zustand 5 angezeigt. Ist die Fortsetzposition erreicht wird der Zustand 6 angezeigt.

Start 3: Fortsetzen des NC-Programms

Wird jetzt Starttaste in der Bedienung erneut betätigt wird die Abarbeitung des NC-Programms fortgesetzt und die Achsen bewegen sich entsprechend.

5.3.1 Ablauf bei aktivem "Block search by visu input" und inaktivem "auto return"

NC-Programm wird im Satzvorlauf gestartet und "auto return" ist als **inaktiv parametriert**



Hinweis

Sind die Achsen nach einem NC-Programmabbruch bewegt worden, so muss bei **inaktivem "auto return"** gewährleistet sein, dass sich die Achsen

1. entweder vor dem Start des NC-Programms im Satzvorlauf an der Position (Fortsetzposition) befinden, die durch die anderen Satzvorlaufparameter vorgegeben wird
2. oder der Wert für die **"deviation"** (Abweichung) so gewählt ist, dass die aktuelle Istposition der Achsen zu einer Abweichung von den Sollpositionen führt, die kleiner als die parametrierte **"deviation"** ist.

Ist dies nicht der Fall erhält man die Fehlermeldung 50474 – "Abweichung von Kontur nach manuellem Wiederanfahren im Satzvorlauf zu groß"(P-ERR-50474).



Hinweis

Die vom Anwender vorgegebenen Satzvorlaufparameter wird nach dem Satzvorlauf im NC-Kern berechnet, von der aus das NC-Programm dann abgearbeitet wird und die Achsen tatsächlich verfahren.

Unterscheidet sich diese Position von der realen Position, die sich aus der Position der Maschinenachsen ergibt, bedeutet dies, dass das NC-Programm bei Start nach erfolgtem Satzvorlauf, mit dem entsprechenden Offset auf der Werkzeugbahn verfahren wird.

Mit dem Satzvorlaufparameter **"deviation"** (Abweichung) kann festgelegt werden, ob eine Abweichung zulässig ist und wie groß diese maximal ausfallen darf. Es ist dann aber zu bedenken, dass damit zulässig ist, das NC-Programm nach dem Satzvorlauf mit dem vorhandenen Offset abzuarbeiten, wenn dieser kleiner als die parametrierte **"deviation"** ist.



Achtung

Bei **inaktivem "auto return"** führt der Start 2 **nicht** zum Anfahren an die Kontur (Fortsetzposition), sondern es wird das NC-Programm ausgeführt und die Achsen bewegen sich – entspricht also Start 3.

5.3.2 Ablauf bei aktivem "PLC code handles block search"

Ist in der Visualisierung die Taste **"PLC code handles block search"** muss der Benutzer außer dem Parametrieren des Satzvorlauftyps und anschließendem Starten des NC-Programms keine Interaktion mehr über eine Visualisierung ausführen.

Die erforderlichen Handshakes und Starts werden vom der Zustandsautomat am Ende der POU BlockSearch quittiert und beauftragt.



Hinweis

Wie man dem Zustandsautomat entnehmen kann, kann der Start 2 und Start 3 auch über die Control Unit `gpCh[..]^bahn_mc_control.continue_motion` (CNC build > 2800) beauftragt werden.

5.3.3 Weitere Visualisierungen

5.3.3.1 Visualisierung AddFunctionality

Die Visualisierung **AddFunctionality** ermöglicht die Beauftragung weiterer Funktionen, die je nach Lösung eines Anwendungsfalles, im Zusammenhang mit Satzvorlauf angewendet werden.

So wird

- der Bereich **"Start NC program at file offset"** dazu verwendet ein NC-Programm an der Stelle zu starten, die durch den vorgegebenen Dateioffset definiert wird. Die **set-** und **copy-**Taste dienen demselben Zweck wie bei den anderen Visualisierungen.
- der Bereich **"ContinueMotion CU"** dazu verwendet, die entsprechende Control Unit zu beauftragen. Dies ist von Interesse, wenn man Start 2 und Start 3 über diese Control Unit beauftragen will.
- die Taste **"For-/Backward"** dazu verwendet das NC-Programm auch rückwärts abfahren zu können. Diese Taste wirkt über die Control Unit `gpCh[..]^bahn_mc_control.backward_motion` (CNC build > 2800) auf den NC-Kern ein.
- der Bereich **"Suspend axis output"** wirkt auf die Control Unit `gpCh[..]^bahn_mc_control.suspend_axis_output` (CNC build > 2800). Damit kann die Sollwertausgabe eines Kanals an sie physikalischen Achsen unterbunden werden und diese von einem anderen Kanal angefordert und bewegt werden.

