

MTX 15VRS

OPC-Kommunikation

Projektierungsbeschreibung
R911399271

Ausgabe 01



Titel	MTX 15VRS OPC-Kommunikation
Art der Dokumentation	Projektierungsbeschreibung
Dokumentations-Type	DOK-MTX***-OPC*COM*V15-PR01-DE-P
Interner Ablagevermerk	RS-67276f289d6743ce0a347e86707807b3-1-de-DE-8
Änderungsverlauf	Ausgabe 01, 2019-07 Siehe Tab. 1-1 "Änderungsverlauf" auf Seite 1
Schutzvermerk	© Bosch Rexroth AG 2019 Alle Rechte vorbehalten, auch bzgl. jeder Verfügung, Verwertung, Reproduktion, Bearbeitung, Weitergabe sowie für den Fall von Schutzrechtsanmeldungen.
Verbindlichkeit	Die angegebenen Daten dienen allein der Produktbeschreibung und sind nicht als zugesicherte Eigenschaften im Rechtssinne zu verstehen. Änderungen im Inhalt der Dokumentation und Liefermöglichkeiten der Produkte sind vorbehalten.
Redaktion	Entwicklung Software und Steuerungs-Firmware - Steuerungsfunktionen Motion und CNC, EbSc (TaDo)

Inhaltsverzeichnis

	Seite
1 Zu dieser Dokumentation.....	1
1.1 Gültigkeit der Dokumentation.....	1
1.2 Zweck.....	1
1.3 Erforderliche und ergänzende Dokumentationen MTX.....	2
1.3.1 Auswählen/Zusammenstellen.....	2
1.3.2 Konfigurieren.....	2
1.3.3 In Betrieb nehmen.....	3
1.3.4 Bedienen.....	4
1.3.5 OEM-Entwicklung.....	5
1.3.6 AddOns.....	5
1.4 Darstellung von Informationen.....	5
1.4.1 Sicherheitshinweise.....	5
1.4.2 Verwendete Symbole.....	6
1.4.3 Bezeichnungen und Abkürzungen.....	6
1.4.4 Kundenfeedback.....	6
2 Einleitung.....	7
2.1 Haftungsausschluss.....	7
2.2 OPC-Tag Aufbau.....	7
2.3 OPC-Test-Client.....	9
3 Wichtige Gebrauchshinweise.....	11
3.1 Bestimmungsgemäßer Gebrauch.....	11
3.1.1 Einführung.....	11
3.1.2 Einsatz- und Anwendungsbereiche.....	12
3.2 Nicht-bestimmungsgemäßer Gebrauch.....	12
4 Items.....	13
4.1 Allgemein.....	13
4.1.1 DeviceTypeTest2.....	13
4.2 Root.....	13
4.2.1 Übersicht.....	13
4.2.2 Root Items.....	13
4.3 Firmware.....	19
4.3.1 Übersicht.....	19
4.3.2 Firmware Items.....	20
4.3.3 Fehler-Codes.....	30
4.4 Sercos.....	31
4.4.1 Übersicht.....	31
4.4.2 Sercos Idents.....	31
4.4.3 Sercos Elemente.....	31
4.4.4 Sercos Attribute und Datentypen.....	32
4.4.5 Sercos Listen.....	33

Inhaltsverzeichnis

	Seite
4.4.6 Anfordern mehrerer Elemente.....	34
4.4.7 Sercos Attributecache.....	35
4.4.8 Sercos Fehlercodes.....	36
4.4.9 Sercos Items.....	38
4.5 Diagnose und Logbuch.....	46
4.5.1 Übersicht.....	46
4.5.2 Diagnose.....	48
4.5.3 Logbuch.....	58
4.6 IndraLogic.....	64
4.6.1 Übersicht.....	64
4.6.2 Firmware.....	65
4.6.3 Info.....	66
4.6.4 Eingriff.....	68
4.6.5 Feldbus.....	69
4.6.6 Variablen.....	69
4.7 NC - global.....	74
4.7.1 Allgemeine Informationen.....	74
4.7.2 Antriebe.....	94
4.7.3 CPL-Daten.....	100
4.7.4 Dateibasierte Tabellen.....	103
4.7.5 Datenbankbasierte Tabellen.....	107
4.7.6 Maschinenparameter.....	113
4.7.7 Systemdaten.....	117
4.7.8 Systemfunktionen.....	123
4.8 NC - kanalabhängig.....	149
4.8.1 Allgemeine Kanalinformationen.....	149
4.8.2 Achspositionen.....	157
4.8.3 Achskorrekturen.....	167
4.8.4 Abgleichfunktionen.....	178
4.8.5 Spindeldaten.....	181
4.8.6 Prozessdaten.....	187
4.8.7 Vorschub- und Drehzahlwerte.....	201
4.9 NC-Kommandos.....	207
4.9.1 Übersicht.....	207
4.9.2 Globale Kommandos.....	207
4.9.3 Kanalabhängige Kommandos.....	210
4.10 Fehlermeldungen.....	223
4.10.1 Fehlerbehandlung.....	223
4.10.2 SAV-Fehler.....	225
4.10.3 MACODA-Fehler.....	227
4.10.4 TAD-Fehler.....	227
4.10.5 DBD-Fehler.....	228
4.10.6 CFG-Fehler.....	229
4.10.7 SD-Fehler.....	231
4.10.8 XML-Fehler.....	232
4.10.9 TAR-Fehler.....	237

	Seite
4.10.10 Unlock-Fehler.....	239
5 Service und Support.....	241
 Index.....	243

1 Zu dieser Dokumentation

Ausgaben dieser Dokumentation

Ausgabe	Stand	Bemerkung
01	2019-07	Erstausgabe

Tab. 1-1: Änderungsverlauf

1.1 Gültigkeit der Dokumentation

Übersicht über Zielgruppen und Produktphasen

In der folgenden Grafik beziehen sich die umrandeten Aktivitäten, Produktphasen und Zielgruppen auf die vorliegende Dokumentation.

Beispiel: In der Produktphase "Engineering" kann die Zielgruppe "Programmierer" mit Hilfe dieser Dokumentation die Aktivität "Konfigurieren" ausführen.

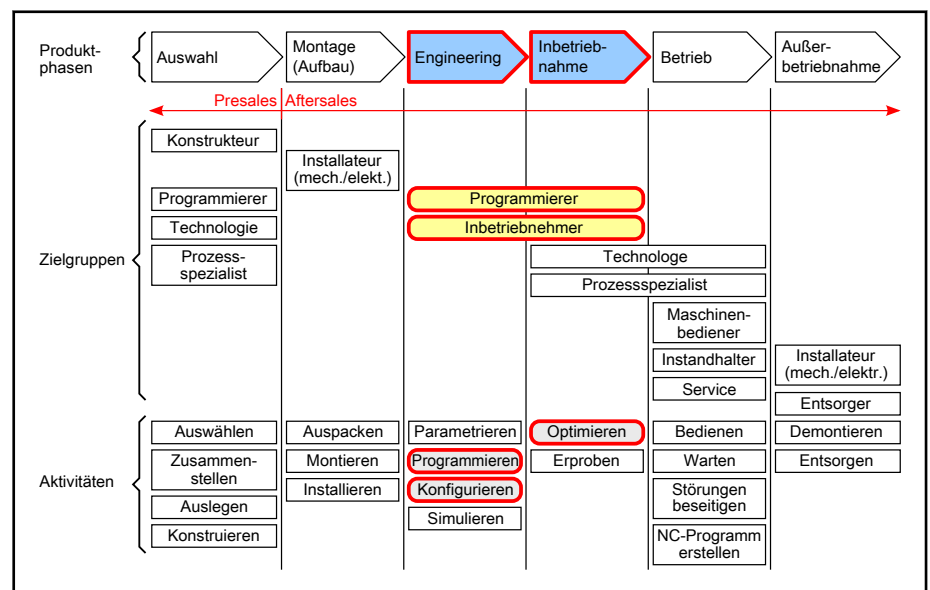


Abb. 1-1: Zuordnung der vorliegenden Dokumentation zu den Zielgruppen, Produktphasen und den Aktivitäten der Zielgruppe

Zielgruppe

Diese Dokumentation richtet sich an OPC-Client-Programmierer und WinStudio-Projektierer. Es werden die OPC-Tags für das Logische Gerät MTX (LDX) beschrieben.

Diese Dokumentation unterstützt den Anwender in den Phasen

- Engineering und
- Inbetriebnahme

1.2 Zweck

Diese Dokumentation dient OPC-Client-Programmierern und WinStudio-Projektierern zur Beschreibung der OPC-Tags für das Logische Gerät MTX (LDX).

Hauptbestandteil der Tag-Syntax ist das sogenannte **Item**, über welches das angeforderte Datum mittels Name und individuellen Parametern spezifiziert wird. Die nachfolgende Dokumentation beschreibt Syntax und Aufbau der verfügbaren Items.

Zum Ermitteln eines gewünschten OPC-Tags siehe [Kap. 2.2 "OPC-Tag Aufbau" auf Seite 7](#).

1.3 Erforderliche und ergänzende Dokumentationen MTX

1.3.1 Auswählen/Zusammenstellen

Dokumentationstitel mit Typenschlüssel und Materialnummer

MTX 15VRS Systembeschreibung DOK-MTX***-SYS*DES*V15-PRRS-DE-P, R911397741 Diese Dokumentation beschreibt das Steuerungssystem MTX. Beschrieben werden die Ausführungen, Technischen Daten, Schnittstellen und die Konfiguration der Steuerungsbaugruppen.
IndraMotion MTX 14VRS SafeLogic Systemübersicht DOK-MTX***-SL**SYS*V14-PRRS-DE-P, R911341697 Diese Dokumentation beschreibt den Einsatz der Sicherheitssteuerung SafeLogic im System MTX.
IndraControl XM42 Steuerungen DOK-CONTRL-XM4X*CTRL**-ITRS-DE-P, R911345565 Diese Dokumentation dient der Beschreibung der Steuerungen IndraControl XM42.

RS entsprechender Ausgabestand
Tab. 1-2: MTX-Dokumentationsübersicht - Auswählen/Zusammenstellen

1.3.2 Konfigurieren

Dokumentationstitel mit Typenschlüssel und Materialnummer

IndraMotion MTX 14VRS Maschinenparameter DOK-MTX***-MA*PAR**V14-RERS-DE-P, R911342629 Diese Dokumentation beschreibt den Aufbau und die Anpassung der verfügbaren Parameter der MTX. Ebenfalls beschrieben werden Funktion und Bedienung des NC-Konfigurators.
IndraMotion MTX 14VRS SPS-Interface DOK-MTX***-PLC*INT*V14-PRRS-DE-P, R911342621 Diese Dokumentation beschreibt die Interface-Signale und die Programmbausteine für die integrierte SPS.
IndraMotion MTX 14VRS Funktionsbeschreibung - Grundlagen DOK-MTX***-NC*F*BA*V14-RERS-DE-P, R911342625 Diese Dokumentation beschreibt die Basisfunktionen der MTX. Erläutert werden grundlegende Inbetriebnahmeschritte und die Funktionen der Steuerung als Beschreibung und Handlungsanweisung.
IndraMotion MTX 15VRS Funktionsbeschreibung - Erweiterung DOK-MTX***-NC*F*EX*V15-RERS-DE-P, R911393315 Diese Dokumentation beschreibt die erweiterten Funktionen der MTX. Erläutert werden grundlegende Inbetriebnahmeschritte und die Funktionen der Steuerung als Beschreibung und Handlungsanweisung.
MTX 15VRS Funktionsbeschreibung - Sonderfunktionen DOK-MTX***-NC*F*SP*V15-RERS-DE-P, R911393308 Diese Dokumentation beschreibt die Sonderfunktionen der MTX. Erläutert werden grundlegende Inbetriebnahmeschritte und die Funktionen der Steuerung als Beschreibung und Handlungsanweisung.

IndraMotion MTX Freiformflächenbearbeitung

DOK-MTX***-FREEFORM***-APRS-DE-P, R911341434

Diese Dokumentation beschreibt die Vorgehensweise bei der Freiformflächenbearbeitung mit der MTX-Steuerung. CNC-Programme, die von einem CAD/CAM-System generiert werden, dienen als Grundlage für den gesamten Bearbeitungsprozess. Es wird ein Überblick über folgende Themen gegeben: Beschreibung der MTX- bzw. CNC-Funktionen zur Freiformflächenbearbeitung, Vorgehensweise bei der NC-Parametrierung, Rahmenbedingungen bei der Erzeugung von CNC-Programmen mit CAM-Software.

MTX Konvertierung von MTX-Projekten

DOK-MTX***-PROCONV****-PRRS-DE-P, R911342483

Diese Dokumentation unterstützt bei der Konvertierung von MTX 1.x Projekten nach MTX 2G. Es wird die Vorgehensweise beschrieben und auf mögliche Schwierigkeiten bei der Konvertierung hingewiesen.

RS

entsprechender Ausgabestand

Tab. 1-3:

MTX-Dokumentationsübersicht - Konfigurieren

1.3.3 In Betrieb nehmen

Dokumentationstitel mit Typenschlüssel und Materialnummer**MTX 15VRS Inbetriebnahme**

DOK-MTX***-STARTUP*V15-CORS-DE-P, R911393280

Diese Dokumentation beschreibt die Inbetriebnahme der Steuerung MTX. Neben einer Gesamtübersicht werden die Inbetriebnahme und Konfiguration der Achsen und Bedienoberfläche und der SPS-Daten erläutert.

IndraWorks 14VRS Basis-Bibliotheken IndraLogic 2G

DOK-IL*2G*-BASLIB**V14-LIRS-DE-P, R911343918

Das vorliegende Handbuch beschreibt die systemübergreifenden SPS-Bibliotheken.

IndraWorks 15VRS Feldbusse

DOK-IWORKS-FB*****V15-APRS-DE-P, R911393282

Die Dokumentation dient der Beschreibung der von den Systemen MLC und MTX unterstützten Feldbus- und lokalen Peripherieanschlüssen. Der Schwerpunkt dieser Dokumentation liegt in der Konfiguration, Parametrierung, Inbetriebnahme und Diagnose der verschiedenen Peripherieanschlüssen. Sie ist die Grundlage für die Online-Hilfe.

IndraWorks 14VRS WinStudio 7.4

DOK-IWORKS-WINSTUD*V14-APRS-DE-P, R911341584

Dieses "Benutzerhandbuch und Technische Referenz" soll Sie dabei unterstützen, mit Ihrer Software "Rexroth WinStudio"™ optimale Ergebnisse zu erzielen. Das Dokument bietet technische Informationen und Schritt-für-Schritt-Anleitungen zum Anlegen von webfähigen HMI/SCADA-Programmen.

IndraWorks 15VRS Software-Installation

DOK-IWORKS-SOFTINS*V15-CORS-DE-P, R911393449

Die Dokumentation beschreibt den Installationsprozess von IndraWorks.

IndraWorks 15VRS Engineering

DOK-IWORKS-ENGINEE*V15-APRS-DE-P, R911393302

Diese Dokumentation beschreibt die Verwendung von IndraWorks, in das sich die Rexroth Engineering-Tools integrieren. Beschrieben werden das Arbeiten mit IndraWorks und die Bedienung der Oszilloskop-Funktion.

Zu dieser Dokumentation

IndraWorks 15VRS SPS-Programmiersystem IndraLogic 2G

DOK-IWORKS-IL2GPRO*V15-APRS-DE-P, R911396134

Diese Dokumentation beschreibt das SPS-Programmierzug IndraLogic 2G und seine Verwendung. Beschrieben werden die Basisverwendung, Erste Schritte, die Visualisierung, Menüpunkte und die Editoren.

IndraWorks 14VRS HMI

DOK-IWORKS-HMI*****V14-APRS-DE-P, R911343567

Diese Dokumentation beschreibt die Funktionen, die Konfiguration und die Bedienung der Bedienoberflächen IndraWorks HMI Engineering und IndraWorks HMI Operation.

RS

entsprechender Ausgabestand

Tab. 1-4:

MTX-Dokumentationsübersicht - In Betrieb nehmen

1.3.4 Bedienen

Dokumentationstitel mit Typenschlüssel und Materialnummer**Rexroth IndraMotion MTX 12VRS Satzvorlauf**

DOK-MTX***-BLK*RUN*V12-APRS-DE-P, R911334378

Die Dokumentation soll den Maschinenhersteller in die Lage versetzen, die Funktion "Satzvorlauf" an seiner Maschine für den Endanwender einzurichten.

IndraMotion MTX 15VRS Programmierhandbuch

DOK-MTX***-NC**PRO*V15-RERS-DE-P, R911393317

Das vorliegende Handbuch beschreibt die Standard-Programmierung der Steuerung MTX. Beschrieben werden neben den Grundlagen der NC-Programmierung sowohl die Anwendung der NC-Funktionen nach DIN 66025, als auch der NC-Funktionen mit Hochsprachensyntax und CPL-Funktionen.

IndraMotion MTX 15VRS Standard-NC-Bedienung

DOK-MTX***-NC*OP***V15-APRS-DE-P, R911393312

Diese Dokumentation beschreibt die Bedienung der Standard-Bedienoberfläche der NC-Steuerung MTX. Beschrieben werden die Bedienung der Oberfläche, die NC-Programmerstellung und die Werkzeugverwaltung.

MTX 15VRS Multitouch

DOK-MTX***-MULTI***V15-APRS-DE-P, R911393310

In dieser Dokumentation wird die Bedienung der Multitouch-Bedienoberfläche der NC-Steuerung MTX beschrieben.

MTX 15VRS Standard-NC-Zyklen

DOK-MTX***-NC*CYC**V15-PRRS-DE-P, R911394939

Diese Dokumentation beschreibt für die Steuerung MTX die Anwendung der Standard-Zyklen der verschiedenen Technologien.

MTX 15VRS NC-Simulation

DOK-MTX***-NC*SIM**V15-APRS-DE-P, R911393272

Diese Dokumentation beschreibt für die Steuerung MTX die NC-Simulation.

IndraMotion MTX 15VRS Messfunktionen

DOK-MTX***-MES*FUN*V15-APRS-DE-P, R91194937

Diese Dokumentation beschreibt die Messzyklen der Steuerung MTX.

RS

entsprechender Ausgabestand

Tab. 1-5:

MTX-Dokumentationsübersicht - Bedienen

1.3.5 OEM-Entwicklung

Dokumentationstitel mit Typenschlüssel und Materialnummer

Rexroth IndraMotion MTX 13VRS Automation Interface DOK-MTX***-AUT*INT*V13-APRS-DE-P, R911337273 Diese Dokumentation beschreibt den scriptbasierten Zugriff auf IndraWorks-Projektdaten über die Schnittstelle des Automation Interfaces. Beschrieben werden die verschiedenen Objekte, einschließlich Code-Beispielen. Eine Beschreibung des Automation Builders schließt sich an.
MTX 15VRS OPC-Kommunikation DOK-MTX***-OPC*COM*V15-PRRS-DE-P, R911399271 Diese Dokumentation beschreibt die Syntax und den Aufbau der Items für die Kommunikation mit den Bosch Rexroth-Geräten.
IndraWorks OPC UA Kommunikation DOK-IWORKS-OPC*UA*****-APRS-DE-P, R911379308 Diese Dokumentation beschreibt die OPC UA Kommunikation der Steuerungssysteme MLC und MTX.

RS entsprechender Ausgabestand
Tab. 1-6: MTX-Dokumentationsübersicht - OEM-Entwicklung

1.3.6 AddOns

Dokumentationstitel mit Typenschlüssel und Materialnummer

Rexroth IndraMotion MTX 11VRS Aktionsrekorder DOK-MTX***-ACR*****V11-APxx-DE-P, R911329942 Diese Dokumentation beschreibt den MTX Aktionsrekorder. Beschrieben werden sowohl die Installation und die Inbetriebnahme, einschließlich Schnittstellensignale, als auch die Anwendung und die Bedienung.
Rexroth IndraMotion MTX 14VRS efficiency workbench MTX cta, MTX ega DOK-MTX***-EWB*****V14-APRS-DE-P, R911343915 Diese Dokumentation beschreibt die Funktionsweise und den Einsatzbereich der Analysetools IndraMotion MTX cta und IndraMotion MTX ega.
Rexroth IndraMotion MTX Remote Condition Monitoring DOK-MTX***-RCM*****V01-APRS-DE-P, R911334382 Diese Dokumentation beschreibt die Bedienung des Remote Condition Monitoring Systems.
Rexroth IndraMotion MTX visIREC Anwenderdokumentation DOK-MTX***-VISIREC*V01-APRS-DE-P, R911344241 Diese Dokumentation beschreibt das Analysewerkzeug visIREC, das zur Optimierung des Freiformflächenbearbeitungsprozesses eingesetzt werden kann. 2D- bzw. 4D-Anzeige der bahnbezogenen Daten. 2D- bzw. 4D-Anzeige der koordinatenbezogenen Daten. Analyse von kritischen Bereichen (Bahn- und Orientierungsabweichung). Vergleich zwischen den programmierten und den interpolierten NC-Sätzen.

xx / RS entsprechender Ausgabestand
Tab. 1-7: MTX-Dokumentationsübersicht - AddOns

1.4 Darstellung von Informationen

1.4.1 Sicherheitshinweise

Die Sicherheitshinweise, soweit sie in der vorliegenden Anwendungsdokumentation vorhanden sind, beinhalten bestimmte Signalwörter (Gefahr, Warnung, Vorsicht, Hinweis) und ggf. eine Signalgrafik (nach ANSI Z535.6-2006).

Das Signalwort soll die Aufmerksamkeit auf den Sicherheitshinweis lenken und bezeichnet die Schwere der Gefährdung.

Die Signalgrafik (Warndreieck mit Ausrufezeichen), welche den Signalwörtern Gefahr, Warnung und Vorsicht vorangestellt wird, weist auf Gefährdungen für Personen hin.

GEFAHR

Bei Nichtbeachtung dieses Sicherheitshinweises **werden** Tod oder schwere Körpervletzung eintreten.

WARNUNG

Bei Nichtbeachtung dieses Sicherheitshinweises **können** Tod oder schwere Körpervletzung eintreten.

VORSICHT

Bei Nichtbeachtung dieses Sicherheitshinweises können mittelschwere oder leichte Körpervletzung eintreten.

HINWEIS

Bei Nichtbeachtung dieses Sicherheitshinweises können Sachschäden eintreten.

1.4.2 Verwendete Symbole

Hinweis Hinweise werden wie folgt dargestellt:



Das ist ein Hinweis für den Anwender.

Tipp Tipps werden wie folgt dargestellt:



Das ist ein Tipp für den Anwender.

1.4.3 Bezeichnungen und Abkürzungen

Begriff	Erklärung
NC	Numerical Control

Tab. 1-8: Verwendete Bezeichnungen bzw. Abkürzungen

1.4.4 Kundenfeedback

Anregungen, Wünsche oder Verbesserungen von unseren Kunden haben bei uns einen hohen Stellenwert. Senden Sie uns Ihre Anmerkungen zu den Dokumentationen per E-Mail an Feedback.Documentation@boschrexroth.de. Sie können direkt im elektronischen PDF-Dokument Kommentare einfügen und uns die PDF-Datei zusenden.

2 Einleitung

2.1 Haftungsausschluss

WARNUNG

Keine Haftung für Schäden durch Kunden-OPC-Clients

Über OPC haben Sie uneingeschränkten Zugriff auf alle Geräte-Daten und die gesamte Geräte-Funktionalität.

Für Schäden und Datenverlust, die auf eine unsachgemäße Verwendung von OPC in "Third Party" OPC-Client Applikationen und WinStudio-Bedienbilder oder auf eine Umgehung bzw. Nichtbeachtung von gerätespezifischen Sicherheitsrichtlinien zurückzuführen sind, übernimmt Bosch Rexroth keine Haftung. Dies betrifft insbesondere das Schreiben gerätesensibler und sicherheitsrelevanter Daten.

2.2 OPC-Tag Aufbau

Ein OPC-Tag besitzt den unten stehenden Aufbau. Die Parameter werden durch Kommata voneinander getrennt.

Syntax `! [ 1 ],/[ 2 ]/[ 3 ],[ 4 ],[ 5 ] { [ 6 ],[ 7 ] }`

1. Namensraum
2. IP-Adresse oder Hostname
3. Geräte-Name oder Geräte-Guid
4. Item-Parameterliste
5. Parameter
6. Zeile
7. Spalte



Die Angabe von Namensraum, Hostname, Zeile und Spalte sind optional.

Wird keine Namensraum angegeben, so werden die Daten im Datentyp **VARIANT** geliefert.

Wird kein Hostname angegeben, so werden die Daten vom lokalen OPC-Server geliefert. Dies entspricht dem Wert "localhost" bzw. "127.0.0.1".

Das Gerät kann über den Namen (z. B. "IndraMotion_MTX_P60") oder über die Geräte-Guid beschrieben werden.

Art und Anzahl der Parameter sind für jedes Item unterschiedlich. Die genauen Parameter finden Sie im jeweiligen Item-Kapitel.

Wird keine {Zeile, Spalte} angegeben so wird die komplette Datenantwort lt. Dokumentation geliefert. Dies entspricht dem Wert "{-1,-1}".

Beispiele:

```
!BSTR, //machine01/IndraMotion_MTX_P60, NC.SystemData-
Watch_U, /Sys_Simulation[1], 1024 {0,0}
!BOOL, //machine02/IndraMotion_MTX_P60, Plc.PVL, .OpMo-
de.AUTO.Key {0,0}
```

```
!R8, //machine03/IndraMotion_MTX_P60, NC.Chan.AxisPosAcs, 1
{0, 7}

!R8, //machine04/IndraMotion_MTX_P60, NC.Chan.ActRotAxis-
sProgrammedSpeed, 1 {3, 7}
```

Namensraum Der Namensraum beschreibt den Datentyp des angeforderten Datums.
Folgende Namensräume stehen zu Auswahl:

COM Datentyp	Namensraum
VT_I1	!I1
VT_I2	!I2
VT_I4	!I4
VT_I8	!I8
VT_INT	!INT
VT_R4	!R4
VT_R8	!R8
VT_UI1	!UI1
VT_UI2	!UI2
VT_UI4	!UI4
VT_UI8	!UI8
VT_UINT	!UINT
VT_BOOL	!BOOL
VT_BSTR	!BSTR
VT_DATE	!DATE
VT_CY	!CY

Tab. 2-1: Namensraum



Der OPC-Server konvertiert u. U. das ursprüngliche Datum (z. B. UI1) in den im Namenraum angegebenen Datentyp (z. B. I4) unter Berücksichtigung der OPC Konvention.