Implementiert ist die Beauftragung der Control Units in der POU "AddFunctionality".

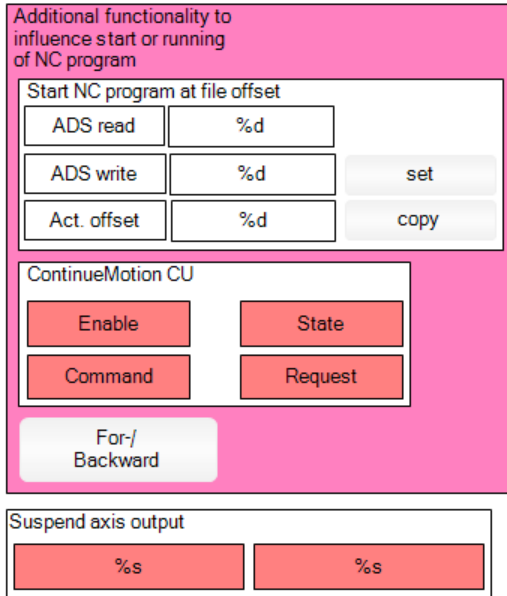


Abb. 10: Visualisierung zur Beauftragung weiterer Funktionen

5.3.3.2 Visualisierung Ax_AxisCoupling

Damit wird die Control Unit `gpAx[.].lr_mc_control.axis_coupling` (CNC build > 2800) genutzt mit der Achskopplungen für eine Achse beauftragt werden. Damit kann die Bewegung einer Achse durch die Bewegung von anderen Achsen zusätzlich oder exklusiv beeinflusst werden.

Die Beauftragung dieser Control Unit findet sich in der POU `ISG_FB_AXCU_AxisCoupling`.