Es sollte immer eine Namensraum-Angabe verwendet werden. Fehlt die Angabe des Namensraumes, so wird das angeforderte Datum im Datentyp **VARIANT** geliefert.

{Zeile, Spalte} Eine Erweiterung "{Zeile, Spalte}" betrifft die Auswahl eines Teilergebnisses aus einem komplexen Ergebnis, welches sich über mehrere Zeilen und Spalten (Matrix) erstrecken kann.

Erweitert man den OPC-Tag um "{Zeile, Spalte}", so kann gezielt ein Teilergebnis aus dem Gesamtergebnis gelesen werden. Solche Tags werden innerhalb des OPC-Servers gesammelt und es wird ein optimierter Lesezugriff durchgeführt.

Folgende Formate ist zulässig:

!R8,//machine01/IndraMotion_MTX_P60,NC.Chan.AxisPosAcs,1	vollständige Tabelle
!R8,//machine01/IndraMotion_MTX_P60,NC.Chan.AxisPosAcs,1 {0,0}	Ergebnis aus Zeile 0 und Spalte 0
!R8,//machine01/IndraMotion_MTX_P60,NC.Chan.AxisPosAcs,1 {0,7}	Ergebnis aus Zeile 0 und Spalte 7
!R8,//machine01/IndraMotion_MTX_P60,NC.Chan.AxisPosAcs,1 {0, -1}	Unzulässig. Ganze Zeile wird nicht unterstützt
!R8,//machine01/IndraMotion_MTX_P60,NC.Chan.AxisPosAcs,1 {-1, 7}	Unzulässig. Ganze Spalte wird nicht unterstützt

Tab. 2-2: {Zeile, Spalte}

2.3 OPC-Test-Client

Zum Testen der OPC-Kommunikation bzw. der Datenantwort eines gewünschten OPC-Tags, existiert die OPC-Client-Applikation "**OPC.TestClient.exe**". Weitere Informationen hierzu entnehmen Sie bitte der OEM SDK Dokumentation.



Die Applikation "OPC.TestClient.exe" befindet sich im IndraWorks-Installationsverzeichnis (z. B. "C:\Program Files \Rexroth\IndraWorks")

3 Wichtige Gebrauchshinweise

3.1 Bestimmungsgemäßer Gebrauch

3.1.1 Einführung

Die Produkte von Bosch Rexroth werden nach dem jeweiligen Stand der Technik entwickelt und gefertigt. Vor ihrer Auslieferung werden sie auf ihren betriebssicheren Zustand hin überprüft.

Die Produkte dürfen nur bestimmungsgemäß eingesetzt werden. Wenn sie nicht bestimmungsgemäß eingesetzt werden, dann können Situationen entstehen, die Sach- und Personenschaden nach sich ziehen.



Für Schäden bei nicht-bestimmungsgemäßigem Gebrauch der Produkte leistet Bosch Rexroth als Hersteller keinerlei Gewährleistung, Haftung oder Schadensersatz; die Risiken bei nicht-bestimmungsgemäßigem Gebrauch der Produkte liegen allein beim Anwender.

Bevor Sie die Produkte der Firma Bosch Rexroth einsetzen, müssen die folgenden Voraussetzungen erfüllt sein, um einen bestimmungsgemäßen Gebrauch der Produkte zu gewährleisten:

- Jeder, der in irgendeiner Weise mit einem unserer Produkte umgeht, muss die entsprechenden Sicherheitsvorschriften und den bestimmungsgemäßen Gebrauch lesen und verstehen.
- Sofern es sich bei den Produkten um Hardware handelt, müssen sie in ihrem Originalzustand belassen werden; d. h. es dürfen keine baulichen Veränderungen an ihnen vorgenommen werden. Softwareprodukte dürfen nicht dekompiert werden und ihre Quellcodes dürfen nicht verändert werden.
- Beschädigte oder fehlerhafte Produkte dürfen nicht eingebaut oder in Betrieb genommen werden.
- Es muss gewährleistet sein, dass die Produkte entsprechend den in der Dokumentation genannten Vorschriften installiert sind.



Die der Steuerung vorliegenden oder vom Benutzer eingegebenen bzw. eingelesenen Daten sind vor der Übernahme/Aktivierung auf ihre Richtigkeit zu prüfen, um ungewollte Bewegungen der Achsen auszuschließen. Im Einzelnen kann es sich um folgende ungültige oder veraltete Daten handeln:

- Teileprogramme
- NPV-Tabellen
- Korrekturtabellen
- Werkzeugtabellen
- Permanente CPL-Variablen
- Remanente SPS-Daten
- Permanente Systemdaten

3.1.2 Einsatz- und Anwendungsbereiche

Die Rexroth MTX-Steuerung dient zum

- Programmieren von Kontur und Bearbeitungstechnologie (Bahnvorschub, Spindeldrehzahl, Werkzeugwechsel) eines Werkstückes.
- Führen eines Bearbeitungswerkzeuges entlang einer programmierten Bahn.

Vorschubantriebe, Spindeln und Hilfsachsen einer Werkzeugmaschine werden per Sercos Schnittstelle angesteuert.



Zusätzlich werden E/A-Komponenten für die integrierte SPS benötigt, die – in Kombination mit der eigentlichen CNC – den Bearbeitungsprozess ganzheitlich steuert und auch in sicherheitstechnischer Hinsicht überwacht.

Der Betrieb darf nur in den ausdrücklich angegebenen Konfigurationen und Kombinationen der Hardware-Komponenten und mit der in den jeweiligen Dokumentationen und den Funktionsbeschreibungen angegebenen und spezifizierten Soft- und Firmware erfolgen.

Das Rexroth CNC-System MTX bietet die perfekte CNC-Systemlösung für Zerspanen und Umformen für die folgenden Technologien:

- Drehen
- Fräsen
- Bohren
- Schleifen
- Biegen
- Nibbeln
- Stanzen
- Formschneiden
- Handhaben

3.2 Nicht-bestimmungsgemäßer Gebrauch

Die Verwendung des Rexroth CNC-Systems MTX außerhalb der vorgenannten Anwendungsgebiete oder unter anderen als den in der Dokumentation beschriebenen Betriebsbedingungen und angegebenen technischen Daten gilt als "nicht-bestimmungsgemäß".

Das Rexroth CNC-System MTX darf nicht eingesetzt werden, wenn sie ...

- Betriebsbedingungen ausgesetzt wird, die die vorgeschriebenen Umgebungsbedingungen nicht erfüllen. Untersagt sind z. B. der Betrieb unter Wasser, unter extremen Temperaturschwankungen oder extremen Maximaltemperaturen.
- Außerdem darf das Rexroth CNC-System MTX nicht bei Anwendungen eingesetzt werden, die von Bosch Rexroth nicht ausdrücklich freigegeben sind. Beachten Sie hierzu bitte unbedingt die Aussagen in den allgemeinen Sicherheitshinweisen!
- Das Rexroth CNC-System MTX darf nicht an Anlagen oder Maschinen eingesetzt werden, die über eine ungesicherte Netzwerkverbindung mit dem Internet verbunden ist. Andernfalls besteht die Gefahr von Störungen bzw. einem Steuerungsausfall aufgrund unautorisierter Zugriffe.

4 Items

4.1 Allgemein

4.1.1 DeviceTypeTest2

Beschreibung	Prüft ob eine Kommunikationsverbindung zu einem bestimmten Gerätetyp besteht.
Anmerkung	Das Item DeviceTypeTest2 liest intern Parameter des Geräts aus (z. B. Firmwarekennung) anhand derer es einen Gerätetyp identifizieren kann.
Verfügbarkeit	Sercans, IndraDrive, MLC, MTX, IndraLogic
Item Syntax	DeviceTypeTest2
Read	Prüft, ob eine Kommunikationsverbindung zum Gerät aufgebaut werden kann und ob der Typ des konfigurierten Gerätes (logisches Gerät) mit dem Typ des angesprochenen Gerätes (Feldgerät) übereinstimmt.
Datenformat	DeviceTypeTest2

row, col	Typ	Beschreibung	Bsp.
0,0	bool	TypeValidated Flag zeigt an, ob Typ des konfigurierten Geräts und Typ des angesprochenen Gerät übereinstimmen und ob Verbindung aufgebaut werden konnte	False

4.2 Root

4.2.1 Übersicht

Funktionen des Kommunikationsprozesses werden über die "Root" Schnittstelle bereitgestellt. Über diese Schnittstelle werden Scanning-, Diagnose-, und Konfigurationsfunktionen zur Verfügung gestellt.

Root ist als Teil der Kommunikation immer verfügbar.

4.2.2 Root Items

AutoItemAccess

Beschreibung	Zugriffsmodus von "AutoItems" eines Gerätes. Für manche Geräte ist es möglich, sowohl eine Online, als auch eine Offline Schnittstelle anzusprechen. Ein gerätespezifischer Zustand erlaubt es, bestimmte Items zwischen der Online- und Offline-Schnittstelle umzuschalten.
Item Syntax	AutoItemAccess
Read	Lesen des "AutoItem" Zustands des Geräts.

AutoItemAccess, Read

row, col	Typ	Beschreibung	Bsp.
0,0	long	Zustand 1: SCPAccessOnlineWriteOffline (Default) 2: SCPAccessOffline 3: SCPAccessOnline	2
1,0	string	Zustand in Textform	"SCPAccessOffline"

Write Lesen des "AutoItem" Zustands des Geräts.

AutoItemAccess, Write

row, col	Typ	Beschreibung	Bsp.
0,0	long	Zustand 1: SCPAccessOnlineWriteOffline (Default) 2: SCPAccessOffline 3: SCPAccessOnline	2

CNF.AddComponent

Beschreibung Erzeugt eine neues Gerät oder einen Treiber.



Ein Gerät oder ein Treiber wird über benannte Konfigurationsparameter konfiguriert. Details hierzu entnehmen Sie der Dokumentation der Geräte- und Treiberkonfiguration.

Item Syntax **CNF.AddComponent,1**

Write Anlegen eines Gerätes oder eines Treibers.

Datenformat Pro Parameter (x) wird die Parameterbezeichnung (x,0) und der Parameterwert (x,1) als String übergeben.

CNF.AddComponent,1

row, col	Typ	Beschreibung	Bsp. Hinzufügen eines TCP/IP Treibers.
0,0	string	Parameterbezeichnung "Name"	"Name"
0,1	string	Parameterwert	"DR.IndraDrive"
1,0	string	Parameterbezeichnung "ProgId"	"ProgId"
1,1	string	Parameterwert	"IwSCPDriverEthernet.SCPDriverEthernet"
2,0	string	Parameterbezeichnung "Guid"	"GUID"
2,1	string	Parameterwert	"074cee90-0190-4cb4-9913-d7dc7d69ab34-On"
3,0	string	Parameterbezeichnung "IpAddress"	"IpAddress"
3,1	string	Parameterwert	"192.23.43.5"

row, col	Typ	Beschreibung	Bsp. Hinzufügen eines TCP/IP Treibers.
4.0	string	Parameterbezeichnung "Port"	"Port"
4.1	string	Parameterwert	"5003"

Tab. 4-1: Hinzufügen eines TCPIP Treibers

CNF.RemoveComponent

Beschreibung

Entfernt ein Gerät oder Treiber aus der Konfiguration.

Item Syntax

CNF.RemoveComponent

Write

Das Gerät oder der Treiber wird entfernt.

Die Items SDL, siehe [Kap. "SDL" auf Seite 19](#) bzw. GDL siehe [Kap. "GDL" auf Seite 18](#) werden das Gerät oder den Treiber nicht mehr anzeigen.



Das Gerät oder der Treiber wird aus der Liste entfernt. Das Gerät oder der Treiber bleibt im Speicher solange es noch in Benutzung ist.

CNF.RemoveComponent

row, col	Typ	Beschreibung	Bsp.
0,0	string	Die GUID oder der Name des Geräts oder des Treibers.	"87979723-228C-4ff4-ACE2-97C5A3F8B8C8"

CNF.CMD

Beschreibung

Speichern und Laden der Kommunikationskonfigurationen.

Der Default Ablageort der Konfiguration kann gelesen und geschrieben werden.

Dieses Item steht an der Root Schnittstelle zur Verfügung.



Die Dateinamen werden von dem jeweilig angesprochenen Kommunikationsprozess genutzt. Ein Remote erzeugter Prozess wird dementsprechend die Pfadnamen auf dem Remoterechner nutzen.

Relative Pfade sollten nicht benutzt werden.

Item Syntax

CNF.CMD

Read

Lesen des Default Ablageortes der Konfigurationsdatei.



Der Dateiname wird als benutzerspezifischer Eintrag aus der "Windows" Registry gelesen.

CNF.CMD (Read)

row, col	Typ	Beschreibung	Bsp.
0,0	string	Konstante "DefaultConfig-File"	DefaultConfigFile
1,0	string	Default Ablageort der Konfigurationsdatei.	"c:\scpcnfig.ini"

Write

Schreiben des Default Ablageortes der Konfigurationsdatei.



Der Dateiname wird als benutzerspezifischer Eintrag in die "Windows" Registry geschrieben.

CNF.CMD (Write)

row, col	Typ	Beschreibung	Bsp.
1,0	string	Default Ablageort der Konfigurationsdatei.	"c:\scpconfig.ini"

Call Laden und speichern einer SCPsServer Konfiguration.
"Save" und "Load".



Laden der Konfiguration löscht alle bereits aktivierten Geräte und Treiber.

Speichern überschreibt eine bereits bestehende Datei.

CNF.CMD (Call)

row, col	Typ	Beschreibung	Bsp. Speichern der Konfiguration in c:\scpconfig.ini
0,0	string	Kommando "Load" oder "Save"	"Save"
1,0	string	Optional: Default Ablageort wird aus der Registry ermittelt.	"c:\scpconfig.ini"

Beispiel: Load

CNF.CMD (Call)

row, col	Typ	Beschreibung	Bsp. Aktivieren der Konfiguration aus c:\scpconfig.ini
0,0	string	Kommando "Load"	"Load"
1,0	string	Optional: Default Ablageort wird aus der Registry ermittelt.	"c:\scpconfig.ini"

Beispiel: Save

Das Kommando mit den Argumentdaten Save sichert die aktuelle Konfiguration des SCPsServers inklusive der Geräte und Treiberkonfiguration. Eine bestehende Datei wird überschrieben.

CNF.CMD (Call)

row, col	Typ	Beschreibung	Bsp.
0,0	string	Kommando "Save"	"Save"
0,1	string	Optional: Default Ablageort wird aus der Registry ermittelt.	"c:\scpconfig.ini"

Beispiel (JavaScript)

```
var server;

function Connect()
{
```

```

if (server == null)
{
    server = new ActiveXObject("IwSCP.ScpServer");
    server.Connect("Root", "Loader");
}

function LoadSCP()
{
    var item;
    var data;
    Connect();
    item = server.CreateItem("CNF.CMD",0);
    data = item.SyncRead(1);
    ConfigFile.value = data.GetValue(0,1);
    data.SetValue(0,0,"Load");
    item.SyncCall(data);
}

function SaveSCP()
{
    var item;
    var data;
    Connect();
    item = server.CreateItem("CNF.CMD",0);
    data = item.SyncRead(1);
    data.SetValue(0,0,"Save");
    item.SyncCall(data);
}

```

ComPortList

Beschreibung	Liste der vorhandenen COM-Ports des PC's
Anmerkung	Über die seriellen Schnittstellen kann SCP mit den Geräten kommunizieren. Hierzu wird der serielle Treiber genutzt. Über dieses Item lässt sich die Liste der verfügbaren COM-Ports ermitteln.
Item Syntax	ComPortList
Read	Auslesen der aktuellen Liste der COM-Ports und deren Status.
Datenformat	Anzahl der Zeilen: Anzahl der COM-Ports(x)

ComPortList

row, col	Typ	Beschreibung	Bsp.
x,0	string	ComPort-Bezeichnung. Es wird der Windows Name für die COM Schnittstelle geliefert.	COM2
x,1	long	ComPort-Nummer. Die extrahierte Nummer aus der ComPort Bezeichnung.	2
x,2	bool	Die COM-Schnittstelle ist verfügbar. -->True Die COM-Schnittstelle ist belegt. --> False	false
x,3	string	Status-Text des Schnittstellentests. Ist bei verfügbarer Schnittstelle leer.	Zugriff verweigert

DTY

Beschreibung	Typ des Geräts bzw. des Treibers.
Item Syntax	DTY
Read	Lesen des Gerätetyps.

DTY

row, col	Typ	Beschreibung	Bsp.
0,0	string	Gerätetyp	MTX

GDL

- Beschreibung** Liefert die Liste aller Geräte und Treiber.
- Item Syntax** **GDL**
- Read** Auslesen der globalen Liste der Geräte und Treiber.
- Datenformat** Anzahl der Zeilen(x) entspricht der Anzahl von Geräten und Treibern.

GDL

row, col	Typ	Beschreibung	Bsp.
x,0	string	Gerätename	IndraMotionMTX
x,1	string	Gerätetyp	MTX
x,2	string	GUID	87979723-228C-4ff4-ACE2-97C5A3F8B8C83847

ListOfItems

- Beschreibung** Liste der Items, die an diesem Gerät oder an diesem Treiber zur Verfügung stehen.
- Anmerkungen** Die Liste wird aufgrund des Gerätetyps und der verfügbaren Komponenten geliefert.
- Die Liste enthält keine Beschreibung der möglichen Parameter. Die Parameterbeschreibung ist der Dokumentation zu entnehmen.



Das Item wird in der Regel zum Test eingesetzt.

- Item Syntax** **ListOfItems**, oder
LOI

- Read** Pro unterstütztem Item wird eine Zeile (x) geliefert.
- Die Itembeschreibung wird als Kurztext in Englisch geliefert.

ListOfItems

row, col	Typ	Beschreibung	Bsp.
x,0	string	Name des Items	"SPA2"
x,1	string	Kurzbeschreibung des Items (Englisch)	"Sercos parameter of the drive"

SCL

- Beschreibung** Geräte, bzw. Treiberinformationen.
- Anmerkung** Die Funktionalität eines Gerätes wird aus mehreren Komponenten zusammengesetzt. Jede dieser Komponenten stellt einen Satz Items zur Verfügung, auf den eine Applikation zugreifen kann.
- Item Syntax** **SCL**

Read Es wird eine Liste ausgegeben mit den Namen und Versionsinformationen der geladenen Komponenten. Das Gerät, bzw. der Treibertyp steht in der ersten Zeile (x=0).

Datenformat Pro Komponente werden 4 Element pro Zeile (x) geliefert.

SCL

row, col	Typ	Beschreibung	Bsp.
x,0	string	Komponentenname	"Root"
x,1	long	MajorVersion	6
x,2	long	MinorVersion	11
x,3	long	Build-Nummer	151

SDL

Beschreibung Liste der untergeordneten Geräte oder Treiber.

Item Syntax **SDL**

Read Es wird eine Liste mit Namen, Typ und GUID der untergeordneten Geräte ausgegeben.

Datenformat Pro untergeordnetem Gerät oder Treiber wird eine Zeile (x) mit 3 Einträgen geliefert.

SDL

row, col	Typ	Beschreibung	Bsp.
x,0	string	Gerätename. Der Name wird inklusive des ggf. übergeordneten Namens ausgegeben.	"Mlc01/Drive_0"
x,1	string	Gerätetyp, bzw. Treibertyp.	"SercosDrive"
x,2	string	Eindeutige Kennzeichnung des Geräts oder Treibers (GUID)	"3528e452-ed32-4a4a-8ee5-6ee77e1dbc6e5"

Über den Namen oder über die GUID kann eine direkte Verbindung aufgebaut werden.



Subgeräte werden über Gerätenamen "/" Subgerätenamen definiert.



Um über DCOM ein Gerät auf einem anderen Rechner anzusprechen, muss dem Gerätenamen der Rechnername oder die IP-Adresse vorangestellt werden.

Z. B.: "//<Hostname>/Mlc01/Drive_0" bzw. "//<IP-Addr>/Mlc01/Drive_0".

4.3 Firmware

4.3.1 Übersicht

Die Firmware-Management-Items unterstützen einen einheitlichen Zugriff auf die Firmware der einzelnen Geräte.

- Items zur Firmware-Erkennung**
- **Firmware.DeviceFirmware**
Auslesen der Firmwareerkennung
 - **Firmware.ComponentFirmware**
Auslesen der Firmwareerkennung der einzelnen Komponenten
 - **Firmware.ModuleFirmware**
Auslesen der Firmwaremodule einer Komponente
 - **Firmware.SubmoduleFirmware**
Auslesen der Kennungen der Submodule eines Moduls
- Items für Download Aktionen**
- **Firmware.CheckDownloadRequirements**
Prüfen der Erlaubnis zum Firmware-Download
 - **Firmware.Download**
Download von Firmware oder Firmwareteilen in ein Gerät
 - **Firmware.Reboot**
Reboot der Steuerung
 - **Firmware.ResumeConnection**
Erkennen der Verbindungsaufnahme nach Reboot
 - **Firmware.Status**
Lesen des Download-Status
 - **Firmware.StatusReset**
Zurückstellung der Download-Status-Informationen

Item	IndraDrive	IndraLogic	MLC	MTX
Firmware.DeviceFirmware Identification Layer 1	x	x	x	x
Firmware.ComponentFirmware Identification Layer 2	x			
Firmware.ModuleFirmware Identification Layer 3	x	x	x	x
Firmware.SubmoduleFirmware Identification Layer 4		x	x	

Tab. 4-2: Items zur Firmware-Erkennung

4.3.2 Firmware Items

Firmware.CheckDownloadRequirements

- Beschreibung** Prüft, ob ein Firmware-Download erlaubt ist. Ist der Download nicht erlaubt, wird der Grund dafür angegeben.
- Item Syntax** **Firmware.CheckDownloadRequirements**
- Aufruf** synchron / asynchron
- Zugriff** Read
- zyklisch (opt.)** ja

Datenformat **Firmware.CheckDownloadRequirements, read**

row, col	Typ	Beschreibung	Bsp.
0,0	bool	Download erlaubt bzw. nicht erlaubt	FALSE
0,1	long	Reason-ID, wenn Download nicht erlaubt ist 1 - PLC-Programm läuft 2 - Sercos Phase > 2	2



MLC

Firmware.CheckDownloadRequirements benutzt den Sercos Parameter C-0-0023 für den Firmware-Download.

IndraDrive

Firmware.CheckDownloadRequirements nutzt die SIS-Dienste für Beenden und Reboot.

Sercans

Der Download ist immer erlaubt.

Firmware.ComponentFirmware

Beschreibung Liefert die ID und Firmwarekennung der einzelnen Komponenten.

Item Syntax **Firmware.ComponentFirmware**

Aufruf synchron / asynchron

Zugriff Read

zyklisch (opt.) ja

Datenformat Anzahl der Zeilen: Anzahl der Komponenten

Firmware.ComponentFirmware

row, col	Typ	Beschreibung	Bsp.
x,0	long	Komponenten-ID	128
x,1	string	Komponenten-Firmwarekennung	MPH-02T02.01-MS

Beispiel IndraDrive mit 3 Komponenten. Die Komponente HPC besitzt die Firmware MPH-02T02.01-MS. Die Komponenten EMD besitzen die Firmware EMD-02T01.01-MS.

Firmware.ComponentFirmware

row, col	Typ	Beschreibung
0,0	long	128
0,1	string	MPH-02T02.01-MS
1,0	long	1
1,1	string	EMD-02T01.01-MS
2,0	long	3
2,1	string	EMD-02T01.01-MS

**IndraDrive, Sercans**

Wenn der Firmware-Download abgebrochen wurde, ist die Liste der Firmwarekomponenten möglicherweise nicht vollständig.

IndraLogic, MTX, MLC

Das Item Firmware.ComponentFirmware wird nicht benötigt.

Firmware.DeviceFirmware

Beschreibung	Liefert die Firmwarekennung des Gerätes.
Item Syntax	Firmware.DeviceFirmware
Aufruf	synchron/asynchron
Zugriff	Read
zyklisch (opt.)	nein
Datenformat	Firmware.DeviceFirmware, read

row, col	Typ	Beschreibung	Bsp.
0,0	string	Firmwarekennung des Gerätes	FWA-INDRV*-MPH-02T02.01-MS

**IndraDrive**

Die IndraDrive-Firmwareaktualisierung mit MMC wird erst nach einem Restart von IndraDrive ausgeführt. Erst danach ist die Firmwarekennung auslesbar.

MLC

Firmware.DeviceFirmware nutzt den Sercos Parameter C-0-0012 um die Geräte-Firmwarekennung zu bestimmen.

Firmware.Download

Beschreibung	Der Download von Firmware oder Firmwareteilen in ein Gerät wird ausgeführt (als zusammenhängender Prozess).
Item Syntax	Firmware.Download
Aufruf	asynchron
Zugriff	Call
zyklisch (opt.)	nein



Die Parameterübergabe erfolgt über ein SCPData Objekt in Form von Schlüssel-Werten Paaren, wobei die zu übergebenden Parameter vom jeweiligen Gerätetyp abhängig sind.

Das Item kehrt zurück, wenn nachdem nach einem Download wieder mit dem Gerät kommuniziert werden kann oder wenn ein Fehler aufgetreten ist.

Die folgende Tabelle bietet eine Übersicht über die möglichen Parameter

Parameterwerte

Schlüssel	Beschreibung	Typ	Bemerkung
Firmware	Absoluter Pfad der Firmware-Datei oder des Firmware-Verzeichnisses	string	
CheckDownloadRequirements	Vor dem Download prüfen, ob ein Download erlaubt ist (z. B. PLC gestoppt, Sercos Phase <=2)	bool	Optional Default = TRUE
Reboot	Nach dem Download ein Reboot durchführen	bool	Optional Default = TRUE
Component<n>	Komponenten-ID. Bestimmt die Gerätekomponekte, in die eine ComponentFirmware bzw. eine ModuleFirmware geladen wird	string	Optional
ComponentFirmware<n>	Komponenten-Firmware-ID. Gibt die Komponenten-Firmware an, die in die Komponente<n> geladen wird oder gibt die Komponenten-Firmware an, in die ModuleFirmware<n> geladen werden soll	string	Optional
ModuleFirmware<n>	Modul-Firmware-ID. Gibt die ModuleFirmware an, die in ComponentFirmware<n> der Component<n> geladen werden soll	string	Optional
UserLevel	1: Standard Benutzer 2: Service 3: Entwickler	long	Optional Default = 1
Date	Download Datum Format: "YYYY.MM.DD" Nur für IndraDrive	string	Optional
Time	Download Zeit Format: "HH:MM:SS" Nur für IndraDrive	string	Optional
Comment	Download Kommentar Nur für IndraDrive	string	Optional

Beispiel Firmware.Download

row, col	Typ	Beschreibung	Bsp.
0,0	string	Schlüssel: "Firmware"	Firmware
0,1	string	Wert: Absoluter Pfad der Firmware-Datei oder des Firmware-Verzeichnisses	C:\Programme \...\Firmware \FWAMPH02T0 2_02.IBF

row, col	Typ	Beschreibung	Bsp.
1,0	string	Schlüssel: "CheckDownloadRequirements"	CheckDownload Requirements
1,1	bool	Value optional, Default = TRUE festsetzen, ob Bedingungen für Download überprüft werden (z. B. PLC gestoppt, Sercos Phase <=2)	TRUE

Ergebnis leer



IndraDrive

- Bei User-Level "1" wird der Download nur durchgeführt, wenn für alle Komponenten Firmware-Module vorhanden sind.
- Bei gesteckter MMC-Card ist der FWA-Download nur mit kompatiblen IBF-Dateien mit Kompatibilitätsliste möglich.
- Der FirmwareComponent-Download erfordert Component<n> und ComponentFirmware<n>.
- Der Modul-Download erfordert Component<n>, ComponentFirmware<n> und ModulFirmware<n>.

MTX

Der Subsystem-Download erfordert die entsprechende Modul-Firmware<n>.

MLC

In Datei "FTP_cmds.txt", Beschreibung eines Downloads, können nur folgende Befehle benutzt werden: update, bin, asc, put <file>, put <file1> <file2>, mkd, del, bye.

Fehlercodes leer

Das Item nutzt den SCP ErrNACK6 (106) zum Anzeigen eines Fehlers.

Fehlercodes

Detaillierte Fehlernummer <long>	Beschreibung
0	kein Fehler
1	Firmware-Datei oder Firmware-Verzeichnis fehlt
2	Firmware-Datei ist defekt (Formatfehler) oder unvollständig
3	Im aktuellen Gerätestatus ist der Download nicht erlaubt
4	Gerätefehler (Hardware- oder Gerätefirmware-Problem im Zielgerät)
5	Fehler bei der Download-Verifikation
6	Fehler beim Schreiben

Tab. 4-3: Item Firmware.Download Fehlercodes

Firmware.ModuleFirmware

Beschreibung	Liefert die Firmwaremodule einer Komponente.
Anmerkung	Die Angabe einer Komponenten-ID ist optional, da nicht jedes Gerät Komponenten besitzt. Logische Geräte ohne Komponenten unterstützen dieses Firmware-Item nicht.
Item Syntax	<i>Firmware.ModuleFirmware,(1)</i> 1. Komponenten-ID
Aufruf	asynchron
Zugriff	Read
zyklisch (opt.)	nein
Datenformat	Anzahl der Zeilen: Anzahl der Module

Firmware.ModuleFirmware

row, col	Typ	Beschreibung	Bsp.
x,0	string	Modul-ID	1
x,1	string	Modul-Firmwarekennung	

Beispiel **Firmware.ModuleFirmware,128**

row, col	Typ	Beschreibung	
0,0	string	1. Modul-ID	1
0,1	string	1. Modul-Firmwarekennung	MPH-02T02.01-MS - Applikation
1,0	string	2. Modul-ID	2
1,1	string	2. Modul-Firmwarekennung	MPH-02T02.01-MS - Loader



IndraDrive, Sercans

Wenn der Firmware-Download abgebrochen wurde, dann ist die Liste der Firmwaremodule möglicherweise nicht vollständig.

Firmware.Reboot

Beschreibung	Durch Schreiben dieses Items wird ein Reboot der Steuerung ausgelöst. Das Item besitzt keine Parameter.
Item Syntax	Firmware.Reboot
Aufruf	synchron / asynchron
Zugriff	Write
zyklisch (opt.)	nein



MLC

Firmware.Reboot nutzt den Sercos Parameter C-0-1020 für den Reboot.

Firmware.ResumeConnection

Beschreibung	Dieses Item wird nach erfolgtem Firmwaredownload eingesetzt um zu erkennen, wann wieder eine Verbindung zum Gerät hergestellt wurde. Das Item besitzt keine Parameter.
Item Syntax	Firmware.ResumeConnection
Aufruf	synchron / asynchron
Zugriff	Write
	Wartet bis eine Verbindung zu Gerät hergestellt wurde bzw. bis zum Ablauf eines Timeouts.
zyklisch (opt.)	ja



MLC

Firmware.ResumeConnection liest wiederholt den Sercos Parameter C-0-0012, um eine Wiederverbindung zu erkennen.

Firmware.Status

Beschreibung	Der Download-Status wird gelesen.
Item Syntax	Firmware.Status
Aufruf	synchron / asynchron
Zugriff	Read
zyklisch (opt.)	ja
Datenformat	Firmware.Status

row, col	Typ	Beschreibung	Bsp. Status IndraDrive
0,0	long	Status-ID 0 - idle 1 - downloading 2 - downloaded 3 - canceling 4 - canceled 5 - error	1
1,0	string	aktuelle Geräte-Firmware-Konfiguration	FWA-INDRV*-MPH-02T02.01-MS
2,0	string	aktuelle Komponenten-ID	HPC01.1
3,0	string	aktuelle Komponenten-Firmware-ID	MPH-02T02.01-MS
4,0	string	aktuelle Module-Firmware-ID	MPH-02T02.01-MS - Loader

row, col	Typ	Beschreibung	Bsp. Status IndraDrive
5,0	long	SIS Download Progress-ID 0 - idle (Download nicht gestartet, oder schon beendet) 1 - check download requirements 2 - get configured baudrate of device 3 - set communication parameters at loader default parameters 4 - read backup medium configuration state 5 - identify device configuration 6 - get module information 7 - get component mode 8 - shutdown 9 - set communication parameters at maximum transfer rate 10 - delete firmware 11 - write file info (MMC) 12 - write file (MMC) 13 - flash firmware 14 - calculate checksum 15 - reboot 16 - waiting until connection is restored 17 - restore communication parameter 18 - get device baudrate 19 - get max download baudrate FTP Download Progress-ID 0 - idle (Download nicht gestartet, oder schon beendet) 1 - extract firmware file 2 - check download requirements 3 - get communication parameters 4 - downloading 5 - reboot 6 - resume connection	9
6,0	long [0..100]	Fortschritt in Progress- ID (in Prozent) FTP: Fortschritt bezieht sich auf den gesamten Downloadvorgang SIS: Fortschritt bezieht sich auf das jeweilige Modul	43
7,0	long	Anzahl der Module	2
8,0	long	aktuelle Modulnummer	1

row, col	Typ	Beschreibung	Bsp. Status IndraDrive
9,0	long	Fehlercode	0
10,0	long	detaillierter Fehlercode	0
11,0	string	zusätzliche Fehlerinformation (z. B. Dateiname)	

**MLC, IndraLogic, MTX**

Die aktuelle Komponenten-ID und die aktuelle Komponenten-Firmware-Kennung sind zur Zeit leer.

Firmware.StatusReset

Beschreibung	Zurücksetzen der Download-Status-Informationen. Das Item besitzt keine Parameter.
Item Syntax	Firmware.StatusReset
Aufruf	synchron / asynchron
Zugriff	Write
zyklisch (opt.)	nein

Firmware.SubmoduleFirmware

Beschreibung	Liefert die Kennungen der Submodule, die im Modul einer Komponente enthalten sind.
Item Syntax	<i>Firmware.SubmoduleFirmware,(1),(2)</i> 1. Modul-ID 2. Komponenten-ID (optional)
Aufruf	asynchron
Zugriff	Read
zyklisch (opt.)	nein
Datenformat	Anzahl der Zeilen: Anzahl der Submodule

Firmware.SubmoduleFirmware, read

row, col	Typ	Beschreibung	Bsp.
x,0	string	Submodul-ID	1
x,1	string	Submodul-Firmwarekennung	

Beispiel Die Firmware besitzt zwei Submodule 3S und Motion.

Firmware.SubmoduleFirmware,st

row, col	Typ	Beschreibung	
0,0	string	1. Submodul - ID	3S
0,1	string	1. Submodul - Firmwarekennung	IL-03V01-01-MS
1,0	string	2. Submodul - ID	Motion
1,1	string	2. Submodul - Firmwarekennung	ML-03V01-01-MS

**IndraDrive, MTX, Sercans**

Das Item Firmware.SubmodulFirmware wird nicht benötigt.

NC.LanguagesDownload

Beschreibung Das Item dient der Übertragung, dem Löschen und der Anzeige von Sprachtexten. Die Übertragungsrichtung der Sprachdateien geht nur vom PC zur NC. Das Löschen von Sprachdateien erfolgt nur auf der NC im Verzeichnis /usrfep/text.

Ein lesender Zugriff auf das Item zeigt die Sprachdateien auf dem Installationsverzeichnis des PC sowie der NC.

Anmerkung Die im Installationsverzeichnis abgelegten Textdateien können in den Formaten ANSI, UNICODE oder UTF8 abgelegt sein. Textdateien mit den Formaten ANSI und UNICODE werden vor der Übertragung in das Format UTF8 konvertiert. Die Übertragung der UTF8 Dateien erfolgt ohne Konvertierung.

Die Länderkennzahl beim Schreiben ist immer dreistellig anzugeben.

Ein Schreib bzw. ein Löschvorgang ist im Gegensatz zu einem Lesevorgang sehr zeitintensiv. Das Schreiben eines kompletten Satzes dauert ca. 30 Sekunden. Das Löschen eines Satzes ca. 15 Sekunden. Aus diesem Grund sollte darauf geachtet werden, dass die Aufrufe dieses Items beim Schreiben oder Löschen asynchron ausgeführt werden.

Um ein beschleunigtes Arbeiten mit den Textdateien zu erlauben, wird zu jeder übertragenen Sprachdatei auf der NC automatisch eine Indexdatei generiert. Bei der Indexdatei handelt es sich um eine binäre Datei. Über eine null-basierte Textnummer gelangt man zu einem Dateizeiger, der direkt auf den entsprechenden Text in der entsprechenden Sprachdatei verweist. Die Indexdatei trägt die um das Präfix „idx“ erweiterte Bezeichnung der Textdatei. Als Beispiel lautet die Bezeichnung der Indexdatei für die deutschen Systemtexte „idxsystemte.049“.

Item Syntax NC.LanguagesDownload,(1),(2)

1. Pfad zu System- und OEM-Texten auf dem PC (optional)

2. Pfad zu Pfi-Texten auf dem PC (optional)



Wird das Item ohne Pfadangabe erzeugt, so ermittelt das Item die Pfade, in denen die zu installierenden Sprachdateien liegen, automatisch aus der Registry. Der Eintrag dazu lautet : (HKEY_LOCAL_MACHINE\SOFTWARE\BOSCH\PC-PANEL\CurrentVersion\TextDir)

Einschränkung: Z.Zt. keine Verarbeitung von Pfi-Texten!

Aufruf asynchron

Zugriff Read / Write

zyklisch (opt.) nein

Datenformat NC.LanguagesDownload, read

row, col	Typ	Beschreibung	Bsp.: Lesen aller Sprachdateien aus dem Installationsverzeichnis des PC und der NC
0,0	string	alle Sprachdateien im Installationsverzeichnis	Systexte.049 28.01.2008 09:56:14 Systexte.044 24.01.2008 18:49:06 Oemtexte.049 24.01.2008 19:03:02 Oemtexte.044 24.01.2008 19:03:02
1,0	string	alle Sprachdateien innerhalb der NC	Systexte.049 28.01.2008 09:56:14 Oemtexte.049 24.01.2008 19:03:02

NC.LanguagesDownload, write

row, col	Typ	Beschreibung	Bsp.: Übertragung von Sprachdateien aus dem Installationsverzeichnis des PC in das Installationsverzeichnis der NC
0,0	string	Angabe der Sprachdatei(en) - bei mehreren Dateien sind diese durch Leerzeichen zu trennen im Beispiel: 044 = englische Sprachdatei 049 = deutsche Sprachdatei	049 044
1,0	long	Modus 0 = Sprachdatei(en) schreiben 1 = Sprachdatei(en) löschen	0

NC.LanguagesDownload, write

row, col	Typ	Beschreibung	Bsp.: Löschen von Sprachdateien aus dem Installationsverzeichnis der NC
0,0	string	Angabe der Sprachdatei(en) - bei mehreren Dateien sind diese durch Leerzeichen zu trennen im Beispiel: 044 = englische Sprachdatei 007 = russische Sprachdatei	044 007
1,0	long	Modus 0 = Sprachdatei(en) schreiben 1 = Sprachdatei(en) löschen	1

4.3.3 Fehler-Codes

Firmware-Management-Error benutzt SCPErrNACK6 (106) zum Anzeigen eines Fehlers.

Detaillierte Fehlernummer <long>	Beschreibung
0	kein Fehler
1	Firmware-Datei oder Firmware-Verzeichnis fehlt
2	Firmware-Datei ist defekt (Formatfehler) oder unvollständig
3	Im aktuellen Gerätestatus ist der Download nicht erlaubt
4	Gerätefehler (Hardware- oder Gerätefirmware-Problem im Zielgerät)
5	Fehler bei der Download-Verifikation

Tab. 4-4: Fehlercodes im Firmware-Management

4.4 Sercos

4.4.1 Übersicht

Sercos-Parameter definieren einen standardisierten Zugriff auf Daten der Sercos-Geräte. Das Schema der Sercos-Parameter hat sich über die Antriebsparametrierung entwickelt.

In Sercos sind die Zugriffe auf Antriebsparameter genormt.

Es gibt weitere Parameter, die in der entsprechenden Gerätedokumentation beschrieben sind.

Der Sercos-Ring wird als Kommunikationsverbindung zwischen Steuerungen und Antrieben benutzt.. Auf dem Sercos-Ring werden die Teilnehmer über die "Sercos Adresse" angesprochen. Steuerungen implementieren in der Regel den Zugriff auf die Antriebsparameter. Der Antrieb wird über die "Sercos Adresse" angesprochen.

4.4.2 Sercos Idents

Zur Identifikation eines Sercos-Parameters werden Idents benutzt. Sie werden als **String** kodiert.

Ident-Format

X-Y-ZZZZ	16Bit-Identnummer-Bereich	
X-Y-ZZZZ.SI[.SE]	32Bit-Identnummer-Bereich	
X	Parametertyp bestehend aus einem Buchstaben. Siehe auch Gerätebeschreibung.	
	Beispiele:	
	S	Standard Parameter
	P	Produktspezifischer Parameter
Y	Parametersatz. (0 ... 7)	
ZZZZ	Datenblock- Nummer (0000 ... 4095) Achtung, die Datenblocknummer wird mit 4 Stellen Dezimal angegeben.	
SI	Sercos Instanz Dezimal ohne führende 0. (0..255)	
SE	Optional: Sercos Element Dezimal ohne führende 0. (0..255)	

Tab. 4-5: Sercos Idents

4.4.3 Sercos Elemente

Sercos Parameter bestehen aus mehreren Elementen. Elemente können explizit gelesen werden.

Sercos Elemente

Element	Beschreibung	Darstellungsformat	Beispiel
Status	Parameterstatus	Hexadezimal 16 Bit	"0x0000"
Name	Identifier des Sercos Parameters	String	"NC cycle time (TNcyc)"

Einheit	physikalische Einheit des Parameters	String	"µs"
Attribut	Typ und Wichtung des Sercos Parameters	Hexadezimal 32 Bit	"0x60110001"
Min	Minimalwert	abh. vom Attribut	"200.000"
Max	Maximalwert	abh. vom Attribut	"2000.000"
Wert	Betriebsdatum des Sercos Parameters	abh. vom Attribut	"1500.000"

Tab. 4-6: Sercos Parameter Elemente

4.4.4 Sercos Attribute und Datentypen

Über das Sercos-Attribut wird der Typ und das Anzeigeformat des Parameters geliefert. Das Attribut wird über einen String in 32 Bit Hex kodiert übertragen.

Das Gerät legt das Attribut für einen bestimmten Sercos-Parameter fest.



Die Umrechnung der Binärdaten zu einem Anzeigewert wird in der Kommunikationsschnittstelle durchgeführt.

Attributkodierung

Beschreibung	Typ	Bits
Bewertungsfaktor	0..65535	0..15
Datenlänge	0: 1 Byte 1: 2 Byte 2: 4 Byte 3: 8 Byte	16..17
Liste	0: keine Liste 1: Liste	18
Parametertyp	0: Betriebsdatum 1: Kommando	19
Anzeigeformat	0: Binär 1: Dezimal ohne Vorzeichen 2: Dezimal mit Vorzeichen 3: Hexadezimal ohne Vorzeichen 4: String 5: Ident 6: Float 7: Sercos Zeit	20..22
Nachkommastellen	0..15	24..27
Schreibschutz (CP2..CP4)		28..30

Tab. 4-7: Sercos Parameter Attributkodierung

Bewertungsfaktor	Ein Faktor, mit dem die Daten des Gerätes gewichtet sind. Der Bewertungsfaktor wird für die Datenelemente Min, Max und Wert genutzt. Eine Verrechnung erfolgt mit den Anzeigeformaten Binär, Dezimal und Hexadezimal.
Datenlänge	Anzahl der Bytes. Wird verwendet für Min, Max und Wert. Gilt auch für Listenelemente. Strings werden mit der Datenlänge 1 Byte übertragen.
Liste	Flag zur Kennzeichnung von Listen.
Anzeigeformat	Interpretationskennzeichnung der gelieferten Daten. Die Daten werden entsprechend des Anzeigeformats interpretiert.
Nachkommastellen	Wird bei Zahlenformaten zur zusätzlichen Bewertung genutzt.

Sercos Anzeigeformate

Name	Datenbreite in Bytes	Kodierung	Beispiel
Binär	2, 4, 8	0: FALSE 1: TRUE	"0000010100000100"
Dezimal o. Vorzeichen	2, 4, 8	vorzeichenlose Dezimalzahl	"12643" "123.234" bei 3 Nachkommastellen Bewertungsfaktor und Nachkommastellen werden berücksichtigt.
Dezimal m. Vorzeichen	2, 4, 8	vorzeichenbehaftete Dezimalzahl	"432347" "-423823" "-24238.23 "bei 2 Nachkommastellen Bewertungsfaktor und Nachkommastellen werden berücksichtigt.
Hexadezimal	1, 2, 4, 8	Hexadezimalzahl	"0x1242" Der Bewertungsfaktor wird berücksichtigt.
String	1	UTF8	"Hallo"
Ident	4	siehe Kapitel Ident Formate	"S-0-0030.4.6"
Float	4, 8	Gleitkommadarstellung	"1.2354"
Sercos Zeit	8	gemäß IEC 61588	"2012-04-25T10:04:25Z123456ns"

Tab. 4-8: Sercos Parameter Anzeigeformate

4.4.5 Sercos Listen

Betriebsdaten können auch Listen enthalten. Dies wird durch das Attribut entsprechend angezeigt.

Kodierung	Listenelemente werden über "\n" getrennt.
Min Max Werte bei Listen	Geben die entsprechenden Grenzwerte für die Elemente der Liste an.

Übertragung mit Elementkodierung "D"

Der Wert der Liste wird in einer Zelle des Ergebnisses angegeben.

Beispiel:

Übertragung mit Elementkodierung "D"

SPA1, read

row, col	Typ	Beschreibung	SPA1,3,S-0-0017,D
0,0	string	Liste	S-0-0001\n S-0-0002\n ... S-0-xxxx

Tab. 4-9: Sercos Parameter Listenformat "D"

Übertragung mit Advanced Elementkodierung

Die Liste wird bei dieser Übertragungsform mit der aktuellen und maximalen Länge der Liste übertragen.

Beispiel:

Liste mit Elementkodierung 40

SPA1, read

row, col	Typ	Beschreibung	SPA1,3,S-0-0017,40
0,0	string	aktuelle Listenlänge	"400"
0,1	string	maximale Listenlänge	"400"
0,2	string	Liste	"S-0-0001\n S-0-0002\n ... S-0-xxxx"

Tab. 4-10: Sercos Parameter Listenformat "40"

4.4.6 Anfordern mehrerer Elemente

Über das Advanced-Format können mehrere Elemente eines Sercos-Parameters gelesen werden.

Elementkodierung

Element des Parameters	Standard-Format	Advanced Format
	nur ein Buchstabe kann angegeben werden	Werte können durch "Oder" verknüpft werden
Datenstatus	S	01H
Name	N	02H
Attribut	A	04H
Einheit	U	08H
min. Eingabewert	L	10H

max Eingabewert	H	20H
Betriebsdatum	D	40H

Tab. 4-11: Sercos Parameter-Elementkodierung

Datenformat

Die Elemente werden in obiger folgender Reihenfolge zurückgeliefert

Beispiel:

Anfordern aller Sercos Elemente

Ergebnisdaten Lesen von SPA1,3,S-0-0111,7f

row, col	Typ	Beschreibung	SPA1,3,S-0-0111,7f
0,0	string	Datenstatus	"0x0000"
1,0	string	Name	"Stillstandsstrom Motor"
2,0	string	Attribut	"0x43120001"
3,0	string	Unit	"A eff "
4,0	string	Min	"0.000"
5,0	string	Max	"500.000"
6,0	string	Wert	"1.000"

Tab. 4-12: Sercos Parameter Reihenfolge der Elemente

Beispiel:

Anfordern aller Sercos Elemente eines Listenparameters

Ergebnisdaten Lesen von SPA1,3,S-0-0017,7f

row, col	Typ	Beschreibung	SPA1,3,S-0-0017,7f
0,0	string	Datenstatus	"0x0000"
1,0	string	Name	"IDN-Liste aller Betriebsdaten"
2,0	string	Attribut	"0x70550001"
3,0	string	Unit	"_"
4,0	string	Min	"_"
5,0	string	Max	"_"
6,0	string	aktuelle Listenlänge	"2116"
7,0	string	maximale Listenlänge	"4986"
8,0	string	Wert	"S-0-0000\nS-0-0001\n... ... P-0-4092\nP-0-4095"

Tab. 4-13: Sercos Parameter Anfordern aller Elemente

4.4.7 Sercos Attributecache

Übersicht

Sercos-Parameterelemente werden in der Kommunikationsschicht entsprechend ihres Aufbaus behandelt. Hierzu ist es erforderlich den Attributwert des Sercos-Parameters zu kennen. Üblicherweise wird bei jedem Zugriff auf den Sercos-Parameter das Attribut immer neu gelesen. Dies führt zu einem ent-

sprechendem Overhead. Um dies zu vermeiden gibt es den Attributecache, der die Attributwerte von Sercos-Parametern für künftige Zugriffe zwischenzeitlich speichert.

Bei verschiedenen Sercos-Parametern ist es möglich, Attributwerte zu ändern. Diese Attributwerte dürfen nicht gecached werden. Der Attributecache wird über Konfiguration für bestimmte Sercos-Parameter bzw. Parameterbereiche freigeschaltet.

Die Konfiguration des Caches erfolgt achsbezogen innerhalb eines Geräts. Siehe hierzu auch [SAC.Config, Seite 44](#)

Zur Konfiguration des Caches werden Hilfsfunktionen für die Programmierumgebung .NET zur Verfügung gestellt. Siehe auch SCPMnoSupport.dll die Klassen `Rexroth.IndraWorks.SCP.SPAAtributeCacheData` und `Rexroth.IndraWorks.SCP.SPAAtributeCache`

4.4.8 Sercos Fehlercodes

Sercos-Fehler werden von den Geräten über den lwSCP-Fehler SCPErr-NACK1 (101) geliefert.

Die Sercos-spezifischen Fehler werden über eine detaillierte Fehlernummer geliefert:

Nummer (dez.)	Nummer (hexadez.)	Beschreibung
0	0x0000	kein Fehler
1	0x0001	Servicekanal ist nicht geöffnet
9	0x0009	Ungültiges Schließen des Servicekanals
144	0x0090	Steuerung z.Z. beschäftigt, Anfrage nicht möglich. Später noch einmal probieren
160	0x00A0	"Anforderung unzulässig" z. B. Zugriff auf S-/P-Parameter im Initialisierungsmodus
176	0x00B0	"Ausgewähltes Element unzulässig" nur "Betriebsdatum" kann geschrieben werden
192	0x00C0	"Antriebsadresse unzulässig" Antriebsadresse ist größer als erlaubt oder Antrieb ist im Sercos Ring nicht aktiv (deaktiviert oder nicht vorhanden)
240	0x00F0	"Fataler Fehler" während einer Parameterübertragung ist ein CLC-interner Fehler aufgetreten (s. C-0-0041), der den Datenaustausch beeinflusst hat
4097	0x1001	Identnummer nicht vorhanden
4105	0x1009	Ungültiger Zugriff auf Element 1
5094	0x13E6	Konvertierungsfehler Ein/Ausgabeformat
5095	0x13E7	"Dateizugriffsfehler, Filesystem prüfen" Zur Übertragung großer Datenmengen werden Dateien benötigt. Eventuell ist der Speicher im RAM-Filesystem der Steuerung voll.