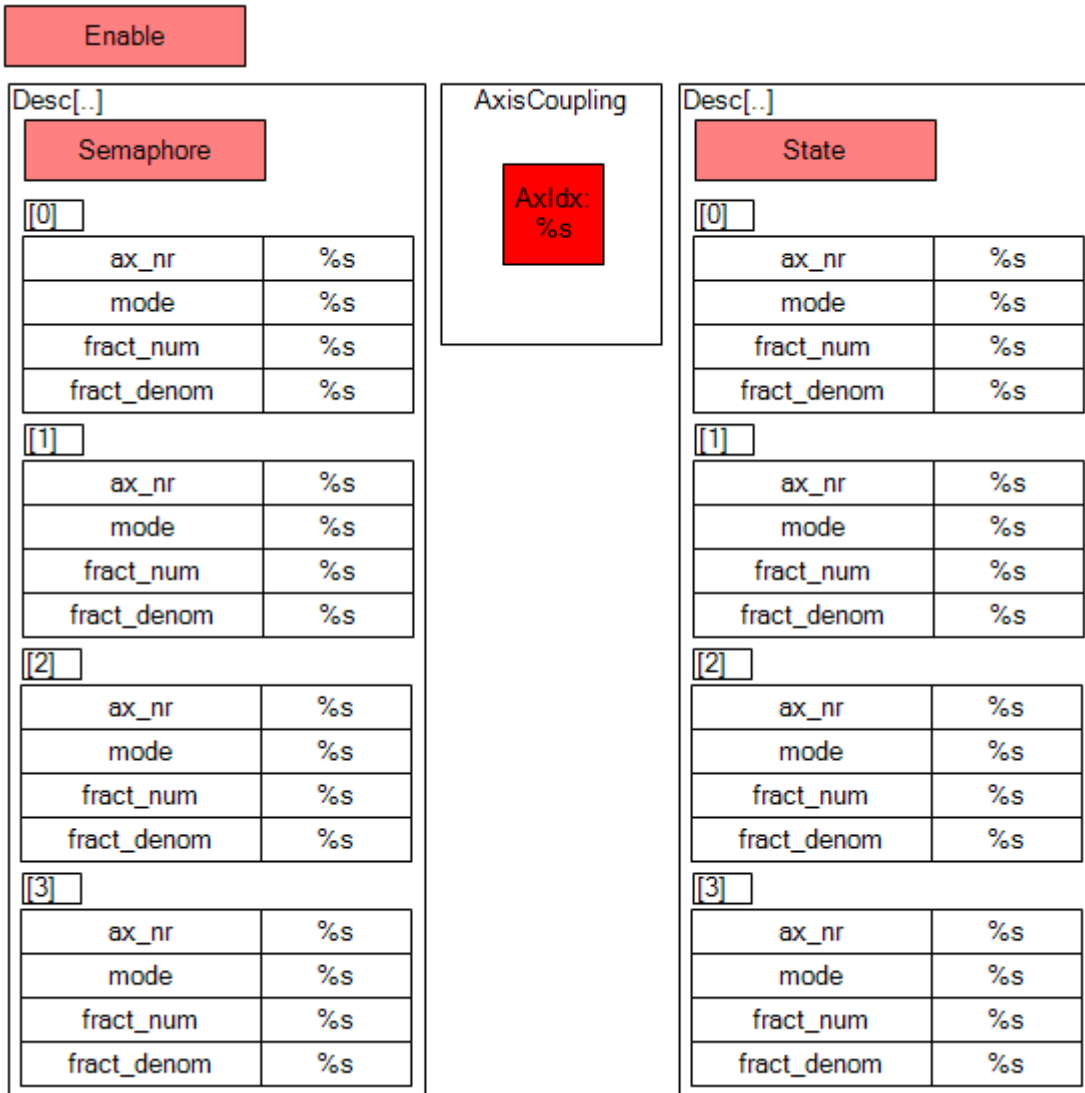


Abb. 11: Visualisierung Ax_AxisCoupling

5.3.4 Tabelle der Zustände des Satzvorlauf

Zahl	Konstante aus HLI-Bibliothek	Beschreibung
0	HLI_BS_INACTIVE	Satzvorlauf ist nicht aktiv
1	HLI_BS_WAIT_FOR_PLC_ON	NC-Kern wartet darauf, dass PLC signalisiert, dass sie detektiert hat, dass Satzvorlauf aktiv ist.
2	HLI_BS_ACTIVE	Satzvorlauf ist aktiv. Dies ist ein dynamischer Zustand, weshalb auf diesen nicht geprüft werden sollte.
3	HLI_BS_WAIT_FOR_PLC_OFF	NC-Kern wartet darauf, dass die PLC signalisiert, dass sie detektiert hat, dass der Satzvorlauf beendet ist.
4	HLI_BS_WAIT_RETURN_TO_CONTOUR	NC-Kern wartet darauf, dass Signal zum Anfahren an die Kontur gegeben wird. Dies geschieht entweder über die Control Unit "Continue motion" (Bewegung fortsetzen) oder über einen Programmstart (Automatic/Active aus PLC oder Start im SystemManager)
5	HLI_BS_RETURNING_TO_CONTOUR	NC-Kern bewegt die Achsen so, dass auf direktem Weg an die Kontur angefahren wird.
6	HLI_BS_WAIT_FOR_CONTINUE_CONTOUR	Die Achsen sind so verfahren, dass das Werkzeug an der Wiederaufsetzposition steht. Nach einem Auftrag über die Control Unit "Continue motion" oder ein Programmstart im SystemManager wird die Bearbeitung ab dem Wiederaufsetzpunkt fortgesetzt.

6 Literaturverzeichnis

[1] PLCopen-Spezifikation: TC2 Task Force Motion Control "Function Blocks for motion control"
Version 1.0, vom 23.Nov.2001

[2] Dokumentation CNC SPS Steuerungsgesamtsystem

[3] Das PLCopen Compliance Statement V1.0 von ISG ist auf der PLCopen Homepage
(www.plcopen.org) zu finden

7 Anhang

7.1 Anregungen, Korrekturen und neueste Dokumentation

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



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

Der o.g. Link ist eine Weiterleitung zu:

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



Hinweis

Mögliche Änderung von Favoritenlinks im Browser:

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

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

PDFs zum Download:

DE:

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

EN:

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

E-Mail: documentation@isg-stuttgart.de

Stichwortverzeichnis

D

D-MCP-APEX-Frame_PLCopenP1	10
D-MCP-APEX-PLCopen.....	11



© Copyright
ISG Industrielle Steuerungstechnik GmbH
STEP, Gropiusplatz 10
D-70563 Stuttgart
Alle Rechte vorbehalten
www.isg-stuttgart.de
support@isg-stuttgart.de