Nummer (dez.)	Nummer (hexadez.)	Beschreibung
8193	0x2001	Name nicht vorhanden
8194	0x2002	Name zu kurz übertragen
8195	0x2003	Name zu lang übertragen
8196	0x2004	Name nicht änderbar
8197	0x2005	Name z.Z. schreibgeschützt
12290	0x3002	Attribut zu kurz übertragen
12291	0x3003	Attribut zu lang übertragen
12293	0x3004	Attribut nicht änderbar
12294	0x3005	Attribut z.Z. schreibgeschützt
16385	0x4001	Einheit nicht vorhanden
16386	0x4002	Einheit zu kurz übertragen
16387	0x4003	Einheit zu lang übertragen
16388	0x4004	Einheit nicht änderbar
16389	0x4005	Einheit z.Z. schreibgeschützt
20481	0x5001	min. Eingabewert nicht vorhanden
20482	0x5002	min. Eingabewert zu kurz übertragen
20483	0x5003	min. Eingabewert zu lang übertragen
20484	0x5004	min. Eingabewert nicht änderbar
20485	0x5005	min. Eingabewert z.Z. schreibgeschützt
24577	0x6001	max. Eingabewert nicht vorhanden
24578	0x6002	max. Eingabewert zu kurz übertragen
24579	0x6003	max. Eingabewert zu lang übertragen
24580	0x6004	max. Eingabewert nicht änderbar
24581	0x6005	max. Eingabewert z.Z. schreibgeschützt
28674	0x7002	Betriebsdatum zu kurz übertragen
28675	0x7003	Betriebsdatum zu lang übertragen
28676	0x7004	Betriebsdatum nicht änderbar
28677	0x7005	Betriebsdatum z.Z. schreibgeschützt
28678	0x7006	Betriebsdatum kleiner als min. Eingabewert
28679	0x7007	Betriebsdatum größer als max. Eingabewert
28680	0x7008	Ungültiges Betriebsdatum nicht korrekt
28681	0x7009	Betriebsdatum durch Passwort schreibgeschützt
28682	0x700A	Betriebsdatum ist z.Z. schreibgeschützt, da es zyklisch konfiguriert ist. (IDN im MDT bzw. AT konfiguriert, Schreiben über Servicekanal nicht erlaubt)

Nummer (dez.)	Nummer (hexadez.)	Beschreibung
28683	0x700B	Ungültige indirekte Adressierung (z. B. Datencontainer, Listenbehandlung)
28684	0x700C	Betriebsdatum z.Z. schreibgeschützt aufgrund anderer Einstellungen (z. B. Parameter, Betriebsart, Antriebsfreigabe, Antrieb EIN)
28688	0x7010	Kommando bereits aktiv
28689	0x7011	Kommando nicht unterbrechbar
28690	0x7012	Kommando z.Z. nicht ausführbar (z. B. Kommando in dieser Phase nicht aktivierbar)
28691	0x7013	Kommando nicht ausführbar (falsche Parameter)
28940	0x710C	Der eingegebene Wert ist außerhalb des gültigen Zahlenbereichs
32769	0x8001	"Zugriff zur Zeit nicht möglich" Servicekanal belegt, Datenübertragung wird nicht durchgeführt
32770	0x8002	"Störung im Servicekanal" Auf gewünschten Antrieb kann z.Z. nicht zugegriffen werden
32779	0x800B	Übertragung wurde durch Steuerung abgebrochen, da sie mit dem selben Antrieb kommunizieren muss (höhere Priorität)
32780	0x800C	Unerlaubter Zugriff, Servicekanal noch aktiv letzte Übertragung noch nicht beendet und neue Anforderung gestartet

Tab. 4-14: Sercos Parameter Fehlercodes

4.4.9 Sercos Items

SPA1

Beschreibung	Liest und schreibt Elemente der Sercos Parameter.
Anmerkung	SPA1 dient dem Zugriff auf Steuerungsparameter, wenn auf sie im Sercos Parameter-Format zugegriffen werden kann. Die Steuerungsparameter werden in der Regel über die Antriebsadresse "0" adressiert. SPA1 dient auch dem Zugriff auf Antriebsparameter, wenn die Antriebe über die Steuerungen erreichbar sind. Antriebe werden über die Antriebsadresse ausgewählt. Fehler, die beim Zugriff auf Sercos Parameter auftreten, werden über die lwSCP-Fehlernummer "SCPErrNACK1" abgebildet.
Item Syntax	SPA1,(1),(2),(3) 1. Antriebsadresse 2. Ident

Ident-Format

X-Y-ZZZZ	16Bit-Identnummer-Bereich
X-Y-ZZZZ.SI[.SE]	32Bit-Identnummer-Bereich
X	Parametertyp bestehend aus einem Buchstaben. Siehe auch Gerätebeschreibung. Beispiele:
	S Standard Parameter
	P Produktspezifischer Parameter
Y	Parametersatz. (0 ... 7)
ZZZZ	Datenblock- Nummer (0000 ... 4095) Achtung, die Datenblocknummer wird mit 4 Stellen Dezimal angegeben.
SI	Sercos Instanz Dezimal ohne führende 0. (0..255)
SE	Optional: Sercos Element Dezimal ohne führende 0. (0..255)

Tab. 4-15: Sercos Idents

3. Elementkodierung

Die Elementkodierung wird genutzt um Datenelemente des Sercos Parameters anzusprechen.

Elementkodierung

Element des Parameters	Standard-Format	Advanced Format
	nur ein Buchstabe kann angegeben werden	Werte können durch "Oder" verknüpft werden
Datenstatus	S	01H
Name	N	02H
Attribut	A	04H
Einheit	U	08H
min. Eingabewert	L	10H
max Eingabewert	H	20H
Betriebsdatum	D	40H

Tab. 4-16: Sercos Parameter-Elementkodierung

Beispiel:

SPA1,1,S-0-0003,L

Minimaler Eingabewert des Parameters S-0-0003 von Antrieb 1

SPA1,1,S-0-0003,D:

Betriebsdatum des Parameters S-0-0003 von Antrieb 1

SPA1,1,S-0-0003,48:

Einheit & Betriebsdatum des Parameters S-0-0003 von Antrieb 1

Read Die Elemente der Sercos Parameter werden grundsätzlich als Strings geliefert.

Ergebnisse sind in folgender Form kodiert:

angefordertes Element	Ergebnis	Beispiel
Datenstatus	hexadezimales Word	"0x0000"
Name	String	"NC cycle time (TNcyc)"
Attribut	hexadezimales Doppel-Word	"0x60110001"
Einheit	String	"µs"
min. Eingabewert	dezimales Word	"2000"
max. Eingabewert	dezimales Word	"20000"
Wert	dezimal hexadezimal (max. 8 Stellen) binär (max. 64 Zeichen) Zeit (nur für "S" und "P" Parametern)	"1.5" "0x80" "1111000010101010" "2012-08-16T09:25:01.123456789Z"

Tab. 4-17: Rückgabeformate für Sercos Elemente

Write Nur das Betriebsdatum kann geschrieben werden, falls das im Parameter möglich ist.

Datenformat Zeilenanzahl:
n=Anzahl der angeforderten Parameter

SPA1, read

row, col	Typ	Beschreibung	SPA1,3,S-0-0111,08
0,0	string	angefordertes Element	"A eff"

Tab. 4-18: SPA1, read

SPA1, write

row, col	Typ	Beschreibung	SPA1,3,S-0-0111,40
0,0	string	zuzuweisender Wert	"1.500"

Tab. 4-19: SPA1, write

Listen einfaches Format Mit Elementkodierung "D" angefordert

SPA1, read

row, col	Typ	Beschreibung	SPA1,3,S-0-0017,D
0,0	string	Liste	S-0-0001\n S-0-0002\n ... S-0-xxxx

Tab. 4-20: Listen einfaches Format - SPA1, read

Listen mit Länge

Mit Elementkodierung "40" angefordert.

SPA1, read

row, col	Typ	Beschreibung	SPA1,3,S-0-0017,40
0,0	string	Ist-Länge der Liste im Antrieb	"400"
0,1	string	max. Länge der Liste im Antrieb	"400"
0,2	string	Liste	"S-0-0001\n S-0-0002\n ... S-0-xxxx"

Tab. 4-21: Listen mit Länge - SPA1, read

SPA2**Beschreibung** Liest und schreibt Elemente der Sercos Parameter eines Antriebs.**Anmerkung** Fehler, die beim Zugriff auf Sercos Parameter auftreten, werden über die lwSCP-Fehlernummer "SCPErrNACK1" abgebildet.**Item Syntax** SPA2,(1),(2)**1. Ident****Ident-Format**

X-Y-ZZZZ	16Bit-Identnummer-Bereich
X-Y-ZZZZ.SI[.SE]	32Bit-Identnummer-Bereich
X	Parametertyp bestehend aus einem Buchstaben. Siehe auch Gerätebeschreibung. Beispiele:
	S Standard Parameter
	P Produktspezifischer Parameter
Y	Parametersatz. (0 ... 7)
ZZZZ	Datenblock- Nummer (0000 ... 4095) Achtung, die Datenblocknummer wird mit 4 Stellen Dezimal angegeben.

SI	Sercos Instanz Dezimal ohne führende 0. (0..255)
SE	Optional: Sercos Element Dezimal ohne führende 0. (0..255)

Tab. 4-22: Sercos Idents

2. Elementkodierung

Die Elementkodierung wird genutzt, um Datenelemente des Sercos Parameters anzusprechen.

Elementkodierung

Element des Parameters	Standard-Format nur ein Buchstabe kann angegeben werden	Advanced Format Werte können durch "Oder" verknüpft werden
Datenstatus	S	01H
Name	N	02H
Attribut	A	04H
Einheit	U	08H
min. Eingabewert	L	10H
max Eingabewert	H	20H
Betriebsdatum	D	40H

Tab. 4-23: Sercos Parameter-Elementkodierung

Beispiel:

SPA2,S-0-0003,L:

Minimaler Eingabewert des Parameters S-0-0003.

SPA2,S-0-0003,D:

Betriebsdatum des Parameters S-0-0003.

SPA1,S-0-0003,48:

Einheit & Betriebsdatum des Parameters S-0-0003.

Read Die Elemente der Sercos Parameter werden grundsätzlich als Strings geliefert.

Ergebnisse sind in folgender Form kodiert:

Angefordertes Element	Ergebnis	Beispiel
Datenstatus	hexadezimaler Word	"0x0000"
Name	String	"NC cycle time (TNcyc)"
Attribut	hexadezimaler Doppel-Word	"0x60110001"
Einheit	String	"µs"
min. Eingabewert	dezimaler Word	"2000"

Angefordertes Element	Ergebnis	Beispiel
max Eingabewert	dezimales Word	"20000"
Wert	dezimal hexadezimal (max. 8 Stellen) binär (max. 64 Zeichen) Zeit (nur für "S" und "P" Parametern)	"1.5" "0x80" "1111000010101010" "2012-08-16T09:25:01.123 456789Z"

Tab. 4-24: Rückgabeformate für Sercos Elemente

Write Nur der Wert kann geschrieben werden.

Datenformat SPA2, read

row, col	Typ	Beschreibung	SPA2,S-0-0111,08
0,0	string	angefordertes Element	"A eff"

Tab. 4-25: SPA2, read

SPA2, write

row, col	Typ	Beschreibung	SPA2,S-0-0111,40
0,0	string	zuzuweisender Wert	"1.500"

Tab. 4-26: SPA2, write

Listen einfaches Format Mit Elementkodierung "D" angefordert.

SPA2, read

row, col	Typ	Beschreibung	SPA2,S-0-0017,D
0,0	string	Liste	S-0-0001\n S-0-0002\n ... S-0-xxxx

Tab. 4-27: Listen einfaches Format - SPA2, read

Listen mit Länge Mit Elementkodierung "40" angefordert.

SPA1, read

row, col	Typ	Beschreibung	SPA2,S-0-0017,40
0,0	string	Ist-Länge der Liste im Antrieb	"400"
0,1	string	max. Länge der Liste im Antrieb	"400"
0,2	string	Liste	"S-0-0001\n S-0-0002\n ... S-0-xxxx"

Tab. 4-28: Listen mit Länge - SPA1, read

DeviceAddress**Beschreibung** Sercos Adresse**Item Syntax** **DeviceAddress**

Read Ermittelt die am Gerät eingestellte Geräteadresse. Das Item wird insbesondere dann verwendet, wenn auf das Gerät mittels Punkt-zu-Punkt- Adresse zugegriffen wird, dann aber die eingestellte Geräteadresse interessiert.

Datenformat **DeviceAddress**

row, col	Typ	Beschreibung	Bsp.
0,0	long	Geräteadresse	5

DeviceAddressList**Beschreibung** Antriebsadressen eines Mehrachsgeräts

Anmerkung Falls ein Gerät über mehrere Achsen verfügt, werden die verschiedenen Achsen über entsprechende Sercos Adressen angesprochen.

Item Syntax **DeviceAddressList**

Read Auslesen der Antriebsadressen des Geräts.

Datenformat Pro Achse wird eine Zeile (x) mit der Achsadresse geliefert.

DeviceAddressList

row, col	Typ	Beschreibung	Bsp.
x,0	long	Achsadresse	1

SAC.Config

Beschreibung Konfiguration des Attribute Caches für Sercos Parameterzugriffe. Pro Achse wird festgelegt, für welche Sercos Parameter das Attribut zwischengespeichert werden kann. Es können einzelne Idents, als auch Identbereiche angegeben werden.



Das Item ist ausschließlich für den internen Gebrauch vorgesehen. Konfiguration falscher Sercos Idents kann zu Fehlinterpretationen von Sercos Parameter Daten führen.

Item Syntax *SAC.Config[, (1)]*

1. Antriebsadresse

Die Adresse des Antriebs, dessen Cache Konfiguration angesprochen werden soll.

Der Parameter ist notwendig bei Sercans und MLC. Für IndraDrives und Subdrives ist der Parameter nicht anzugeben.



Steuerungsparameter werden in der Regel über Antriebsadresse 0 angesprochen. --> SAC.Config,0

Identformat

Idents und Bereichsgrenzen sind im Format <Typ>-<Set>-<Block> anzugeben.

Typ: Ein Großbuchstabe 'A' bis 'Z'

Set: Der Parametersatz 0 bis 7

Block: Die Blocknummer 4-stellige Dezimalzahl von 0000 bis 4095.

Identbereiche

Sie haben 2 Spalten mit min und max. Idents. Der Identbereich wird aus den jeweiligen Min und Max Werten der Angaben berechnet.

Beispiel:

A-0-0100 bis A-2-0200

Alle Idents zwischen A-0-0100 A-0-200 , zwischen A-1-0100 A-1-0200 und zwischen A-2-0100 A-2-0200

Verfügbarkeit SAC.Config

Sercans, IndraDrive, EcoDrive, MLC, nicht MTX

Read

Auslesen der Idents und Identbereiche, die gecached werden.

SAC.Config, Read

row, col	Typ	Beschreibung	Bsp.
0,0	string	Anfang eines Identbereichs	C-0-0000
0,1	string	Ende des Identbereichs	C-3-0123
1,0	string	Einzelnes Ident	C-0-0447

Write

Beschreiben von Identbereichen und einzelnen Idents.

SAC.Config, Write

row, col	Typ	Beschreibung	Bsp.
0,0	string	Anfang eines Identbereichs	C-0-0000
0,1	string	Ende des Identbereichs	C-3-0123
1,0	string	Einzelnes Ident	C-0-0447

Beim Schreiben der Konfiguration werden die gecachten Attributewerte gelöscht.

Call

Löschen der gecachten Attributewerte.

SAC.Config, Call Argument

row, col	Typ	Beschreibung	Bsp.
0,0	string	Kommando: Muss Reset beinhalten.	Reset

Die Konfiguration des Caches bleibt bestehen.

4.5 Diagnose und Logbuch

4.5.1 Übersicht

Die Diagnose ist eine Komponente in den Geräten MTX und IndraLogic, die den Zugriff auf Meldungen und das Logbuch zulässt. Meldungen werden gerätespezifisch abgelegt. Die entsprechenden Items stehen als Teil der Gerätefunktionalität zur Verfügung.

Das Logbuch steht als eigenständiger Teil zur Verfügung. Hier werden die Einträge verschiedener Geräte gesammelt.

Diagnose Items	• System.MessageStore Auslesen der aktuell anstehenden Meldungen. • System.MessageStoreEx Auslesen der weiterführenden Daten einer Meldung. • System.MessageStatus Abfrage ob Meldungen anstehen, sich ändern oder neue gekommen sind. • System.ErrorStatus Abfrage ob Meldungen der Klassifikation Fehler anstehen, sich ändern oder neue gekommen sind. • System.InfoStatus Abfrage ob Meldungen der Klassifikation Hinweis anstehen, sich ändern oder neue gekommen sind. • System.WarningStatus Abfrage ob Meldungen der Klassifikation Warnung anstehen, sich ändern oder neue gekommen sind. • System.StartupStatus Abfrage ob Meldungen der Klassifikation Startvoraussetzung anstehen, sich ändern oder neue gekommen sind. • System.SetupStatus Abfrage ob Meldungen der Klassifikation Einrichtdiagnose anstehen, sich ändern oder neue gekommen sind. • System.MessageReset Setzt aktuell anstehenden Meldungen zurück.
Logbuch Items	• System.LogbookStore Auslesen der Daten des Logbuchs. • System.LogbookStoreEx Auslesen der weiterführenden Daten eines Logbuch-Eintrages. • System.LogbookClear Löschen von Einträgen im Logbuch. • System.LogbookDevices Ermitteln der Steuerungen, die in dem Logbuch protokolliert werden. • System.LogbookStart Starten der Logbuch-Protokollierung.
Beschreibung	<i>Es gibt verschiedene Arten von Diagnose-Items:</i> • Allgemeine Kommandos

Mit diesen Kommandos können gleichzeitig Diagnosemeldungen unterschiedlichen Typs abgefragt werden, z. B. ProVi- und MTX-Meldungen.

- Meldungstyp-spezifische Kommandos

Dies Meldungen sind speziell für einen bestimmten Meldungstyp definiert, z. B. spezielle Kommandos für die ProVi-Kriterienanalyse

Auch wenn die allgemeinen Kommandos Daten für alle Diagnosearten liefern, sind die zurückgelieferten Ergebnisse für jeden Typ unterschiedlich.

Es gibt normalerweise immer einen fest definierten Teil, der bei allen Typen gleich ist und daran schließen sich dann die typspezifischen Daten an. Anhand eines Ergebnisses im fest definierten Teil kann erkannt werden, um welchen Typ es sich handelt und welche zusätzlichen Daten zu erwarten sind.

Parameter der Diagnose-Items

Hier sind die gültigen Wertebereiche der einzelnen Parametertypen für die Steuerungs-Diagnose-Items beschrieben.

Filter Diagnosetyp

Es gibt verschiedene Arten von Diagnose, diese werden im IDL-File "ScpDiagnosisComponent" durch den Enumerator "SCPDiagnosisType" definiert:

- "SCPDiagnosisType_MTX 0x01":
MTX-Meldungen
- "SCPDiagnosisType_ProVi 0x02":
ProVi-Meldungen
- "SCPDiagnosisType_IndraStep 0x04":
Schrittketten-Meldungen

Werden alle Diagnosetypen benötigt, kann der Wert "SCPDiagnosisType_All" verwendet werden. (Dieser Wert entspricht im Item-String dem String -1, z. B. System.MessageStore,-1,2,3,0)

Für eine Kombination von verschiedenen Typen können die entsprechenden Typen ODER verknüpft werden.

Beispiel:

MTX- und ProVi-Meldungen

SCPDiagnosisType_MTX | SCPDiagnosisType_ProVi entspricht 0x01 | 0x02 und ergibt 0x03



Auch wenn der Wert hexadezimal definiert ist, wird als Filter ein Dezimal-Wert erwartet

Filter Grobklassifikation

Es gibt verschiedene Arten von Diagnose-Klassen, diese werden im IDL-File "ScpDiagnosisComponent" durch den Enumerator "SPCDiagnosisClass" definiert:

SCPDiagnosisClass_Error	0x01	Fehler
SCPDiagnosisClass_Info	0x02	Hinweis
SCPDiagnosisClass_Warning	0x04	Warnung
SCPDiagnosisClass_Startup	0x08	Startvoraussetzung
SCPDiagnosisClass_Setup	0x10	Einrichtdiagnose

Werden alle Diagnose-Klassen benötigt, so kann der Wert "SCPDiagnosisClass_All" verwendet werden. (Dieser Wert entspricht im Item-String dem String -1, z. B. System.MessageStore,2,-1,3,0).

Für eine Kombination von verschiedenen Klassen können die entsprechenden Typen ODER verknüpft werden.

Beispiel: Warnung und Einrichtdiagnose

SCPDiagnosisClass_Warning | SCPDiagnosisClass_Setup

entspricht 0x04 | 0x10 und ergibt 0x14 als Dezimalwert 20 im String

Filter Fehlerquelle

Dieser Filter definiert die angeforderte Modulnummer.

Es wird direkt die angeforderte Modulnummer angegeben, z. B. System.MessageStore,2,1,3,0 (Modulnummer 3 angefordert).

Sollen Daten für mehrere Module angefordert werden, so werden diese durch ein Pluszeichen verbunden, z. B. System.MessageStore,2,1,2+3+5,0 (Modulnummer 2,3 und 5 angefordert).

Sollen die Daten für alle Module angefordert werden oder sind keine Module definiert, so kann der Wert SCPErrSource_All verwendet werden. Dieser ist im IDL-File "ScpDiagnosisComponent" durch den Enumerator "SCPErrSource" definiert. (Dieser Wert entspricht im Item-String dem String -1, z. B. System.MessageStore,2,2,-1,0)

4.5.2 Diagnose

System.MessageStore

Beschreibung	Auslesen aller aktuell anstehenden Meldungen. Beim Erzeugen des Items kann angegeben werden, welche Meldungen zurückgeliefert werden sollen.		
Anmerkung	Dieses Item ist nicht dafür geeignet jede Meldung zu erfassen. Wenn eine Meldung nur sehr kurz ansteht, wird diese hier nicht gemeldet. Ist es notwendig, jede Meldung zu erfassen, muss das Logbuch genutzt werden.		
Item Syntax	System.MessageStore,(1),(2),(3),(4)		
	1. Filter Diagnosetyp, es können mehrere Diagnosetypen gleichzeitig angefordert werden.		
	SCPDiagnosisType_MTX	0x01	MTX-Meldungen
	SCPDiagnosisType_ProVi	0x02	ProVi-Meldungen
	SCPDiagnosisType_IndraStep	0x04	Schritketten-Meldungen
	SCPDiagnosisType_All	-1	alle Diagnosetypen
	2. Filter Grobklassifikation, es können mehrere Klassifikationen gleichzeitig angefordert werden		
	SCPDiagnosisClass_Error	0x01	Fehler
	SCPDiagnosisClass_Info	0x02	Hinweis
	SCPDiagnosisClass_Warning	0x04	Warnung
	SCPDiagnosisClass_Startup	0x08	Startvoraussetzung
	SCPDiagnosisClass_Setup	0x10	Einrichtdiagnose
	3. Filter Fehlerquelle, es können mehrere Fehlerquellen gleichzeitig angegeben werden.		

Es wird direkt die angeforderte Modulnummer angegeben

Sollen Daten für mehrere Module angefordert werden, so werden die entsprechenden Modulnummern durch Pluszeichen verbunden

Sollen die Daten für alle Module angefordert werden oder sind keine Module definiert, so wird "-1" angegeben.

4. Anzahl der gewünschten Meldungen, 0 – alle Meldungen

Read Für jede anstehende Meldung wird eine Zeile zurückgeliefert. Wenn z. B. 3 Meldungen angefordert sind, aber nur eine Meldung ansteht, so hat das Ergebnis nur eine Zeile. Das Ergebnis hat 0 Zeilen, wenn keine Meldung ansteht.

Die Spaltenanzahl einer Zeile richtet sich nach dem Diagnosetyp der Meldung. Jede Zeile hat mindestens 9 Spalten, die wie folgt definiert sind.

Weil jede Meldung eindeutig einem Diagnosetyp, einer Grobklassifikation und einer Fehlerquelle zugeordnet werden kann, sind auch die zurückgegebenen Ergebnisse eindeutig, d. h. es gibt im Ergebnis keine Oder-Verknüpfung mehr.

Beispiel System.MessageStore

Datenformat Bei diesem Item werden verschiedene Formate abhängig vom Meldungstyp geliefert. Pro Meldung wird eine neue Zeile geliefert (x).

System.MessageStore, Allgemeiner Teil

row, col	Typ	Beschreibung	Bsp.
x,0	long	Diagnosetyp dieser Meldung, beschreibt auch die Daten ab der 9. Spalte	
x,1	long	Grobklassifikation dieser Meldung	
x,2	long	Fehlerquelle dieser Meldung	
x,3	string	Fehlertext mit aufgelösten Platzhaltern	
x,4	double	Zeitstempel der Meldung mit Datum und Uhrzeit vom Typ Variant Time. Die Zeit ist angegeben in Coordinated Universal Time (UTC). Filter Fehlerquelle	
x,5	long	Zeitindex zur zeitlichen Sortierung von Meldungen mit gleichem Zeitstempel. Wenn zwei Meldungen den gleichen Zeitstempel haben, ist die Meldung zuerst gekommen, deren Zeitindex kleiner ist. Wenn beide den gleichen Index haben, sind beide gleichzeitig gekommen oder der Unterschied ist nicht festzustellen.	
x,6	bool	TRUE – Kriterienanalyse Daten vorhanden FALSE – keine Kriterienanalyse Daten vorhanden	

row, col	Typ	Beschreibung	Bsp.
x,7	bool	TRUE – weiterführende Diagnose-Informationen vorhanden (können mit System.MessageStoreEx abgerufen werden) FALSE – keine weiterführenden Diagnose-Informationen vorhanden	
x,8	string	Eindeutige Meldungs-ID. Anhand dieser ID und des Diagnosetyps können Meldungen unterschieden werden, auch wenn mehrere Meldungen den gleichen Meldungstext haben. Dient als Eingangsparameter weiterer Diagnose-Kommandos.	

zusätzliche Spalten einer MTX-Meldung

Wird als Diagnosetyp "SCPDiagnosisType_MTX" zurückgegeben, so sind die weiteren Spalten wie folgt definiert:

System.MessageStore, Zusatz MTX-Meldung

row, col	Typ	Beschreibung	Bsp.
x,9	string	Fehler, Warnungs- oder Hinweisnummer	
x,10	long	Feinklassifikation Meldungsklasse (gerätespezifisch, ~20 bei BRIC)	
x,11	string	Meldungsklasse als Text	
x,12	long	Kanal oder Kennzeichnung als kanalunabhängig	
x,13	string	Kanalname (Konfigurationsdatum)	

zusätzliche Spalten einer ProVi-Meldung

Wird als Diagnose Typ "SCPDiagnosisType_ProVi" zurückgegeben, so sind die weiteren Spalten wie folgt definiert:

System.MessageStore, Zusatz ProVi-Meldung

row, col	Typ	Beschreibung	Bsp.
x,9	long	Meldungsnummer	
x,10	long	Fehlerkategorie als Nummer	
x,11	long	Meldungsgruppe als Nummer	

zusätzliche Spalten einer Schrittketten-Meldung

Wird als Diagnose Typ "SCPDiagnosisType_IndraStep" zurückgegeben, so sind die weiteren Spalten wie folgt definiert.

System.MessageStore, Zusatz Schrittketten-Meldung

row, col	Typ	Beschreibung	Bsp.
x,9	long	Meldungsart 0 = aktiver Schritt, 1 = Grundstellungsfahren, 2 = Grundstellung erreicht, 3 = Monitorfehler, 4 = Zeitfehler, 5 = Startvoraussetzungs-Meldung, 6 = Grundstellungs-Meldung	
x,10	string	Schrittketteninstanz	

System.MessageStoreEx

Beschreibung Ergänzende Daten zu einer Meldung, die mit System.MessageStore abgerufen wurde.

Anmerkung Die Datenart ist abhängig vom Diagnosetyp.
Das Ergebnis hat immer nur eine Zeile.

Item Syntax *System.MessageStoreEx, (1), (2)*

1. Diagnosetyp - Wert, der bei System.MessageStore in der Spalte 0 zurückgegeben wurde.
2. eindeutige Meldungs-ID - Wert, der bei System.MessageStore in der Spalte 7 zurückgegeben wurde.

Zugriff Read

Datenformat Die Spaltenanzahl richtet sich nach dem Diagnosetyp der Meldung. Es gibt keinen einheitlich definierten Teil des Ergebnisses. Je nach Diagnosetyp unterscheiden sich die folgenden Spalten.

Spalten einer MTX-Meldung

Wird als Diagnosetyp "SCPDiagnosisType_MTX" zurückgegeben, sind die Spalten wie folgt definiert.

System.MessageStoreEx, MTX-Meldung

row, col	Typ	Beschreibung	Bsp.
0,0	string	Ursachentext	
0,1	string	Behebungstext	

Spalten einer ProVi-Meldung

Wird als Diagnosetyp "SCPDiagnosisType_ProVi" zurückgegeben, sind die Spalten wie folgt definiert.

System.MessageStoreEx, ProVi-Meldung

row, col	Typ	Beschreibung	Bsp.
0,0	string	Ursachentext aus dem Message-Integrator	
0,1	string	Behebungstext aus dem Message-Integrator	
0,2	string	Anwender-Notizen aus dem Message-Integrator	

Items

row, col	Typ	Beschreibung	Bsp.
0,3	string	HTML-Link aus dem Message-Integrator	
0,4	long	Typ der POE, in der die Meldung programmiert ist. 1 = Programm, 2 = Funktionsblock	
0,5	string	Name der POE, in der die Meldung programmiert ist	
0,6	long	Typ des Details, in dem die Meldung programmiert ist 1 = Implementation, 2 = Aktion, 3 = Transition	
0,7	string	Name des Details, leer bei Implementation	
0,8	long	Netzwerknummer oder Zeile, in der die Meldung programmiert ist	
0,9	string	Variable, die als Zuweisung für die ProVi-Meldung verwendet wird	
0,10	string	POE Instanzen-Name	
0,11	string	Kriterienanalyse (nur interne Verwendung)	

Spalten einer Schrittketten-Meldung

Wird als Diagnosetyp "SCPDiagnosisType_IndraStep" zurückgegeben, so sind die Spalten wie folgt definiert.

System.MessageStoreEx, Schrittketten-Meldung

row, col	Typ	Beschreibung	Bsp.
0,0	long	Typ der POE, in der die Schrittkette programmiert ist. 1 = Programm, 2 = Funktionsblock	
0,1	string	Name der POE, in der die Schrittkette programmiert ist.	
0,2	string	Schritt, aus dem die Meldung kommt.	
0,3	long	Typ des Details, aus dem die Meldung kommt. 2 = Aktion, 3 = Transition. Nur bei Meldungsart Monitorfehler und Zeitfehler. Bei Monitorfehler immer Aktion.	
0,4	string	Name der Aktion, aus der die Meldung kommt. Nur bei Meldungsart Monitorfehler, Zeitfehler und Detail = Aktion.	

row, col	Typ	Beschreibung	Bsp.
0,5	long	Netzwerknummer oder Zeile, in der die Meldung programmiert ist. Nur bei Meldungsart Monitorfehler.	
0,6	string	Kriterienanalyse Nur bei Meldungsart Monitorfehler, Zeitfehler, Startvoraussetzungs-Meldung und Grundstellungs-Meldung. (nur interne Verwendung)	

System.MessageStatus

Beschreibung Es wird abgefragt, ob bestimmte Meldungen anstehen, ob sich diese geändert haben oder ob neue Meldungen gekommen sind. Beim Erzeugen des Items kann angegeben werden, welche Meldungen zurückgeliefert werden sollen.

Anmerkung Dieses Item ist nicht dafür geeignet, jede Meldung zu erfassen. Wenn eine Meldung nur sehr kurz ansteht, wird diese hier nicht gemeldet. Ist es notwendig jede Meldung zu erfassen, muss das Logbuch genutzt werden.

Item Syntax **System.MessageStatus,(1),(2),(3)**

1. Filter Diagnosetyp, es können mehrere Diagnosetypen gleichzeitig angefordert werden

SCPDiagnosisType_MTX	0x01	MTX-Meldungen
SCPDiagnosisType_ProVi	0x02	ProVi-Meldungen
SCPDiagnosisType_IndraStep	0x04	Schrittketten-Meldungen
SCPDiagnosisType_All	-1	alle Diagnosetypen

2. Filter Grobklassifikation, es können mehrere Diagnosetypen gleichzeitig angefordert werden

SCPDiagnosisClass_Error	0x01	Fehler
SCPDiagnosisClass_Info	0x02	Hinweis
SCPDiagnosisClass_Warning	0x04	Warnung
SCPDiagnosisClass_Startup	0x08	Startvoraussetzung
SCPDiagnosisClass_Setup	0x10	Einrichtdiagnose

3. Filter Fehlerquelle

Es wird direkt die angeforderte Modulnummer angegeben

Sollen Daten für mehrere Module angefordert werden, so werden die entsprechenden Modulnummern durch Pluszeichen verbunden

Sollen die Daten für alle Module angefordert werden oder sind keine Module definiert, so wird "-1" angegeben.

Read

Beispiel **System.MessageStatus,?,?,?**

Datenformat System.MessageStatus

row, col	Typ	Beschreibung	Bsp.
0,0	bool	TRUE – es stehen Meldungen an FALSE – es stehen keine Meldungen an	
0,1	double	Zeitstempel der letzten angekommenen Meldung mit Datum und Uhrzeit vom Typ Variant Time. Die Zeit ist angegeben in Coordinated Universal Time (UTC).	
0,2	double	Zeitstempel der letzten gesendeten Meldung mit Datum und Uhrzeit vom Typ Variant Time. Die Zeit ist angegeben in Coordinated Universal Time (UTC).	

Beschreibung

Die Ergebnisse können wie folgt interpretiert werden:

- Stehen Meldungen an? Dazu wird Spalte 0 abgefragt.
- Kommt die erste Meldung oder geht die letzte Meldung? Zyklische Anmeldung dieses Items, wenn die Callback-Methode aufgerufen wird, Spalte 0 abfragen.
- Haben sich Meldungen geändert? Zyklische Anmeldung dieses Items, wenn die Callback-Methode aufgerufen wird, haben sich die Meldungen geändert.
- Gibt es neue Meldungen? Kommen-Zeitstempel merken. Zyklisches Anmelden dieses Items, wenn die Callback-Methode aufgerufen wird und der Kommen-Zeitstempel sich geändert hat, sind neue Meldungen angekommen.

System.ErrorStatus

Beschreibung Es wird abgefragt, ob Fehler anstehen, ob sich diese geändert haben oder ob neue Fehler hinzugekommen sind. Beim Erzeugen des Items kann angegeben werden, welche Fehler zurückgeliefert werden sollen.

Anmerkung Dieses Item ist nicht dafür geeignet jeden Fehler zu erfassen. Wenn eine Fehlermeldung nur sehr kurz ansteht, wird sie hier nicht gemeldet. Ist es notwendig jeden Fehler zu erfassen, muss das Logbuch genutzt werden.

Item Syntax **System.ErrorStatus,(1),(2)**

1. Filter Diagnosetyp, es können mehrere Diagnosetypen gleichzeitig angefordert werden.

SCPDiagnosisType_MTX	0x01	MTX-Meldungen
SCPDiagnosisType_ProVi	0x02	ProVi-Meldungen
SCPDiagnosisType_IndraStep	0x04	Schrittketten-Meldungen
SCPDiagnosisType_All	-1	alle Diagnosetypen

2. Filter Fehlerquelle, es können mehrere Fehlerquellen gleichzeitig angegeben werden.

Es wird direkt die angeforderte Modulnummer angegeben

Sollen Daten für mehrere Module angefordert werden, so werden die entsprechenden Modulnummern durch Pluszeichen verbunden

Sollen die Daten für alle Module angefordert werden oder sind keine Module definiert, so wird "-1" angegeben.

Read**Beispiel** System.ErrorStatus,?,?**Datenformat** System.ErrorStatus

row, col	Typ	Beschreibung	Bsp.
0,0	bool	TRUE – es stehen Fehler an FALSE – es stehen keine Fehler an	
0,1	double	Zeitstempel des letzten angekommenen Fehlers mit Datum und Uhrzeit vom Typ Variant Time. Die Zeit ist angegeben in Coordinated Universal Time (UTC).	
0,2	double	Zeitstempel des letzten gesendeten Fehlers mit Datum und Uhrzeit vom Typ Variant Time. Die Zeit ist angegeben in Coordinated Universal Time (UTC).	

System.InfoStatus

Beschreibung Es wird abgefragt, ob Hinweise anstehen, ob sich diese geändert haben oder ob neue Hinweise hinzugekommen sind. Beim Erzeugen des Items kann angegeben werden, welche Hinweise zurückgeliefert werden sollen.

Anmerkung Dieses Item ist nicht dafür geeignet jeden Hinweis zu erfassen. Wenn ein Hinweis nur sehr kurz ansteht, wird er hier nicht gemeldet. Ist es notwendig jeden Hinweis zu erfassen, muss das Logbuch genutzt werden.

Item Syntax System.InfoStatus**Read****Beispiel** System.InfoStatus**Datenformat** System.InfoStatus

row, col	Typ	Beschreibung	Bsp.
0,0	bool	TRUE – es stehen Hinweise an FALSE – es stehen keine Hinweise an	
0,1	double	Zeitstempel des zuletzt angekommenen Hinweises mit Datum und Uhrzeit vom Typ Variant Time. Die Zeit ist angegeben in Coordinated Universal Time (UTC).	
0,2	double	Zeitstempel des zuletzt gesendeten Hinweises mit Datum und Uhrzeit vom Typ Variant Time. Die Zeit ist angegeben in Coordinated Universal Time (UTC).	

System.WarningStatus

Beschreibung Es wird abgefragt, ob Warnungen anstehen, ob sich diese geändert haben oder ob neue Warnungen hinzugekommen sind. Beim Erzeugen des Items kann angegeben werden, welche Warnungen zurückgeliefert werden sollen.

Items

Anmerkung Dieses Item ist nicht dafür geeignet jede Warnung zu erfassen. Wenn eine Warnung nur sehr kurz ansteht, wird sie hier nicht gemeldet. Ist es notwendig jede Warnung zu erfassen, muss das Logbuch genutzt werden.

Item Syntax **System.WarningStatus**

Read

Beispiel **System.WarningStatus**

Datenformat **System.WarningStatus**

row, col	Typ	Beschreibung	Bsp.
0,0	bool	TRUE – es stehen Warnungen an FALSE – es stehen keine Warnungen an	
0,1	double	Zeitstempel der zuletzt angekommenen Warnung mit Datum und Uhrzeit vom Typ Variant Time. Die Zeit ist angegeben in Coordinated Universal Time (UTC).	
0,2	double	Zeitstempel der zuletzt gesendeten Warnung mit Datum und Uhrzeit vom Typ Variant Time. Die Zeit ist angegeben in Coordinated Universal Time (UTC).	

System.StartupStatus

Beschreibung Es wird abgefragt, ob Startvoraussetzungen anstehen, ob sich diese geändert haben oder ob neue Startvoraussetzungen hinzugekommen sind. Beim Erzeugen des Items kann angegeben werden, welche Startvoraussetzungen zurückgeliefert werden sollen.

Anmerkung Dieses Item ist nicht dafür geeignet jeden Meldung zu erfassen. Wenn eine Meldung nur sehr kurz ansteht, wird sie hier nicht gemeldet. Ist es notwendig jeden Meldung zu erfassen, muss das Logbuch genutzt werden

Item Syntax **System.StartupStatus**

Read

Beispiel **Konkretes Item mit Parameter , Z.B.System.StartupStatus**

Datenformat **System.StartupStatus**

row, col	Typ	Beschreibung	Bsp.
0,0	bool	TRUE – es stehen Startvoraussetzungen an FALSE – es stehen keine Startvoraussetzungen an	
0,1	double	Zeitstempel der zuletzt angekommenen Startvoraussetzung mit Datum und Uhrzeit vom Typ Variant Time. Die Zeit ist angegeben in Coordinated Universal Time (UTC).	
0,2	double	Zeitstempel der zuletzt gesendeten Startvoraussetzung mit Datum und Uhrzeit vom Typ Variant Time. Die Zeit ist angegeben in Coordinated Universal Time (UTC).	

System.SetupStatus

Beschreibung	Es wird abgefragt, ob Einrichtdiagnosen anstehen, ob sich diese geändert haben oder ob neue Einrichtdiagnosen hinzugekommen sind. Beim Erzeugen des Items kann angegeben werden, welche Einrichtdiagnosen zurückgeliefert werden sollen.
Anmerkung	Dieses Item ist nicht dafür geeignet jeden Meldung zu erfassen. Wenn eine Meldung nur sehr kurz ansteht, wird sie hier nicht gemeldet. Ist es notwendig jeden Meldung zu erfassen, muss das Logbuch genutzt werden
Item Syntax	System.SetupStatus
Read	
Beispiel	System.SetupStatus

Datenformat **System.SetupStatus**

row, col	Typ	Beschreibung	Bsp.
0,0	bool	TRUE – es stehen Einrichtdiagnosen an FALSE – es stehen keine Einrichtdiagnosen an	
0,1	double	Zeitstempel der zuletzt angekommenen Einrichtdiagnose mit Datum und Uhrzeit vom Typ Variant Time. Die Zeit ist angegeben in Coordinated Universal Time (UTC).	
0,2	double	Zeitstempel der zuletzt gesendeten Einrichtdiagnose mit Datum und Uhrzeit vom Typ Variant Time. Die Zeit ist angegeben in Coordinated Universal Time (UTC).	

System.MessageReset

Beschreibung	Aktuell anstehende Meldungen werden zurückgesetzt. Das Item selber hat keine Parameter. Welche Meldungen zurückgesetzt werden sollen, wird beim Write angegeben.
Anmerkung	Für zukünftige Erweiterung. Dieses Item setzt noch keine Meldungen zurück.
Item Syntax	System.MessageReset
Write	
Beispiel	System.MessageReset

Datenformat **System.MessageReset**

row, col	Typ	Beschreibung	Bsp.
0,0	long string	Filter Diagnosetyp, es können mehrere Diagnosetypen gleichzeitig zurückgesetzt werden.	
0,1	long string	Filter Grobklassifikation, es können mehrere Klassifikationen gleichzeitig zurückgesetzt werden.	
0,2	long string	Filter Fehlerquelle, es können mehrere Fehlerquellen gleichzeitig angegeben werden.	

Items

Filter Diagnosetyp	SCPDiagnosisType_MTX	0x01	MTX-Meldungen
	SCPDiagnosisType_ProVi	0x02	ProVi-Meldungen
	SCPDiagnosisType_IndraStep	0x04	Schrittketten-Meldungen
	SCPDiagnosisType_All	-1	alle Diagnosetypen
Filter Grobklassifikation	SCPDiagnosisClass_Error	0x01	Fehler
	SCPDiagnosisClass_Info	0x02	Hinweis
	SCPDiagnosisClass_Warning	0x04	Warnung
	SCPDiagnosisClass_Startup	0x08	Startvoraussetzung
	SCPDiagnosisClass_Setup	0x10	Einrichtdiagnose
Filter Fehlerquelle	Es wird direkt die angeforderte Modulnummer angegeben Sollen Daten für mehrere Module angefordert werden, so werden die entsprechenden Modulnummern durch Pluszeichen verbunden Sollen die Daten für alle Module angefordert werden oder sind keine Module definiert, so wird "-1" angegeben.		

4.5.3 Logbuch

System.LogbookStore

<

SCPDiagnosisClass_Error	0x01	Fehler
SCPDiagnosisClass_Info	0x02	Hinweis
SCPDiagnosisClass_Warning	0x04	Warnung
SCPDiagnosisClass_Startup	0x08	Startvoraussetzung
SCPDiagnosisClass_Setup	0x10	Einrichtdiagnose

4. Filter Fehlerquelle

Es können mehrere Fehlerquellen gleichzeitig angegeben werden.

Es wird direkt die angeforderte Modulnummer angegeben

Sollen Daten für mehrere Module angefordert werden, so werden die entsprechenden Modulnummern durch Pluszeichen verbunden

Sollen die Daten für alle Module angefordert werden oder sind keine Module definiert, so wird "-1" angegeben.

5. Zeitstempel, ab dem die Meldungen geliefert werden sollen. Meldung mit Datum und Uhrzeit vom Typ Variant Time. Die Zeit ist angegeben in Coordinated Universal Time (UTC).

Es werden alle Meldungen geliefert, deren Zeitstempel neuer oder gleich ist. Wird **0** angegeben, so werden alle Meldungen zurückgeliefert, die älter oder gleich dem Parameter 6 sind. Es ist dann nicht festgelegt, wie alt die Messung höchstens sein darf.

6. Zeitstempel, bis zu dem die Meldungen geliefert werden sollen. Meldung mit Datum und Uhrzeit vom Typ Variant Time. Die Zeit ist angegeben in Coordinated Universal Time (UTC).

Es werden alle Meldungen geliefert, deren Zeitstempel älter oder gleich ist. Wird **0** angegeben, so werden alle Meldungen zurückgeliefert, die neuer oder gleich dem Parameter 5 sind. Es ist dann nicht festgelegt, wie alt die Messung mindestens sein muss.

Zugriff Read

Die Spaltenanzahl einer Zeile richtet sich nach dem Diagnosetyp der Meldung.

Jede Meldung ist eindeutig einem Diagnosetyp, einer Grobklassifikation und einer Fehlerquelle zugeordnet. Dadurch sind die zurückgegebenen Ergebnisse eindeutig, d. h. es gibt im Ergebnis keine Oder-Verknüpfung.

Jede Zeile hat mindestens 10 Spalten, die wie folgt definiert sind.

Datenformat Die Anzahl der Zeilen entspricht der Anzahl der Logbucheinträge.

System.LogbookStore, Allgemein

row, col	Typ	Beschreibung	Bsp.
x,0	string	Device-Name	
x,1	string	Device-GUID	
x,2	long	Diagnosetyp dieser Meldung, bestimmt die Daten ab Spalte 10	
x,3	long	Grobklassifikation dieser Meldung	
x,4	long	Fehlerquelle dieser Meldung	
x,5	string	Fehlertext mit aufgelösten Platzhaltern	
x,6	double	Zeitstempel der Meldung mit Datum und Uhrzeit vom Typ Variant Time. Die Zeit ist angegeben in Coordinated Universal Time (UTC)	

row, col	Typ	Beschreibung	Bsp.
x,7	uint	1 – Die Meldung ist gekommen 0 – Die Meldung ist gegangen	
x,8	uint	1 – weiterführende Diagnose-Informationen vorhanden (mit System.LogbookStoreEx abrufen) 0 – keine weiterführenden Diagnose-Informationen vorhanden	
x,9	string	Eindeutige Meldungs-ID. Anhand dieser ID und des Diagnosetyps können Meldungen eindeutig unterschieden werden, z. B. um eine gehende Meldung der kommenden zuzuordnen. Dient als Eingangsparameter weiterer Diagnose-Kommandos.	

Zusätzliche Spalten einer MTX-Meldung

Wird als Diagnosetyp "SCPDiagnosisType_MTX" zurückgegeben, so sind die weiteren Spalten wie folgt definiert.

System.LogbookStore, Zusatz MTX-Meldung

row, col	Typ	Beschreibung	Bsp.
x,10	string	Fehler, Warnung oder Hinweis	
x,11	long	Feinklassifikation Meldungsklasse (gerätespezifisch, ~20 bei BRIC)	
x,12	string	Meldungsklasse als Text	
x,13	long	Kanal oder Kennzeichnung als kanalunabhängig	
x,14	string	Kanalname (Konfigurationsdatum)	

Zusätzliche Spalten einer ProVi-Meldung

Wird als Diagnosetyp "SCPDiagnosisType_ProVi" zurückgegeben, so sind die weiteren Spalten wie folgt definiert.

System.LogbookStore, Zusatz ProVi-Meldung

row, col	Typ	Beschreibung	Bsp.
x,10	long	Meldungsnummer	
x,11	long	Fehlerkategorie als Nummer	
x,12	long	Meldungsgruppe als Nummer	
x,13	bool	TRUE – Kriterienanalyse-Daten vorhanden FALSE – keine Kriterienanalyse-Daten vorhanden	

Spalten einer Schrittketten-Meldung

Wird als Diagnosetyp "SCPDiagnosisType_IndraStep" zurückgegeben, so sind die weiteren Spalten wie folgt definiert.

System.LogbookStore, Zusatz Schrittketten-Meldung

row, col	Typ	Beschreibung	Bsp.
x,10	long	Meldungsart 0 = aktiver Schritt, 1 = Grundstellungsfahren, 2 = Grundstellung erreicht, 3 = Monitorfehler, 4 = Zeitfehler, 5 = Startvoraussetzungs-Meldung, 6 = Grundstellungs-Meldung	
x,11	string	Schrittketteninstanz	
x,12	bool	TRUE – Kriterienanalyse-Daten vorhanden FALSE – keine Kriterienanalyse-Daten vorhanden	

System.LogbookStoreEx**Beschreibung**

Auslesen weiterer Daten zu einem Logbucheintrag, der mit System.LogbookStore abgerufen wurde. Die Datenart ist abhängig vom Diagnosetyp.

Item Syntax

System.LogbookStoreEx,(1),(2)

1. Diagnosetyp; Wert, der bei System.LogbookStore in Spalte 0 zurückgegeben wurde

SCPDiagnosisType_MTX	0x01	MTX-Meldungen
SCPDiagnosisType_ProVi	0x02	ProVi-Meldungen
SCPDiagnosisType_IndraStep	0x04	Schrittketten-Meldungen
SCPDiagnosisType_All	-1	alle Diagnosetypen

2. eindeutige Meldungs-ID; Wert, der bei System.LogbookStore in Spalte 7 zurückgegeben wurde

Read

Die Spaltenanzahl richtet sich nach dem Diagnosetyp der Meldung. In diesem Fall gibt es keinen einheitlichen Teil des Ergebnisses.

Beispiel**Datenformat****Spalten einer MTX-Meldung**

Wird als Diagnosetyp "SCPDiagnosisType_MTX" zurückgegeben, so sind die Spalten wie folgt definiert.

System.LogbookStoreEx, MTX-Meldung

row, col	Typ	Beschreibung	Bsp.
0,0	string	Ursachentext	
0,1	string	Behebungstext	

Spalten einer ProVi-Meldung

Wird als Diagnosetyp "SCPDiagnosisType_ProVi" zurückgegeben, so sind die Spalten wie folgt definiert.

System.LogbookStoreEx, ProVi-Meldung

row, col	Typ	Beschreibung	Bsp.
0,0	string	Ursachentext aus dem Message-Integrator (z.Z. noch leer)	
0,1	string	Behebungstext aus dem Message-Integrator (z.Z. noch leer)	
0,2	string	Anwender-Notizen aus dem Message-Integrator (z.Z. noch leer)	
0,3	string	HTML-Link aus dem Message-Integrator (z.Z. noch leer)	
0,4	long	Typ der POE, in der die Meldung programmiert ist. 1 = Programm, 2 = Funktionsblock	
0,5	string	Name der POE, in der die Meldung programmiert ist	
0,6	long	Typ des Details, in dem die Meldung programmiert ist 1 = Implementation, 2 = Aktion, 3 = Transition	
0,7	string	Name des Details, leer bei Implementation	
0,8	long	Netzwerknummer oder Zeile, in der die Meldung programmiert ist	
0,9	string	Variable, die als Zuweisung für die ProVi-Meldung verwendet wird	
0,10	string	POE Instanzen-Name	

Spalten einer Schrittketten-Meldung

Wird als Diagnosetyp "SCPDiagnosisType_IndraStep" zurückgegeben, so sind die Spalten wie folgt definiert.

System.LogbookStoreEx, Schrittketten-Meldung

row, col	Typ	Beschreibung	Bsp.
0,0	long	Typ der POE, in der die Schrittkette programmiert ist. 1 = Programm, 2 = Funktionsblock	
0,1	string	Name der POE, in der die Schrittkette programmiert ist.	
0,2	string	Schritt, aus dem die Meldung kommt. Nur bei Meldungsarten "aktiver Schritt" und "Zeitfehler"	
0,3	long	Typ des Details, aus dem die Meldung kommt. 2 = Aktion, 3 = Transition. Nur bei Meldungsarten "Monitorfehler" und "Zeitfehler". Bei "Monitorfehler" immer "Aktion".	
0,4	string	Name der Aktion, aus der die Meldung kommt. Nur bei Meldungsarten "Monitorfehler", "Zeitfehler" und "Detail" = "Aktion".	

row, col	Typ	Beschreibung	Bsp.
0,5	long	Netzwerknummer oder Zeile, in der die Meldung programmiert ist. Nur bei Meldungsart "Monitorfehler".	
0,6	string	Kriterienanalyse String. Nur bei Meldungsarten "Monitorfehler", "Zeitfehler", "Startvoraussetzung", "Grundstellung" und "Diagnosetyp".	

System.LogbookClear

Beschreibung Logbucheinträge löschen

Das Item hat keine Parameter. Welche Einträge gelöscht werden sollen, wird beim Write angegeben.

Anmerkung Wenn Zeitbereiche angegeben werden, kann es vorkommen, dass von einer Meldung der Kommen-Eintrag gelöscht wird und der Gehen-Eintrag erhalten bleibt, oder auch umgekehrt.

Item Syntax **System.LogbookClear**

Write Löschen der angegebenen Einträge

Datenformat **System.LogbookClear**

row, col	Typ	Beschreibung	Bsp.
0,0	long string	Filter Diagnosetyp, es können mehrere Diagnosetypen gleichzeitig zurückgesetzt werden.	-1
0,1	long string	Filter Grobklassifikation, es können mehrere Klassifikationen gleichzeitig zurückgesetzt werden.	0x01
0,2	long string	Filter Fehlerquelle, es können mehrere Fehlerquellen gleichzeitig angegeben werden.	2+3+5
0,3	long string	Zeitstempel, ab wann die Einträge gelöscht werden sollen. Meldung mit Datum und Uhrzeit vom Typ Variant Time. Die Zeit ist angegeben in Coordinated Universal Time (UTC). Es werden nur die Einträge gelöscht, die neuer oder gleich alt sind. Wird 0 angegeben, so werden alle Einträge gelöscht, deren Zeitstempel älter oder gleich dem Parameter in 4 ist. Es ist dann nicht festgelegt, wie alt der zu löschende Eintrag höchstens sein darf.	
0,4	double string	Zeitstempel, bis zu dem die Einträge gelöscht werden sollen. Meldung mit Datum und Uhrzeit vom Typ Variant Time. Die Zeit ist angegeben in Coordinated Universal Time (UTC). Es werden alle Einträge gelöscht, die älter oder gleich alt sind. Wird 0 angegeben, so werden alle Einträge gelöscht, deren Zeitstempel neuer oder gleich dem Parameter in 3 ist. Es ist dann nicht festgelegt, wie alt der zu löschende Eintrag mindestens sein muss.	

Items

Filter Diagnosetyp	SCPDiagnosisType_MTX	0x01	MTX-Meldungen
	SCPDiagnosisType_ProVi	0x02	ProVi-Meldungen
	SCPDiagnosisType_IndraStep	0x04	Schrittketten-Meldungen
	SCPDiagnosisType_All	-1	alle Diagnosetypen
Filter Grobklassifikation	SCPDiagnosisClass_Error	0x01	Fehler
	SCPDiagnosisClass_Info	0x02	Hinweis
	SCPDiagnosisClass_Warning	0x04	Warnung
	SCPDiagnosisClass_Startup	0x08	Startvoraussetzung
	SCPDiagnosisClass_Setup	0x10	Einrichtdiagnose
Filter Fehlerquelle	Es wird direkt die angeforderte Modulnummer angegeben Sollen Daten für mehrere Module angefordert werden, so werden die entsprechenden Modulnummern durch Pluszeichen verbunden Sollen die Daten für alle Module angefordert werden oder sind keine Module definiert, so wird "-1" angegeben.		

System.LogbookDevices

Beschreibung	Es werden die logischen Geräte ausgegeben, die ihre Meldungen im Logbuch protokollieren.
Item Syntax	System.LogbookDevices
Write	Gerätename und GUID
Datenformat	System.LogbookDevices

row, col	Typ	Beschreibung	Bsp.
0,0	string	Name des logischen Gerätes	
0,1	string	GUID des logischen Gerätes	

System.LogbookStart

Beschreibung	startet die Protokollierung der Logbuchdaten Die Daten werden nur in das Logbuch geschrieben, wenn für die entsprechende Steuerung ein allgemeines Diagnose-Item angemeldet ist. Sollen Logbuchdaten protokolliert werden, ohne dass bereits ein anderes Item angemeldet ist, so kann dieses Item dazu verwendet werden.
Anmerkung	Dieses Item kann weder geschrieben noch gelesen werden. Es muss nur angelegt werden.
Item Syntax	System.LogbookStart,(1) Filter Steuerung -1: es werden die Meldungen aller im Logbuch konfigurierten Steuerungen protokolliert. Sollen nur die Meldungen einer bestimmten Steuerung protokolliert werden, so kann hier die GUID angegeben werden.

4.6 IndraLogic**4.6.1 Übersicht**

Item-Überblick • **Plc.PVT**

- Liest Symbolinformation eines SPS-Projekts
- **Plc.PVL**
Liest und schreibt Variablenwerte
- **Plc.PVA**
Liest die Adressen von symbolischen SPS-Variablen
- **Plc.State2**
Zustandsinformation der IndraLogic-Gerätekomponente
- **Plc.PVF**
Symbolinformation einer SPS-Variablen
- **Plc.ModStatus**
Status des DP-Masters
- **Plc.ModDiagnose**
Diagnose eines DP-Slaves
- **Plc.PVLK**
Zugriff auf SPS Variablen (Strukturen, Arrays)
- **Plc.FirmwareVersion**
Lesen der Firmwareversion der SPS
- **Plc.FirmwareConnection**
Status der Kommunikation mit der SPS
- **Plc.FirmwareReboot**
Auslösen des SPS-Rebootvorgangs
- **Plc.FirmwareStatus**
Status der SPS
- **Plc.Plc.PVBS**
Zugriff auf SPS Variablen (Strukturen, Arrays) als Bytestream
- **Plc.StartStop**
Umschalten der SPS in RUN/STOP
- **Plc.FeldBus**
Tunneling der Feldbusdaten

4.6.2 Firmware

Plc.FirmwareConnection

Beschreibung	Prüft in Zeitabständen von einer Sekunde, ob die Kommunikation mit der SPS ansteht.
Anmerkung	Item wird beim Reboot benutzt. Gleiche Funktionalität wie Item Firmware.ResumeConnection
Item Syntax	Plc.FirmwareConnection
Write	

Plc.FirmwareReboot

Beschreibung	Startet den Reboot-Vorgang in der SPS
Anmerkung	Während des Rebootvorgangs gibt es keine Möglichkeit mit der SPS zu kommunizieren! Gleiche Funktionalität wie Item Firmware.Reboot

Item Syntax **Plc.FirmwareReboot**
Write

Plc.FirmwareStatus

Beschreibung Stellt fest, ob der Status der SPS erlaubt, den Downloadvorgang zu starten
Anmerkung Gleiche Funktionalität wie Item **Firmware.CheckDownloadRequirements**
Item Syntax **Plc.FirmwareStatus**
Read SPS läuft: "false" wird geliefert
 SPS steht im STOP: "true" wird geliefert
Datenformat **Plc.FirmwareStatus, read**

row, col	Typ	Beschreibung	Bsp.
0,0	string	Firmware-Status	false
0,1	long	???	1

Plc.FirmwareVersion

Beschreibung Ermittelt die aktuelle Firmwareversion der SPS
Anmerkung Gleiche Funktionalität wie Item **Firmware.DeviceFirmware**
Item Syntax **Plc.FirmwareVersion**
Read
Datenformat **Plc.FirmwareVersion, read**

row, col	Typ	Beschreibung	Bsp.
0,0	string	Firmware-Version	FWC-ILsL40-RUN-03T00.00-NN

4.6.3 Info

Plc.ModDiagnose

Beschreibung Ermittelt die Diagnose eines DP-Slaves.
Anmerkung Die Diagnose gilt für einen Slave des DP-Masters.
Item Syntax **Plc.ModDiagnose,(1),(2),(3),(4),(5),(6),(7),(8)**
 1. Master-Id (z.Z. 2000)
 2. Mastertyp-Id (z.Z. 1)
 3. Mastertyp-Index (z.Z. 0)
 4. Masterbusadresse
 5. Slave-Id (z.Z. 2001)
 6. Slavetyp-Id (z.Z. 1)
 7. Slavetyp-Index (z.Z. 0)
 8. Slavebusadresse
Read *Liefert u.a. Fehlerinformationen:*

- Fehlerinfo-Flag 0 = Fehlerinfo ist ein String
- Fehlerinfo-Flag 1 = Fehlerinfo ist ein binäres Array

Datenformat Spaltenanzahl: n=Länge des Arrays der Fehlerinfo, für Zeile 5

Plc.ModDiagnose, read

row, col	Typ	Beschreibung	Plc.ModDiagnose, 2000,1,0,1,2001,1,0,3
0,0	WORD	Resultat-Flag (soll 0 sein)	0
1,0	DWORD	Busnode-Id	3
2,0	DWORD	Fehler-Id	344
3,0	WORD	Fehlerinfo-Flag	1
4,0	WORD	Länge des Arrays von Fehlerinfo	100
5,n	COM	Fehlerinfo als String (VT_BSTR) oder Array (VT_UI1)	255

Plc.ModStatus

Beschreibung Ermittelt den aktuellen Status des DP-Masters.

Item Syntax **Plc.ModStatus,(1),(2),(3),(4)**

1. Master-Id (z.Z. 2000)
2. Mastertyp-Id (z.Z. 1)
3. Mastertypindex (z.Z. 0)
4. Masterbusadresse

Read *Liefert den Zustand des DP-Masters und der am Master angeschlossenen Slaves:*

- Status 1 = OK
- Status 2 = fault
- Status 3 = not communicating
- Status 4 = stopped

Datenformat Spaltenanzahl: n=130 für Zeile 3 und Zeile 4 (eine Spalte pro möglichem Busteilnehmer)

Plc.ModStatus, read

row, col	Typ	Beschreibung	Plc.ModStatus,2000,1,0,1
0,0	WORD	Resultat-Flag (soll 0 sein)	0
1,0	WORD	Anzahl der Busteilnehmer	2
2,0	WORD	Busmasterstatus	1
3,n	WORD	Node-Id des n. möglichen Busteilnehmers	2
4,n	WORD	Status des n. möglichen Busteilnehmers	5

Plc.State2

Beschreibung Zustandsinformation der Gerätekomponente und direkte Weitergabe der Zustände des PLCHandlers.

Anmerkung Der Aufruf "AsyncRead" mit "SCPCacheChanged" liefert nur bei Zustandsänderung ein Ergebnis.

Item Syntax **Plc.State2**

Read Datenformat

row/col	Typ	Beschreibung
0,0	long	Status
0,1	string	textuelle Statusbeschreibung

Tab. 4-29: *Read Datenformat*

Verbindungszustände Die nachfolgende Tabelle zeigt die Zustände der SPS Verbindung:

Verbindungszustände

Zustand	textuelle Kurzbeschreibung	Beschreibung
-1	STATE_TERMINATE	PLCHandler ist beendet
0	STATE_PLC_NOT_CONNECTED	Keine Verbindung zur SPS (Initialisierungszustand)
1	STATE_PLC_CONNECTED	Verbindung zur SPS
2	STATE_NO_SYMBOLS	Keine Symbole verfügbar oder Projektwechsel
3	STATE_SYMBOLS_LOADED	Verbindung zur SPS und Symbole sind geladen.
4	STATE_RUNNING	Verbindung zur SPS, Symbole sind geladen und diese sind verifiziert.
5	STATE_DISCONNECT	Verbindungsunterbrechung
6	STATE_NO_CONFIGURATION	Konfiguration des PLCHandlers ist ungültig.
7	STATE_PLC_NOT_CONNECTED_SYMBOLS_LOADED	SPS ist nicht verfügbar, aber die Symbole stehen "offline" zur Verfügung. Der PLCHandler versucht ständig eine Verbindung zur SPS zu erstellen.

Tab. 4-30: *State Machine*

4.6.4 Eingriff

Plc.StartStop

Beschreibung Schaltet SPS in RUN oder in STOP

Item Syntax **Plc.StartStop**

Write 0: Stop

1: Run

Datenformat **Plc.StartStop, write**

row, col	Typ	Beschreibung	Bsp.
0,0	long	Stop oder Run	1

4.6.5 Feldbus

Plc.FeldBus

Beschreibung	Tunnelt Antriebsdaten via SPS zum Profibus-Gateway
Anmerkung	Das Item soll die bisherige Vorgehensweise via serielle Schnittstelle ergänzen oder ersetzen.
Item Syntax	Plc.FeldBus,(1),(2),(3),(4),(5) 1. PB (Daten als BSTR als HexASCII-Bytefolge) oder PBbin (Daten als BSTR als binäre Bytefolge) 2. ProfibusMaster-ID (z. B. 0 = Profibus-DP/M OnBoard) 3. Profibus-Slave-Adresse 4. Slot-Index im Slave 5. Index im Slot
Read	
Write	
Datenformat	Plc.FeldBus, read

row, col	Typ	Beschreibung	Plc.FeldBus,PB,0,3,0,3
0,0	VT_BS TR	Nutzdaten als BSTR	123456

4.6.6 Variablen

Plc.PVA

Beschreibung	Liest die Adressen von symbolischen Variablen
Anmerkung	Das Erzeugen der Symbolinformationen muss bei einem SPS-Projekt aktiv sein. Dies wird im Menü Projekt ► Optionen in der Auswahl "Symbolkonfiguration" eingestellt.
Item Syntax	Plc.PVA,(1),..., (n) 1...n: symbolische Variablennamen
Read	Der Adresswert von ungültigen symbolischen Variablen ist 0, es wird kein Fehler erzeugt.
Datenformat	Zeilenanzahl: entspricht der Anzahl der SPS-Variablen Plc.PVA, read

row, col	Typ	Beschreibung	Bsp.
n,0	long	Adresse	30120947

Plc.PVArray

Beschreibung	Liest die aktuellen Werte einer Array-Variablen aus der SPS.
Anmerkung	Array-Variablen entsprechen den z. B. PVL-Variablen.
Item Syntax	Plc.PVArray,(1) 1. symbolischer Variablenname ohne Index
Read	
Datenformat	kompletter Inhalt byteweise

Plc.PVArray, read

row, col	Typ	Beschreibung	Plc.PVArray,myArray Beispiel Variable myArray besteht aus vier Byte-Elementen
0,0	COM	aktueller Wert der Variablen myArray (Byte 0)	01
0,1	COM	aktueller Wert der Variablen myArray (Byte 1)	02
0,2	COM	aktueller Wert der Variablen myArray (Byte 2)	03
0,3	COM	aktueller Wert der Variablen myArray (Byte 3)	04

Plc.PVBS

Beschreibung Liest Variablenwerte aus der SPS. Der symbolische Name der SPS-Variablen wird als Parameter angegeben. Zu den Variablen gehören hier nur Strukturen und Arrays. Um die einzelnen Elemente solcher Variablen zu lesen oder zu schreiben, muss das Item "Plc.PVL" benutzt werden.

Anmerkung Das Erzeugen der Symbolinformationen muss bei einem SPS-Projekt aktiv sein. Dies wird im Menü **Projekt ► Optionen** in der Auswahl "Symbolkonfiguration" eingestellt. Es müssen alle entsprechenden Punkte aktiviert sein.

Wird das Item in einem OPC-Client verwendet, darf nur eine SPS-Variable pro Item bearbeitet werden.



Unter **Indralogic 2G** ist kein direkter Speicherzugriff erlaubt. Daher ist bei Verwendung des Items unter IL2G zu berücksichtigen, dass ggf. **nicht alle Daten in einem SPS-Zyklus** geschrieben sind.

Item Syntax **Plc.PVBS,PLC_PRG.myStrukt1**

myStrukt1 Variablen-Name vom Datentyp "MyStruct" – eine vom Anwender definierte Struktur, z. B.:

```
TYPE MyStrukt :
```

```
STRUCT
```

```
x: UINT;
```

```
y: UINT;
```

```
END_STRUCT
```

```
END_TYPE
```

Länge der Struktur in Byte: n=4

Plc.PVBS,PLC_PRG.myArr1

myArr1 Variablen-Name vom Datentyp "ARRAY" – eine vom Anwender definierte Struktur mit folgenden Elementen:

Read

Write

Datenformat Spaltenanzahl: Anzahl Bytes der Struktur

Plc.PVBS, read

row, col	Typ	Beschreibung	Plc.PVBS,PLC_PRG.myStrukt1 (x := 3333, y :=4444)
0,n	byte	Byte der Struktur	92

Plc.PVF

Beschreibung Liest die Symbolinformationen einer SPS-Variablen

Item Syntax **Plc.PVF,(1)**

1. SPS-Variablenname

Read

Datenformat **Plc.PVF, read**

row, col	Typ	Beschreibung	Plc.PVF,PLC_TYPEN.bool1
0,0	string	Variablenname	PLC_TYPEN.bool1
0,1	string	Typ	BOOL
0,2	long	Ref-ID	4
0,3	long	Offset	1
0,4	long	Größe	1
0,5	string	Zugriff	b

Plc.PVL

Beschreibung Liest und schreibt Variablenwerte. Die symbolischen Namen der SPS-Variablen werden als Parameter angegeben.

Anmerkung Das Erzeugen der Symbolinformationen muss bei einem SPS-Projekt aktiv sein. Dies wird im Menü **Projekt ► Optionen** in der Auswahl "Symbolkonfiguration" eingestellt.

Die Werte der symbolischen Variablen werden in COM-Variablentypen konvertiert. Arrays und Strukturen werden nicht unterstützt:

Konvertierung

SPS-Variablen-Typ	COM-Variablen-Typ
USINT	VT_UI1
BYTE	VT_UI1
SINT	VT_I1
INT	VT_I2
WORD	VT_UI2
UINT	VT_UI2
DINT	VT_I4
DWORD	VT_UI4
UDINT	VT_UI4
DATE	VT_UI4
DT (DATE and TIME)	VT_UI4
TOD (Time of Day)	VT_UI4
BOOL	VT_BOOL

SPS-Variablen-Typ	COM-Variablen-Typ
REAL	VT_R4
LREAL	VT_R8
STRING	VT_BSTR

Hier nicht aufgeführte SPS-Variablen werden entsprechend der Typgröße in VT_UI1, VT_UI2 und VT_UI4 konvertiert.

Item Syntax **Plc.PVL,(1),...,(n)**

1...n: symbolische Variablennamen

Read

Write

Datenformat Zeilenanzahl: entspricht der Parameteranzahl

Plc.PVL, read

row, col	Typ	Beschreibung	Plc.PVL,PLC_TYPEN.bool,PLC_TYPEN.real
0,0	bool	Wert der 1. angegebenen Variablen	TRUE
...			
n,0	long	Wert der n. angegebenen Variablen	3.14

Plc.PVLK

Beschreibung Liest Variablenwerte aus der SPS. Die symbolischen Namen der SPS-Variablen werden als Parameter angegeben. Zu den Variablen gehören hier sowohl die "einfachen" Variablen als auch Strukturen und Arrays.

Anmerkung Man kann Variablennamen von allen im SPS-Programm definierten Datentypen angeben.

Item Syntax **Plc.PVLK,PLC_PRG.myStrukt1,PLC_PRG.myArr1**

***myStrukt1** Variablen-Name vom Datentyp "MyStruct" – eine vom Anwender definierte Struktur mit folgenden Elementen:*

- **myArr2:** Array [1..2] OF BYTE
- **x:** BOOL-Variable
- **y:** DWORD-Variable

myArr1 Variablen-Name vom Datentyp ARRAY [1..2] OF UINT

Read

Datenformat Zeilenanzahl: Anzahl der vorhandenen komplexen Variablen zzgl. der in den komplexen Variablen vorhanden Elementardaten.

Plc.PVLK, read

row, col	Typ	Beschreibung	Plc.PVLK,PLC_PRG.myStrukt1,PLC_PRG.myArr1
0,0	string	symbolischer Name der komplexen Variablen	PLC_PRG.myStrukt1
0,1	COM	Datentyp	DATA
0,2	long	Bytelänge der Variablen	7
0,3	COM	Datenwert	DATA

row, col	Typ	Beschreibung	Plc.PVLK,PLC_PRG.myStrukt1,PLC_PRG.myArr1
1,0	string	symbolischer Name des 1. Elements der komplexen Variablen (0,0)	PLC_PRG.myStrukt1.myArr2
1,1	COM	Datentyp	ARRAY
1,2	long	Bytelänge des Elements	2
1,3	COM	Datenwert	ARRAY
2,0	string	symbolischer Name des 1. Elements des 1. Variablen-Elements (1,0)	PLC_PRG.myStrukt1.myArr2[1]
2,1	COM	Datentyp	BYTE
2,2	long	Bytelänge des Elements	1
2,3	COM	Datenwert	201
3,0	string	symbolischer Name des 2. Elements des 1. Variablen-Elements (1,0)	PLC_PRG.myStrukt1.myArr2[2]
3,1	COM	Datentyp	BYTE
3,2	long	Bytelänge des Elements	1
3,3	COM	Datenwert	202
4,0	string	symbolischer Name des 2. Elements der komplexen Variablen (0,0)	PLC_PRG.myStrukt1.x
4,1	COM	Datentyp	BOOL
4,2	long	Bytelänge des Elements	1
4,3	COM	Datenwert	wahr
5,0	string	symbolischer Name des 3. Elements der komplexen Variablen (0,0)	PLC_PRG.myStrukt1.y
5,1	COM	Datentyp	DWORD
5,2	long	Bytelänge des Elements	4
5,3	COM	Datenwert	???

Plc.PVT

Beschreibung Liest die Symbolinformationen eines SPS-Projektes.

Anmerkung Das Erzeugen der Symbolinformationen muss bei einem SPS-Projekt aktiv sein. Dies wird im Menü **Projekt ► Optionen** in der Auswahl "Symbolkonfiguration" eingestellt..

Item Syntax **Plc.PVT**

Read

Datenformat Zeilenanzahl: entspricht der Anzahl der SPS-Variablen

Plc.PVT

row, col	Typ	Beschreibung	Bsp.
n,0	string	Name	PLC_TYPEN.bool1
n,1	string	Typ	BOOL

row, col	Typ	Beschreibung	Bsp.
n,2	long	Referenz-ID	4
n,3	long	Offset	1
n,4	long	Größe	1
n,5	string	Zugriff	b

4.7 NC - global

4.7.1 Allgemeine Informationen

NC.AxisNamesPhys

Beschreibung	Liefert die physikalische Achsbezeichnung aller parametrisierten Achsen
Item Syntax	NC.AxisNamesPhys
Aufruf	synchron / asynchron
Zugriff	Read
zyklisch (opt.)	ja
Datenformat	NC.AxisNamesPhys - read

row, col	Typ	Beschreibung	Bsp.
0,0	string	Bezeichnung der 1. Achse	K1X

NC.BitInterface

Beschreibung	Bietet Zugriff auf die Signale der Bit-Interface-Ein-Ausgänge. Neben dem aktuellen Zustand der Ein- und Ausgänge wird die Funktion und die jeweilige Bezeichnung des Signals geliefert. Über Parameter werden der Interface-Typ (globales Interface, Kanal-Interface, Achs-Interface, Spindel-Interface) und die gewünschte Instanz (Kanalnummer, Achsnummer, Spindelnummer) vorgegeben.
Anmerkung	Die Anzahl der gelieferten Einträge variiert je nach angefordertem Interface-Bereich. In der ersten Zeile der gelesenen Daten wird deshalb die Anzahl der Ausgänge und Eingänge geliefert. Die weiteren Zeilen enthalten die SPS-Ausgänge gefolgt von den SPS-Eingängen. Die Signal-Beschreibungen werden in der eingestellten Fremdsprache geliefert. Zur Vorgabe des Interface-Bereichs kann der Enumerator "LDX_AcblfType" verwendet werden:

Enumerator LDX_AcblfType

LDX_AcblfType	Beschreibung
LDX_ACB_IFTYPE_CHAN_C	Kanal-Interface
LDX_ACB_IFTYPE_AXIS_C	Achs-Interface
LDX_ACB_IFTYPE_SPINDLE_C	Spindel-Interface
LDX_ACB_IFTYPE_GENERAL_C	Globales Interface

Item Syntax	NC.BitInterface,(1),(2)
--------------------	--------------------------------

1. Typ - Interface-Bereich (0=Kanal-, 1=Achs-, 2=Spindel-, 3=globales Interface)

2. Index - für Kanal, Achse oder Spindel

Aufruf synchron / asynchron

Zugriff Read

zyklisch (opt.) ja

Datenformat NC.BitInterface, read

row, col	Typ	Beschreibung	NC.BitInterface,0,3 Kanal-Interface von Kanal 3 anfordern
0,0	long	Anzahl SPS-Ausgänge	112
0,1	long	Anzahl SPS-Eingänge	112
1,0	string	Funktion des 1. Ausganges	Betriebsart Anwahl Bit0
1,1	string	Bezeichnung des Signals	qCh_OpModeSel_00
1,2	long	aktueller Zustand des 1. Ausganges	1
...	...	Informationen zu allen weiteren Ausgängen	...
x,0	string	Funktion des 1. Eingangs	Aktive Betriebsart Bit0
x,1	string	Bezeichnung des Signals	iCh_OpMode_00
x,2	long	aktueller Zustand des 1. Eingangs	1
...	...	Informationen zu allen weiteren Eingängen	...

NC.ConnectionQuality

Beschreibung Liefert den aktuellen Verbindungszustand (good / bad) sowie die aktuelle Prozessorauslastung in Prozent.

Anmerkung *Der Enumerator "LDX_ConnectionQuality" steht zur Auswertung des gelieferten Verbindungszustands zur Verfügung und kann folgende Werte annehmen:*

- LDXConnectionQualityBad - keine Verbindung zum MTX-Kern
- LDXConnectionQualityGood - Verbindung zum MTX-Kern vorhanden

Item Syntax NC.ConnectionQuality

Aufruf synchron / asynchron

Zugriff Read

zyklisch (opt.) ja

Datenformat NC.ConnectionQuality - read

row, col	Typ	Beschreibung	Bsp.
0,0	long	Verbindungszustand 2 = Kommunikationsverbindung 0 = keine Verbindung	2
0,1	long	Prozessorauslastung in Prozent	15

NC.CpuUsage2

Beschreibung Liefert die aktuelle CPU-Auslastung der NC

Anmerkung Für verschiedene Tasks bzw. Taskgruppen werden jeweils folgende Informationen zur Verfügung gestellt:

- Kennung
- aktuelle CPU-Auslastung in Prozent
- Bewertung der CPU-Auslastung
- Kennung des CPU-Cores (1..n)
- englische Kurzbezeichnung

Der Enumerator "LDX_CpuUsageEntryId" liefert die Kennung der Task bzw. Taskgruppe:

Enumerator LDX_CpuUsageEntryId

LDX_CpuUsageEntryId	Beschreibung
LDX_CPU_USAGE_IPO_C	Interpolator
LDX_CPU_USAGE_PLC_C	SPS-Anwendertasks und SPS-Systemtasks
LDX_CPU_USAGE_SAV_CPL_C	Satzverarbeitung und CPL
LDX_CPU_USAGE_SYS_C	sonstige Tasks (Kommunikation, Fileserver, Datenbank)
LDX_CPU_USAGE_IDLE_C	Idle Task (freie Kapazität)

Der Enumerator "LDX_CpuUsageEval" liefert die Bewertung der CPU-Auslastung durch die Task bzw. Taskgruppe:

Enumerator LDX_CpuUsageEval

LDX_CpuUsageEval	Beschreibung
LDX_CPU_USAGE_EVAL_OK_C	Normale Last
LDX_CPU_USAGE_EVAL_HIGHLOAD_C	Hohe Last
LDX_CPU_USAGE_EVAL_OVERLOAD_C	Überlast
LDX_CPU_USAGE_EVAL_INVALID_C	Keine Bewertung möglich (Kompatibilität)



Steuerungen mit älteren Softwareständen liefern für:

- die Bewertung der CPU-Auslastung >>> LDX_CPU_USAGE_EVAL_INVALID_C
- die Kennung des CPU-Cores >>> 0

Item Syntax NC.CpuUsage2

Aufruf synchron / asynchron

Zugriff Read

zyklisch (opt.) ja

Datenformat Zeilenanzahl: Anzahl der Tasks bzw. Taskgruppen

NC.CpuUsage2 - read

row, col	Typ	Beschreibung	Bsp.
n,0	long	Kennung	LDX_CPU_USA-GE_IPO_C
n,1	long	Prozentuale CPU-Auslastung	5
n,1	long	Bewertung der CPU-Auslastung	LDX_CPU_USA-GE_EVAL_OK_C
n,1	long	Kennung des CPU-Cores	1
n,2	string	englische Kurzbezeichnung	IPO

NC.DryRunMode

Beschreibung Gibt den Status der Gesamtsperre wieder.

Item Syntax **NC.DryRunMode**

Aufruf synchron / asynchron

Zugriff Read

zyklisch (opt.) ja

Datenformat **NC.DryRunMode, read**

row, col	Typ	Beschreibung	Bsp.
0,0	bool	Status der Gesamtsperre	TRUE

NC.Filesystem

Beschreibung *Liefert Informationen interner und angebundener File-Systeme der Steuerung:*

- Anzahl der eingebundenen File-Systeme
- Typ der gemounteten File-Systeme (PxFile,NFS)
- Mount-Bedingungen dieser File-Systeme
- Name des Verzeichnisses (mount point)
- *Name des angebundenen File-Systems:*
 - bei lokalem File-System: Gerätenamen
 - bei NFS: Host-Name des NFS-Servers und Name des Verzeichnisses, das der NFS-Server exportiert

Anmerkung Der Enumerator "LDX_MountType" liefert den Typ des gemounteten Filesystems:

Enumerator LDX_MountType

LDX_MountType	Beschreibung
LDX_FSR_FSMOUNT_UNKNOWN_C	unbekanntes File-System
LDX_FSR_FSMOUNT_LOCAL_C	lokales PxFile-File-System
LDX_FSR_FSMOUNT_NFS_C	NFS-File-System
LDX_FSR_FSMOUNT_NOFOUND_C	kein File-System gefunden

Der Enumerator "LDX_MountMode" liefert die Art der File-System-Anbindung.

Enumerator LDX_MountMode

LDX_MountMode	Beschreibung
LDX_FSR_FSMOUNT_RDONLY_C	nur lesbar
LDX_FSR_FSMOUNT_CACHABLE_C	Zugriff wird "ge-cached"
LDX_FSR_FSMOUNT_FEPRMWRI-TEABLE_C	beschreibbares FEPRM-Filesystem
LDX_FSR_FSMOUNT_RDWR_C	lesen und schreiben

Item Syntax NC.Filesystem**Aufruf** synchron / asynchron**Zugriff** Read**zyklisch (opt.)** nein**Datenformat** Zeilenanzahl: Anzahl der eingebundenen File-Systeme**NC.Filesystem, read**

row, col	Typ	Beschreibung	Lesen aller File-System-Informationen
n,0	long	Typ des gemounteten File-Systems	LDX_FSR_FSMOUNT_NFS_C
n,1	long	Mount-Bedingung des File-Systems	LDX_FSR_FSMOUNT_RDONLY_C
n,2	string	Name des Verzeichnisses	/mnt
n,3	string	Name des File-Systems	

NC.CheckFile**Beschreibung** Liefert Informationen über eine Datei. Über den 2. Parameter (Modus) wird gesteuert, welche Information erfragt wird.**Anmerkung** Folgende Modi werden unterstützt:

0 - Ist die Datei verschlüsselt?

1 - Wie lautet der Verschlüsselungskommentar?

Item Syntax NC.CheckFile,(1),(2)

1. Dateiname mit Pfad

2. Modus

Aufruf synchron / asynchron**Zugriff** Read**zyklisch (opt.)** nein**Datenformat** **NC.CheckFile, read**

row, col	Typ	Beschreibung	Bsp.: NC.CheckFile,/usr/user/sekt.cnc,0
0,0	long	0 = Datei ist nicht verschlüsselt 1 = Datei ist verschlüsselt	1

NC.CheckFile, read

row, col	Typ	Beschreibung	Bsp.: NC.CheckFile,/usr/user/ sekt_crypt.cnc,1
0,0	string	Kommentar aus der Datei	** Dateiversion 1.1.1 **

NC.FIS

Beschreibung	Liefert einen Unknown-Zeiger auf die Instanz des Fis-Servers (IID_IFIS).
Anmerkung	Dieser Zeiger ermöglicht den Zugriff auf die drei Schnittstellen des Fis-Servers "IID_IFISDirectory", "IID_IFISFile" und "IID_IFISFileSystem".
Item Syntax	NC.FIS
Aufruf	synchron / asynchron
Zugriff	Read
zyklisch (opt.)	nein
Datenformat	NC.FIS

row, col	Typ	Beschreibung	Bsp.
0,0	string	Unknown-Zeiger	LPUNKNOWN

NC.MtxCommand

Beschreibung	Das Item liefert Statusinformationen oder bewirkt Zustandsänderungen in der Steuerung je nach Kommandovorgabe.
---------------------	--



Das Item gilt nur für CMLxx-Steuerungen

Item Syntax	NC.MtxCommand
--------------------	----------------------

Aufruf asynchron

Zugriff Read

- Lesen von verschiedenen Statusinformationen als XML-String



Einschränkung: Die Read-Funktion ist für eine zyklische Abfrage optimiert. Deshalb darf der Lesezugriff auf das Item nur innerhalb einer Gruppe erfolgen. Die ersten Lesezugriffe können "SCPQualityUncertain" liefern.

Write

- Schreiben verschiedener Kommandos für Steuerungsfunktionen, die über ein Zusatzkommando variiert werden können

zyklisch (opt.) nur lesend

Datenformat

NC.MtxCommand, read

row, col	Typ	Beschreibung möglicher Status-Informationen	Bsp.
0,0	string (XML-Stream)	StartupPhase - aktuelle Startup-Phase StartupMode - eingestellter Startup-Mode Ready 0 = kein Ready 1 = Ready aktiv -1 = undefiniert PlcState - Zustand SPS 0 = SPS steht 1 = SPS läuft -1 = undefiniert FieldBus - Zustand Feldbus 0 = Feldbus inaktiv 1 = Feldbus aktiv -1 = undefiniert ScsState - aktuelle Sercos Phase CritSysState - Zustand kritischer Systemfehler 0 = kein Fehler 1 = Fehler -1 = undefiniert ShutdownState - Zustand Shutdown der Steuerung 0 = kein Shutdown 1 = Shutdown aktiv 2 = Shutdown beendet -1 = undefiniert BootPanicState - Zustand Boot Panic Fehler 0 = kein Fehler 1 = Fehler 2 = Fehler Bootlader -1 = undefiniert BootPanicError - Boot Panic Fehler-Nr Battery - Zustand Pufferbatterie 0 = defekt 1 = in Ordnung -1 = undefiniert FanState - Zustand Lüfter 0 = defekt oder fehlt 1 = in Ordnung -1 = undefiniert TargetTimeS - UTC-Zeit in der Steuerung	<pre> <SysCtrl> <SysState> <StartupPhase>0</StartupPhase> <StartupMode>0</StartupMode> <Ready>1</Ready> <PlcState>1</PlcState> <FieldBus>0</FieldBus> <ScsState>4</ScsState> <CritSysState>0</CritSysState> <ShutdownState>0</ShutdownState> <BootPanicState>0</BootPanicState> <BootPanicError>0</BootPanicError> <Battery>1</Battery> <FanState>1</FanState> <TargetTimeS>1334736235</TargetTimeS> </SysState> </SysCtrl> </pre>

NC.MbxCommand, write

row, col	Typ	Beschreibung möglicher Kommandos und Zusatzoptionen	Bsp.: Kommando "4" + "6" "Startup mit Urladen"
0,0	long	Folgende Kommandos sind möglich: 1 od. 2 = Reset der Steuerung 3 = Shutdown der Steuerung 4 = Setzen des Startup-Mode 10 = Starten und Stoppen der SPS 11 = Coredump erzeugen	4
1,0	long	Für die Kommandos 1, 2 und 11 entfällt diese Vorgabe Optionen bei Kommando 3: 0 = nur Shutdown (default) 1 = Shutdown mit anschließendem Reset Optionen bei Kommando 4: Startup-Mode Wertebereich [0..15] setzen: 0 = Normalbetrieb 1 = SPS stopp 2 = Neuladen des SPS-Bootprojekts 3 = Abgesicherter Hochlauf 4 = Löschen der permanenten CPL-Variablen und der Systemdaten 5 = Kaltstart 6 = Urladen 7 = Neuanlegen des Benutzer-FEPROM-Filesystems 8 - 15 = intern Optionen bei Kommando 10: 1 = SPS starten 0 = SPS stoppen	6

NC.NofChannels

Beschreibung	Liefert die parametrisierte Kanalanzahl.
Item Syntax	NC.NofChannels keine Parameter
Aufruf	synchron / asynchron
Zugriff	Read
zyklisch (opt.)	nein

Datenformat NC.NofChannels

row, col	Typ	Beschreibung	Bsp.
0,0	long	Kanalanzahl	4

NC.OsciUtil

Beschreibung Das Item ermöglicht Zugriff auf die Oszilloskop-Funktionalität.**Anmerkung** Dem Anwender stehen 22 Funktionen zur Verfügung, die Initialisierungsoptionen bieten und das Lesen von Initialisierungsdaten sowie Aufzeichnungsdaten erlauben. Auch Statusinformationen über den augenblicklichen Zustand des Oszilloskops sind abrufbar.**Item Syntax** NC.OsciUtil

keine Parameter

Aufruf synchron / asynchron**Zugriff** Read

Lesen der momentan aktiven Statusinformation.

Folgende 5 Zustände sind definiert:

- 0 : NotReadyForRecording
- 1 : ReadyForRecording
- 2 : RecordingActive
- 3 : RecordingTriggered
- 4 : RecordingCompleted

Call

Lesen der durch Kommandos angeforderten Daten des Oszilloskops



Die Antwortdaten liefern in der ersten Zeile immer eine Fehlernummer. Alle folgenden Zeilen liefern Nutzdaten.

zyklisch (opt.) ja

Datenformat NC.OsciUtil, call - Kommando

row, col	Typ	Beschreibung	Bsp.: "Connect"
0,0	string	Verbindung zur Steuerung herstellen und Ressourcen reservieren	Connect

NC.OsciUtil, call - Antwortdaten

row, col	Typ	Beschreibung	Bsp.: "Connect" - mögliche Antwort
0,0	long	Mögliche Quittierungsmeldungen	0 = Verbindung steht -3 = keine Verbindung - Speicherreservierung fehlgeschlagen -4 = Verbindungsherstellung bereits erfolgt

NC.OsciUtil, call - Kommando

row, col	Typ	Beschreibung	Bsp.: "Disconnect"
0,0	string	Verbindung beenden und Ressourcen freigeben	Disconnect

NC.OsciUtil, call - Antwortdaten

row, col	Typ	Beschreibung	Bsp.: "Disconnect" - mögliche Antwort
0,0	long	Mögliche Quittierungsmeldungen	0 = kein Fehler

NC.OsciUtil, call - Kommando

row, col	Typ	Beschreibung	Bsp.: "Start"
0,0	string	Starten einer Aufzeichnung	Start

NC.OsciUtil, call - Antwortdaten

row, col	Typ	Beschreibung	Bsp.: "Start" - mögliche Antwort
0,0	long	Mögliche Quittierungsmeldungen	0 = 1 = Datei konnte nicht erzeugt werden 2 = Es steht kein Speicher zur Verfügung 3 = Speicher reserviert 4 = Fileserver Konflikt 5 = Startvorgang bereits eingeleitet 6 = Fehlerhafte Parameter 7 = Triggerbedingung fehlerhaft 8 = SCS-Kommunikationsfehler 10 = Unzulässiger Status für Start-Vorgabe 11 = Mind. ein Kanal muss ausgewählt sein 12 = Ausgewählte Kanäle müssen aktiv sein 13 = Fehlerhafte Aufzeichnungszeit 14 = Falscher Modus

NC.OsciUtil, call - Kommando

row, col	Typ	Beschreibung	Bsp.: "Stop"
0,0	string	Stoppen einer Aufzeichnung	Stop

NC.OsciUtil, call - Antwortdaten

row, col	Typ	Beschreibung	Bsp.: "Stop" - mögliche Antwort
0,0	long	Mögliche Quittierungsmeldungen	0 = Aufzeichnung fehlerfrei gestoppt Negative Werte lt. Fehlerbeschreibung s.u.

NC.OsciUtil, call - Kommando

row, col	Typ	Beschreibung	Bsp.: "Trigger"
0,0	string	Trigger auslösen	Trigger

NC.OsciUtil, call - Antwortdaten

row, col	Typ	Beschreibung	Bsp.: "Trigger" - mögliche Antwort
0,0	long	Mögliche Quittierungsmeldungen	0 = Trigger ausgeführt Negative Werte lt. Fehlerbeschreibung s.u.

NC.OsciUtil, call - Kommando

row, col	Typ	Beschreibung	Bsp.: "GetSampleTime"
0,0	string	Aufzeichnungsdauer lesen	GetSampleTime

NC.OsciUtil, call - Antwortdaten

row, col	Typ	Beschreibung	Bsp.: "GetSampleTime"
0,0	long	Quittungsmeldung	0 = kein Fehler
1,0	double	Aufzeichnungsdauer	4.0

NC.OsciUtil, call - Kommando

row, col	Typ	Beschreibung	Bsp.: "SetSampleTime"
0,0	string	Aufzeichnungsdauer setzen	SetSampleTime
1,0	double	Aufzeichnungsdauer	16.0

NC.OsciUtil, call - Antwortdaten

row, col	Typ	Beschreibung	Bsp.: "SetSampleTime" - mögliche Antwort
0,0	long	Quittungsmeldung	0 = kein Fehler

NC.OsciUtil, call - Kommando

row, col	Typ	Beschreibung	Bsp.: "GetSignalIds"
0,0	string	Kennung der ausgewählten Signale lesen	GetSignalIds

NC.OsciUtil, call - Antwortdaten

row, col	Typ	Beschreibung	Bsp.: "GetSignalIds"
0,0	long	Quittungsmeldung	0 = kein Fehler
0,1	long	Kennung Signal 1	0
0,n	long	Kennung Signal 2	2

NC.OsciUtil, call - Kommando

row, col	Typ	Beschreibung	Bsp.: "SetSignalIds"
0,0	string	Aufzeichnungssignale auswählen	SetSignalIds
0,1	long	Kennung Signal 1	0
0,n	long	Kennung Signal n	3

NC.OsciUtil, call - Antwortdaten

row, col	Typ	Beschreibung	Bsp.: "SetSignalIds" - mögliche Antwort
0,0	long	Mögliche Quittierungsmeldungen	0 = kein Fehler 1 = Messpunktliste ist leer 2 = Fehler Fileservers Negative Werte lt. Fehlerbeschreibung s.u.

NC.OsciUtil, call - Kommando

row, col	Typ	Beschreibung	Bsp.: "GetSampleData"
0,0	string	Aufzeichnungsdaten lesen	GetSampleData

NC.OsciUtil, call - Antwortdaten

row, col	Typ	Beschreibung	Bsp.: "GetSampleData"
0,0	long	Mögliche Quittierungsmeldungen	0 = kein Fehler Negative Werte lt. Fehlerbeschreibung s.u.
1,0..n	long	Daten des Aufzeichnungskanals 1	
n,0..n	long	Daten weiterer Aufzeichnungskanäle	

NC.OsciUtil, call - Kommando

row, col	Typ	Beschreibung	Bsp.: "GetSampleBuffer"
0,0	string	Größe des Aufzeichnungspuffers lesen	GetSampleBuffer

NC.OsciUtil, call - Antwortdaten

row, col	Typ	Beschreibung	Bsp.: "GetSampleBuffer"
0,0	long	Quittungsmeldung	0 = kein Fehler
1,0	long	Größe des Aufzeichnungspuffers	12000

NC.OsciUtil, call - Kommando

row, col	Typ	Beschreibung	Bsp.: "SetSampleBuffer"
0,0	string	Größe des Aufzeichnungspuffers setzen	SetSampleBuffer
1,0	long	Größe des Aufzeichnungspuffers	20000

NC.OsciUtil, call - Antwortdaten

row, col	Typ	Beschreibung	Bsp.: "SetSampleBuffer" - mögliche Antwort
0,0	long	Mögliche Quittierungsmeldungen	0 = kein Fehler 2 = Fehler Fileserver Negative Werte lt. Fehlerbeschreibung s.u.

NC.OsciUtil, call - Kommando

row, col	Typ	Beschreibung	Bsp.: "GetPreTriggerPosition"
0,0	string	Triggerposition lesen	GetPreTriggerPosition

NC.OsciUtil, call - Antwortdaten

row, col	Typ	Beschreibung	Bsp.: "GetPreTriggerPosition"
0,0	long	Quittungsmeldung	0 = kein Fehler
1,0	long	Triggerposition (0...100%)	10

NC.OsciUtil, call - Kommando

row, col	Typ	Beschreibung	Bsp.: "SetPreTriggerPosition"
0,0	string	Triggerposition setzen	SetPreTriggerPosition
1,0	long	Triggerposition (0...100%)	30

NC.OsciUtil, call - Antwortdaten

row, col	Typ	Beschreibung	Bsp.: "SetPreTriggerPosition"
0,0	long	Quittungsmeldung	0 = kein Fehler

NC.OsciUtil, call - Kommando

row, col	Typ	Beschreibung	Bsp.: "GetTriggerKind"
0,0	string	Triggerart lesen	GetTriggerKind

NC.OsciUtil, call - Antwortdaten

row, col	Typ	Beschreibung	Bsp.: "GetTriggerKind"
0,0	long	Quittungsmeldung	0 = kein Fehler
1,0	long	Mögliche Triggerarten	0 = unbekannt 2 = kein Trigger 4 = manueller Trigger 64 = Programmsatz-Trigger

NC.OsciUtil, call - Kommando

row, col	Typ	Beschreibung	Bsp.: "SetTriggerKind"
0,0	string	Triggerart setzen	SetTriggerKind
1,0	long	Manuellen Trigger setzen	4

NC.OsciUtil, call - Antwortdaten

row, col	Typ	Beschreibung	Bsp.: "SetTriggerKind"
0,0	long	Quittungsmeldung	0 = kein Fehler

NC.OsciUtil, call - Kommando

row, col	Typ	Beschreibung	Bsp.: "GetMinSampleTime"
0,0	string	Minimale Aufzeichnungszeit lesen	GetMinSampleTime

NC.OsciUtil, call - Antwortdaten

row, col	Typ	Beschreibung	Bsp.: "GetMinSampleTime"
0,0	long	Quittungsmeldung	0 = kein Fehler
1,0	double	Minimale Aufzeichnungszeit	4000

NC.OsciUtil, call - Kommando

row, col	Typ	Beschreibung	Bsp.: "GetProgramTrigger"
0,0	string	Programmtrigger lesen	GetProgramTrigger

NC.OsciUtil, call - Antwortdaten

row, col	Typ	Beschreibung	Bsp.: "GetProgramTrigger"
0,0	long	Mögliche Quittierungsmeldungen	0 = kein Fehler -1 = Programmtrigger nicht aktiv
1,0	string	Programmname	Test.npg
1,1	string	Programmzeile	470
1,2	long	Kanalnummer	1

NC.OsciUtil, call - Kommando

row, col	Typ	Beschreibung	Bsp.: "SetProgramTrigger"
0,0	string	Programmtrigger setzen	SetProgramTrigger
1,0	string	Programmname	Test.npg
1,1	string	Programmzeile	470
1,2	long	Kanalnummer	1

NC.OsciUtil, call - Antwortdaten

row, col	Typ	Beschreibung	Bsp.: "SetProgramTrigger"
0,0	long	Mögliche Quittierungsmeldungen	0 = kein Fehler -1 = Dateiname > 32 Zeichen -2 = Fehler beim Öffnen der Datei -3 = Triggerbedingung nicht gefunden -4 = Fehlerhafte Kanalnummer

NC.OsciUtil, call - Kommando

row, col	Typ	Beschreibung	Bsp.: "GetSignalInfo"
0,0	string	Signalinformation lesen	GetSignalInfo

NC.OsciUtil, call - Antwortdaten

row, col	Typ	Beschreibung	Bsp.: "GetSignalInfo"
0,0	long	Quittungsmeldung	0 = kein Fehler
1,0	long	Nummer der durchgeführten Messung	1

row, col	Typ	Beschreibung	Bsp.: "GetSignalInfo"
1,1	long	Länge der Signaldaten	500
1,2	long	Triggerposition	20

NC.OsciUtil, call - Kommando

row, col	Typ	Beschreibung	Bsp.: "NumberOfMpo- ints"
0,0	string	Anzahl möglicher Signalpunkte lesen	NumberOfMpoints

NC.OsciUtil, call - Antwortdaten

row, col	Typ	Beschreibung	Bsp.: "NumberOfMpo- ints"
0,0	long	Quittungsmeldung	0 = kein Fehler
1,0	long	Anzahl möglicher Messpunkte	400

NC.OsciUtil, call - Kommando

row, col	Typ	Beschreibung	Bsp.: "GetAvailableSig- nallds"
0,0	string	Liste der möglichen Signalpunkte lesen	GetAvailableSignallds

NC.OsciUtil, call - Antwortdaten

row, col	Typ	Beschreibung	Bsp.: "GetAvailableSig- nallds"
0,0	long	Quittungsmeldung	0 = kein Fehler -1 = keine Liste vorhan- den
1..n,0	string	Signalname	Sollposition
2..n,1	string	Einheit	mm
3..n,2	long	Achsnummer	3
4..n,3	long	Identifikationsnummer	1
5..n,4	double	Multiplikationsfaktor	1.5

NC.PlcData

Beschreibung	Das Item erlaubt den lesenden und schreibenden Zugriff auf PLC-Daten.
Anmerkung	Mögliche PLC-Daten sind Eingänge, Ausgänge und Merker, die durch Typvorgabe feldweise selektiert werden.



Einschränkung: Das Schreiben von Ein- und Ausgängen ist nur mit dem Target MTX-Simulation möglich!

Mit Hilfe der Übergabeparameter kann auf bis zu 20 Typ-Felder parallel lesend oder schreibend zugegriffen werden. Wird auf mehrere Felder gleichzeitig zugegriffen, müssen die Feldanzahl erhöht und für jedes Feld die vier Parameter "Typ, Länge, Index und Offset" zusätzlich beim Anlegen des Items vorgegeben werden.

Zur Reduzierung der Systembelastung sollte bei zyklischen Aufrufen immer mit **Verzögerung** erfolgen.

Items

Item Syntax *NC.PlcData,(1),(2),(3),(4),(5),(6) [(7),(8),(9),(10)] ... [(79),(80),(81),(82)]*

1. Feldanzahl

- max. 20 Felder

2. Verzögerung

- 0 : keine
- 1 : Verzögerung 300 ms

3. Datentyp (Feld 1)

- 65: Ausgänge
- 69: Eingänge
- 77: Merker

4. Datenlänge (Feld 1)

- Feldlänge in Byte

5. Index (Feld 1)

- z.Zt. immer "0" (reserviert für Typenerweiterung)

6. Offset (Feld 1)

- PLC-Adressoffset innerhalb des gewählten Datentyps

optional weitere Felder mit jeweils Datentyp, Datenlänge, Index und Offset

Aufruf synchron / asynchron

Zugriff Read / Write

zyklisch (opt.) ja

Datenformat **NC.PlcData,1,0,77,8,0,6110 read**

row, col	Typ	Beschreibung	Bsp.: Lesen von Merkerdaten. (8 Bytes ab Merker-Adresse 6110)
0,0	string	Statusmeldung der gelesenen Felddaten Mögliche Meldungen sind: 0 = kein Fehler, Daten sind gültig -10 = Datengröße überschritten -11 = Datentyp unbekannt -14 = Offsetwert ungültig	0
0,1	long	Wert des 1. Datenbyte	76
0,2	long	Wert des 2. Datenbyte	0
0,3	long	Wert des 3. Datenbyte	4
0,4	long	Wert des 4. Datenbyte	0
0,5	long	Wert des 5. Datenbyte	0
0,6	long	Wert des 6. Datenbyte	255
0,7	long	Wert des 7. Datenbyte	3
0,8	long	Wert des 8. Datenbyte	12

NC.PlcData,1,0,77,8,0,6110 write

row, col	Typ	Beschreibung	Bsp.: Schreiben von 8 Bytes im Merkerbereich. Die Merker-Adresse 6115 mit "132", alle anderen mit "0" überschreiben
0,0	long	Wert des 1. Datenbyte	0
0,1	long	Wert des 2. Datenbyte	0
0,2	long	Wert des 3. Datenbyte	0
0,3	long	Wert des 4. Datenbyte	0
0,4	long	Wert des 5. Datenbyte	0
0,5	long	Wert des 6. Datenbyte	132
0,6	long	Wert des 7. Datenbyte	0
0,7	long	Wert des 8. Datenbyte	0

NC.ProgramChanInfo

Beschreibung

Liefert die Information, in welchen Kanälen das erfragte Programm aktiv ist.

Ein Programm ist aktiv, wenn es entweder angewählt oder in der Aufrufstruktur des Hauptprogramms als Unterprogramm vorhanden ist.

Falls in der NC **bis zu 32 Kanäle** konfiguriert sind, wird eine einzelne Bitmaske verwendet:

- Bit 0 – das Programm ist in Kanal 1 aktiv
- Bit 1 – das Programm ist in Kanal 2 aktiv
- ...
- Bit 31 – das Programm ist in Kanal 32 aktiv

Falls **mehr als 32 Kanäle** konfiguriert sind, wird zusätzlich eine zweite Bitmaske verwendet:

- Bit 0 – das Programm ist in Kanal 33 aktiv
- Bit 1 – das Programm ist in Kanal 34 aktiv
- ...
- Bit 31 – das Programm ist in Kanal 64 aktiv

Bei **mehr als 64 Kanälen** werden entsprechend weitere Bitmasken verwendet.

Anmerkung

Beim Instanzieren des Items wird der Programmname mit Pfad vorgegeben.

Item Syntax

NC.ProgramChanInfo,(1)

1. Programmname mit Pfad

Aufruf

synchron / asynchron

Zugriff

Read

zyklisch (opt.)

nein

Datenformat

Die Zeilenanzahl ist abhängig von der Anzahl Bitmasken

NC.ProgramChanInfo, read

row, col	Typ	Beschreibung	Bsp. NC.ProgramChanInfo,/mnt/ beispiel.npg
0,0	ulong	Bitmaske für die Kanäle 1-32	3 bitcodierter Wert, d. h. das Programm ist in Kanal 1 + 2 aktiv
1,0	ulong	Bitmaske für die Kanäle 33-64	1 bitcodierter Wert, d. h. das Programm ist in Kanal 33 aktiv

NC.ReloadExportUs**Beschreibung**

Über das Item wird der NFS-Server aufgefordert, die in der Datei "export.us" stehende Verzeichnisstruktur für Zugriffe der NC freizugeben.

Über das Item kann auf die Datei sowohl lesend als auch schreibend zugegriffen werden. Der Lesezugriff liefert dem Anwender den kompletten Inhalt der Datei "export.us". Dieser Inhalt kann vom Anwender überarbeitet (Eintragen bzw. Löschen von Verzeichnisfreigaben) und über den Schreibzugriff des Items zurück in die Datei geschoben werden.



Beim Ändern der Datei "export.us" ist darauf zu achten, dass **kein** CR (0x0d) im zu schreibenden String übergeben wird. Ansonsten kommt es zu Fehlinterpretationen des NFS-Servers

Beim Zurückschreiben der neuen Daten werden diese nicht in die alte Datei geschrieben, sondern in eine neue Datei "export.us". Die alte Datei wird vorher in "export.us_bak" umbenannt und steht dann für eventuelle Kontrollzwecke weiter zur Verfügung. Nach dem Beenden des Schreibvorgangs wird die neue Struktur an den NFS-Server gemeldet.

Anmerkung

Der Schreibvorgang ist auch ohne Übergabe von neuen Daten möglich. In so einem Fall wird die existierende Datei "export.us" nicht angepackt. Es wird lediglich der NFS-Server veranlasst, den Inhalt der Datei erneut zu lesen und die Verzeichnisse freizugeben.

Die Zugriffspfade zum Lesen und Schreiben der "export.us"-Datei werden intern über den Path-Service ermittelt.

Beim Aufruf des Items können die folgenden Fehler auftreten:

2 = SCPErrFailed - "Registry-Pfad auf export.us kann nicht gelesen werden"

110 = SCPErrNACK10 - "Dateizugriff oder Umbenennung nicht möglich"

Item Syntax

NC.ReloadExportUs

Aufruf

asynchron

Zugriff

Read / Write

zyklisch (opt.)

nein

Datenformat

NC.ReloadExportUs, read

row, col	Typ	Beschreibung	Bsp.
0,0	string	Inhalt der Datei "export.us"	C:\mnt mtxctrl localhost C:\mnt1 mtxctrl

NC.ReloadExportUs, write

row, col	Typ	Beschreibung	Bsp.
0,0	string	Eintrag neuer Verzeichnisfreigaben	C:\mnt mtxctrl localhost C:\mnt1 mtxctrl C:\mnt2 mtxctrl

NC.WaitingStates

Beschreibung Gibt den Systemwartezustand wieder. Es handelt sich um ein Summensignal über alle Kanäle.

Anmerkung Für Wartezustände steht der Enumerator "LDX_McoWaitingState" zur Verfügung. Er kann folgende Werte annehmen:

Enumerator LDX_McoWaitingState

LDX_McoWaitingState	Beschreibung
LDX_McoDwellTime_C	Verweilzeit
LDX_McoAuxiliaryFunction_C	Quittungspflichtige Hilfsfunktion
LDX_McoBlockTransferInhibit_C	Einlesesperre
LDX_McoChannelFeedrate0_C	Vorschub im Kanal gleich 0
LDX_McoProgramStop_C	Programm Halt mit M0/M1
LDX_McoFeedHold_C	Vorschub Halt im Kanal
LDX_McoFeedInhibit_C	Vorschubsperr im Kanal oder in einer Kanalachse
LDX_McoCustomerStop_C	Vom Kunden vorgegebene Einlesesperre
LDX_McoSynchronizedMovementStop_C	Synchronisierter Bewegungsstopp zwischen Kanälen (ASTOP, ...)
LDX_McoWaitingForAxisTransfer_C	Warten auf Achse bei Achsentauch (G511)
LDX_McoWaitingForPermanentVariable_C	Warten auf permanente Variable (WPV)
LDX_McoWaitingForIFSignalActive_C	Warten auf Interface-Signal zum aktiven Zeitpunkt (WAITA, ...)
LDX_McoWaitingForIFSignalPreparation_C	Warten auf Interface-Signal (WAIT(IC...)) oder auf eine festgelegte Zeitspanne (WAIT(,ZEIT%)) in der Satzvorbereitung
LDX_McoMotionControlDataService_C	Motion Control Datendienst (MCODS(...))
LDX_MCoWaitingForInpos_C	Warten auf Inpos der Kanalachsen
LDX_MCoFeedProgramming_C	Warten auf G95-Programmierung
LDX_MCoWaitingForSpindleG33_C	Warten bis Hauptspindle dreht G33
LDX_MCoNcProgramRestart_C	Warten auf Satzvorlauf
LDX_MCoWaitingForSpindle_C	Warten auf Drehzahl erreicht, Spindel steht, ...
LDX_MCoWaitingForEvent_C	Warten auf Event >>> Funktion WEV
LDX_MCoWaitingForInternalResource_C	Warten auf interne Ressource (z. B. Speicher)

Item Syntax **NC.WaitingStates**

	keine Parameter
Aufruf	synchron / asynchron
Zugriff	Read
zyklisch (opt.)	ja
Datenformat	NC.WaitingStates, read

row, col	Typ	Beschreibung	Bsp.
0,0	long	aktiver Systemwartezustand: "Vorschub Halt"	32 - LDX_McoFeedHold_C

4.7.2 Antriebe

NC.OscilloscopeConfiguration

Beschreibung	Das Item ermittelt, ob die im System befindlichen Steuerungsachsen die Oszilloskop-Funktion unterstützen. Beim Schreiben kann für das Aufzeichnen eine Konfiguration der Parameter vorgegeben werden.
Item Syntax	NC.OscilloscopeConfiguration keine Parameter
Aufruf	synchron / asynchron
Zugriff	Read 0 - Antrieb unterstützt keine Oszilloskop-Funktion 1 - Antrieb unterstützt die Oszilloskop-Funktion Write <i>Beim Schreiben der Konfiguration muss der Steuerung mitgeteilt werden,</i> - welcher der vorhandenen Antriebe an der Messung teilnimmt - ob ein vorhandener Antrieb oder die NC den Trigger auslöst. Dazu muss hier die Nummer des Antriebs bzw. "-1" für die NC eingetragen werden
zyklisch (opt.)	nein
Datenformat	Die Daten pro Antrieb werden zeilenweise übertragen. Jede Zeile besteht aus 2 Spalten. Spalte 1 enthält die Nummer des Antriebs (1..n). Spalte 2 kann zwei Werte annehmen (0 = nimmt nicht an Messung teil; 1 = Teilnehmer der Messung). Vorgaben für Antriebe, die nicht an der Messung teilnehmen, sind nicht erforderlich. Die letzte Zeile enthält die Triggerinformation. Wird diese Zeile weggelassen, steht der Trigger automatisch auf NC!

NC.OscilloscopeConfiguration, Read

row, col	Typ	Beschreibung	Bsp.
0,0	long	Antrieb 1 unterstützt nicht das Oszilloskopieren	0
1,0	long	Antrieb 2 unterstützt das Oszilloskopieren	1
n,1	long	Antrieb n unterstützt nicht das Oszilloskopieren	0

NC.OscilloscopeConfiguration, Write

row, col	Typ	Beschreibung	Beispiel: Achsen 1-3 für Aufzeichnung konfigu- riert. Triggerbedingung löst Achse 1 aus
0,0	long	Nummer des Antriebs	1
0,1	long	Teilnehmer der Messung	1
1,0	long	Nummer des Antriebs	2
1,1	long	Teilnehmer der Messung	1
2,0	long	Nummer des Antriebs	3
2,1	long	Teilnehmer der Messung	1
n,n	long	...	...
n,0	long	Trigger-Information	1

NC.SercosPhase2

Beschreibung

Lesen und Schreiben der Sercos Phase.

Das Schreiben des Items hat eine Phasenumschaltung zur Folge.

Beim Schreiben werden folgende Daten übergeben:

- Zielphase, in welche umgeschaltet werden soll
Operation-Mode: 0..4
Parametrier-Mode: 0x14
- ZK1-Fehler rücksetzen
- Download der .scs-Files in Phase 2 durchführen
- Download der .scs-Files in Phase 3 durchführen
- Startup-Mode (optional)
0 = Hochlauf normal (default)
1 = Hochlauf mit Topologieadressierung
- Gerätenummer (optional)
0 = Umschaltung für gesamten Sercos Ring (default)
>0 = Umschaltung für jeweiliges Sercos Gerät
- Platzhalter für spätere Erweiterungen (optional)

Item Syntax

NC.SercosPhase2

keine Parameter

Aufruf

synchron / asynchron

Zugriff

Read / Write

zyklisch (opt.)

nein

Datenformat NC.SercosPhase2, Read

row, col	Typ	Beschreibung	Bsp.:
0,0	long	Aktive Phase [0..4]	4
0,1	long	Status der Phasenmaschine Folgende Statusmeldungen sind möglich: 0 - Kein Fehler beim Umschalten der Phase 32771 - Phasenumschaltung über serielles Protokoll nicht möglich 32773 - Antriebsadresse nicht korrekt - Phasenhochschaltung wird nicht durchgeführt, wenn die Adressen der Antriebe im Ring nicht mit den parametrisierten Antrieben übereinstimmen 53250 - Umschaltung nach Phase 3 nicht möglich 53251 - Umschaltung: Phase 3 => 4 nicht möglich 53253 - Phasenumschaltung noch aktiv - wird gemeldet, wenn die Phasenumschaltung noch nicht abgeschlossen ist und eine neue Phase vorgegeben wird	0
0,2	long	Zustand der Phasenumschaltung 0 - Phasenumschaltung inaktiv 1 - Phasenumschaltung aktiv	0
0,3	long	Startup-Mode (verfügbar ab Version 14V14 R3) 0 = Hochlauf normal 1 = Hochlauf mit Topologieadressierung	0
1,0	long	Gerätenummer 0 = Umschaltung für gesamten Sercos Ring >0 = Umschaltung für jeweiliges Sercos Gerät	0

NC.SercosPhase2, Write

row, col	Typ	Beschreibung	Umschalten in Sercos Phase 4, Rücksetzen der ZK1-Fehler, Download der scs-Dateien, Hochlauf "normal".
0,0	long	Zielphase	4
0,1	bool	Reset ZK1	TRUE
0,2	bool	Download in Phase 2	TRUE
0,3	bool	Download in Phase 3	TRUE
0,4	long	Startup mode (optional)	0
0,5	void	Platzhalter (optional)	
1,0	long	Gerätenummer (optional)	0

NC.SPA1

Beschreibung Liest und schreibt Elemente der Sercos Parameter

Anmerkung NC.SPA1 dient dem Zugriff auf Steuerungsparameter, wenn sie im Sercos Parameter-Format zugreifbar sind. Die Steuerungsparameter werden in der Regel über die "Antriebsadresse" 0 adressiert.

NC.SPA1 dient auch dem Zugriff auf Antriebsparameter, wenn die Antriebe über die Steuerungen erreichbar sind. Antriebe werden über die Antriebsadresse ausgewählt.

Fehler, die beim Zugriff auf Sercos Parameter auftreten, werden über die lwSCP-Fehlernummer "SCPErrNACK1" abgebildet.

Item Syntax

NC.SPA1,(1),(2),(3)

1. Antriebsadresse

2. Ident

Ident-Format

X-Y-ZZZZ	16Bit-Identnummer-Bereich	
X-Y-ZZZZ.SI[.SE]	32Bit-Identnummer-Bereich	
X	Parametertyp bestehend aus einem Buchstaben. Siehe auch Gerätebeschreibung.	
	Beispiele:	
	S	Standard Parameter
	P	Produktspezifischer Parameter
Y	Parametersatz. (0 ... 7)	
ZZZZ	Datenblock- Nummer (0000 ... 4095) Achtung, die Datenblocknummer wird mit 4 Stellen Dezimal angegeben.	
SI	Sercos Instanz Dezimal ohne führende 0. (0..255)	
SE	Optional: Sercos Element Dezimal ohne führende 0. (0..255)	

Tab. 4-31: Sercos Idents

3. Elementkodierung

Die Elementkodierung wird genutzt, um Datenelemente des Sercos-Parameters anzusprechen.

Elementkodierung

Element des Parameters	Standard-Format	Advanced Format
	nur ein Buchstabe kann angegeben werden	Werte können durch "Oder" verknüpft werden
Datenstatus	S	01H
Name	N	02H
Attribut	A	04H
Einheit	U	08H
min. Eingabewert	L	10H
max Eingabewert	H	20H
Betriebsdatum	D	40H

Tab. 4-32: Sercos Parameter-Elementkodierung

Beispiele

NC.SPA1,1,S-0-0003,L:

Minimaler Eingabewert des Parameters S-0-0003 von Antrieb 1

NC.SPA1,1,S-0-0003,D:

Betriebsdatum des Parameters S-0-0003 von Antrieb 1

NC.SPA1,1,S-0-0003,48:

Einheit & Betriebsdatum des Parameters S-0-0003 von Antrieb 1

Aufruf synchron / asynchron

Zugriff Read

Die Elemente der Sercos Parameter werden grundsätzlich als Strings geliefert.

Ergebnisse sind in folgender Form kodiert:

angefordertes Element	Ergebnis	Beispiel
Datenstatus	hexadezimaler Word	"0x0000"
Name	String	"NC cycle time (TNcyc)"
Attribut	hexadezimaler Doppel-Word	"0x60110001"
Einheit	String	"µs"
min. Eingabewert	dezimaler Word	"2000"
max Eingabewert	dezimaler Word	"20000"
Wert	dezimal hexadezimal (max. 8 Stellen) binär (max. 64 Zeichen)	"1.5" "0x80" "1111000010101010"

Write

Nur das Betriebsdatum kann geschrieben werden, falls das im Parameter möglich ist.

Datenformat Zeilenanzahl entspricht der Anzahl der angeforderten Parameter

NC.SPA1, read

row, col	Typ	Beschreibung	NC.SPA1,3,S-0-0111,08
0,0	string	angefordertes Element	"A eff"

Listen im einfachen Format Mit Elementkodierung "D" angefordert

NC.SPA1, read

row, col	Typ	Beschreibung	NC.SPA1,3,S-0-0017,D
0,0	string	Liste	S-0-0001\n S-0-0002\n ... S-0-xxxx

Listen mit Längenangabe Mit Elementkodierung "40" angefordert

NC.SPA1, read

row, col	Typ	Beschreibung	NC.SPA1,3,S-0-0017,40
0,0	string	Ist-Länge der Liste im Antrieb	"400"
0,1	string	max. Länge der Liste im Antrieb	"400"
0,2	string	Liste	"S-0-0001\n S-0-0002\n ... S-0-xxxx"

NC.SPA1, write

row, col	Typ	Beschreibung	NC.SPA1,3,S-0-0111,40
0,0	string	zuzuweisender Wert	"1.500"

NC.SPH1

Beschreibung Das Item SPH1 wird zum Lesen und Schreiben der Sercos Phase verwendet. Das Schreiben des Items hat eine Phasenumschaltung zur Folge.

Beim Schreiben werden folgende Daten übergeben:

- Zielphase, in welche umgeschaltet werden soll
Wertebereich [0..4]
- Kennung für die Art des Hochlaufs (optional)
0 = Hochlauf normal
1 = Hochlauf mit Topologieadressierung
- Platzhalter für spätere Erweiterungen (optional)

Item Syntax **NC.SPH1**

keine Parameter

Aufruf synchron / asynchron

Zugriff Read / Write

zyklisch (opt.) nur lesend

Datenformat NC.SPH1 read

row, col	Typ	Beschreibung	Bsp.:
0,0	long	Aktive Phase [0..4]	4
0,1	long	Status der Phasenmaschine Folgende Statusmeldungen sind möglich: 0 - Kein Fehler beim Umschalten der Phase 32771 - Phasenumschaltung über serielles Protokoll nicht möglich 32773 - Antriebsadresse nicht korrekt - Phasenhochschaltung wird nicht durchgeführt, wenn die Adressen der Antriebe im Ring nicht mit den parametrisierten Antrieben übereinstimmen 53250 - Umschaltung nach Phase 3 nicht möglich 53251 - Umschaltung: Phase 3 => 4 nicht möglich 53253 - Phasenumschaltung noch aktiv - wird gemeldet, wenn die Phasenumschaltung noch nicht abgeschlossen ist und eine neue Phase vorgegeben wird	0

NC.SPH1 write

row, col	Typ	Beschreibung	Bsp.: Umschalten auf Phase 4, Hochlauf "normal"
0,0	long	Zielfase [0..4]	4
0,4	long	Startup mode (optional)	0
0,5	void	Platzhalter (optional)	

4.7.3 CPL-Daten

NC.CplPermVariable

Beschreibung

Ermöglicht Lese- und Schreib-Zugriff auf permanente CPL-Variable. Bei Feldvariablen können die Indizes verwendet werden, um auf einzelne Elemente zuzugreifen

Indizes sind in CPL 1-basiert. Es gilt also:

- $1 \leq \text{Index1} \leq \text{Dimension1}$
- $1 \leq \text{Index2} \leq \text{Dimension2}$



Das Lesen erfolgt mit Hilfe eines Caches, der ca. alle 100 ms aktualisiert wird.

Item Syntax

NC.CplPermVariable,(1),(2),(3)

1. Variablenname
2. Index1 (optional)
3. Index2 (optional)

Aufruf synchron / asynchron
Zugriff Read / Write
zyklisch (opt.) nur lesend (Arraygröße beachten)
Datenformat Zeilenanzahl abhängig von MACODA

NC.CplPermVariable, read

row, col	Typ	Beschreibung	NC.CplPermVariable,@ToolNo
0,0	long	Variablenname	4711

NC.CplPermVariable, write

row, col	Typ	Beschreibung	NC.CplPermVariable,@Darray, 2,5
0,0	double	Variablenname	2.75

NC.CplPermVariableList

Beschreibung *Stellt alle permanenten CPL-Variablen in einer Liste bereit:*

- feste CPL-Variable (@1-@100)
- alle definierbaren CPL-Variable

Item Syntax NC.CPLPermVariableList

Aufruf asynchron

Zugriff Read

zyklisch (opt.) nein

Datenformat Zeilenanzahl: Anzahl der CPL-Variablen, definiert in "wmhperm.dat" und "anwperm.dat"

NC.CplPermVariableList, read

row, col	Typ	Beschreibung	Beispiel
0,0	string	Name der 1. festen Variablen	@1
0,1	long	Typ der 1. festen Variablen	1
0,2	long	Dimension1 der 1. festen Variablen	1
0,3	long	Dimension2 der 1. festen Variablen	1
...			
99,0	string	Name der 100. festen Variablen	@100
99,1	long	Typ der 100. festen Variablen	1
99,2	long	Dimension1 der 100. festen Variablen	1
99,3	long	Dimension2 der 100. festen Variablen	1
100,0	string	Name der 1. definierbaren Variablen	@_X
100,1	long	Typ der 1. definierbaren Variablen	1

row, col	Typ	Beschreibung	Beispiel
100,2	long	Dimension1 der 1. definierbaren Variablen	10
100,3	long	Dimension2 der 1. definierbaren Variablen	2
...			

NC.CplPermVariableWatch

Beschreibung Überwacht Zugriffe auf permanente CPL-Variablen und liefert deren aktuellen Werte.

Anmerkung Beim Instanzieren des Items werden der Variablenname, ein Watch-Mode und optional Indizes angegeben.

Die Überwachung von dimensionierten CPL-Variablen reagiert sobald auf ein Element dieser CPL-Variable zugegriffen wird.

Zwei optionale Parameter bestimmen, welche Daten das Item liefert. Durch Angabe von Indizes können auch aus einem Feld Elemente spezifiziert werden (max. 100), die in der Antwort geliefert werden. Falls kein Index angegeben wird, liefert das Item alle Werte der jeweiligen CPL-Variable. Bei zweidimensionalen CPL-Variablen und fehlendem Index2, liefert das Item die komplette Zeile des unter Index1 angegebenen Wertes.

Indizes sind in CPL 1-basiert. Es gilt also:

- $1 \leq \text{Index1} \leq \text{Dimension1}$
- $1 \leq \text{Index2} \leq \text{Dimension2}$

Neben den aktuellen Daten der permanenten CPL-Variable liefert das Item eine Versionskennung. Der Vergleich mit der zuvor gelesenen Versionskennung zeigt, ob Zugriffe auf die angegebene CPL-Variable erfolgten.

Item Syntax **NC.CplPermVariableWatch,(1),(2),(3),(4)**

1. Variablenname

2. Watch-Mode

- 0: Überwachung von Lesezugriffen
- 1: Überwachung von Schreibzugriffen

3. Index1 (optional)

4. Index2 (optional)

Aufruf synchron / asynchron

Zugriff Read

zyklisch (opt.) ja

Datenformat **NC.CplPermVariableWatch, read**

row, col	Typ	Beschreibung	Bsp.: NC.CplPermVariable-Watch,@ToolNmb,1
0,0	long	Liefert bei Schreibzugriffen auf "@ToolNmb" den aktuellen Wert	1
1,0	long	Liefert bei Schreibzugriffen auf "@ToolNmb" die aktuelle Versionskennung	245633

Beispiel - Überwachung eines 1-dimensionalen Arrays

row, col	Typ	Beschreibung	NC.CplPermVariable- Watch,@_R,0,4,1
0,n	double	Liefert bei Lesezugriffen auf Elemente von "@_R" den aktuellen Wert des vierten Elementes	2.0
1,0	long	Liefert bei Lesezugriffen auf "@_R" die aktuelle Versionskennung	12890

Beispiel - Überwachung eines 2-dimensionalen Arrays

row, col	Typ	Beschreibung	NC.CplPermVariable- Watch,@SampleArray,1,2,9
0,0	double	Liefert bei Schreibzugriffen auf Elemente der zweidimensionalen CPL-Variable "@SampleArray" den aktuellen Wert von Element 9 in Zeile 2.	21.4
1,0	long	Liefert bei Schreibzugriffen auf Elemente der zweidimensionalen CPL-Variable "@SampleArray" die aktuelle Versionskennung.	7421

4.7.4 Dateibasierte Tabellen**NC.XmlRootElementInfo**

Beschreibung	Liefert zum vorgegebenen Root-Element (Tabellentyp) das Top-Level-Schema und die Extension.
Anmerkung	Die Extension wird nur für dateibasierte Tabellen geliefert.
Item Syntax	NC.XmlRootElementInfo,(1) 1. Root-Element
Aufruf	synchron / asynchron
Zugriff	Read
zyklisch (opt.)	nein
Datenformat	Zeilenanzahl: zwei für dateibasierte Tabellen, sonst eine Zeile

NC.XmlRootElementInfo, read

row, col	Typ	Beschreibung	Bsp. NC.XmlRootElementInfo,/ZOT
0,0	string	Information zu NPV-Tabellen Top-Level-Schema	/feprom/zot_sd.xsd
0,1	string	Extension	zot

NC.XmlRootElementList

Beschreibung	Liefert zu einer XML-Datenklasse (z. B. dateibasierte Tabelle, Konfigurationsdaten) eine Liste aller Root- Elemente. Die Anzahl der gelieferten Root-Elemente ist je nach XML-Datentyp unterschiedlich. Außerdem ist sie abhängig von anderen Randbedingungen (z. B. Nutzung von User-definierbaren Tabellen).
---------------------	---

Anmerkung Die XML-Datenklasse wird als Parameter beim Instanzieren des Items vorgegeben.

Zur Vorgabe der XML-Datenklasse steht der Enumerator "LDX_XmlData-Class" zur Verfügung. Die XML-Datenklasse eines Root-Elements wird über das Attribut "Class" im jeweiligen Schema festgelegt. Es besteht folgende Zuordnung zwischen dem Wert des "Class"-Attributs und der XML-Datenklasse:

Enumerator LDX_XmlDataClass

LDX_XmlDataClass	Beschreibung
LDX_XML_DATA_CLASS_UNKNOWN_C	sonstiges
LDX_XML_DATA_CLASS_CFG_C	XML-Datenklasse "CFG"
LDX_XML_DATA_CLASS_DBTAB_C	XML-Datenklasse "DBTAB"
LDX_XML_DATA_CLASS_FILETAB_C	XML-Datenklasse "FILETAB"

Item Syntax NC.XmlRootElementList,(1)

1. Datenkategorie

Aufruf asynchron

Zugriff Read

zyklisch (opt.) nein

Datenformat Zeilenanzahl: Anzahl der gelieferten Root-Elemente

NC.XmlRootElementList, read

row, col	Typ	Beschreibung	NC.XmlRootElementList,3 Root-Elemente für die XML-Datenklasse "FILETAB"
n,0	string	Root-Element	/DCT

NC.XmlSchemaList

Beschreibung Liefert die Liste aller im Kern verwendeten Schemas.

Jeder Listeneintrag enthält den vollständigen Pfad, den Namen und die Extension des Schemas sowie das Datum (Zeitstempel) der letzten Änderung (UTC).

Item Syntax NC.XmlSchemaList

Aufruf asynchron

Zugriff Read

zyklisch (opt.) nein

Datenformat Zeilenanzahl: Anzahl aller im Kern verwendeten Schemas (variiert je nach Software-Version)

NC.XmlSchemaList, read

row, col	Typ	Beschreibung	Bsp.
n,0	string	Pfad/Name.Extension des Schemas	/feprom/schemas/basic_ty.xsd
n,1	date	Datum der letzten Änderung	04.04.04 16:26:12

NC.XmlTable_U

Beschreibung Lese- und Schreib-Zugriff auf dateibasierte Tabellen.

Dateibasierte Tabellen sind z. B. Nullpunktverschiebungs-Tabellen, D-Korrekturtabellen und Placement-Tabellen. Beim Anlegen des Items wird festgelegt, auf welche Tabelle der Zugriff erfolgt. Außerdem wird der Knoten in der Tabelle identifiziert.

Anmerkung Bei NPV-Tabellen besteht die Möglichkeit, Spalten einzufügen oder Spalten zu löschen.

Um Spalten einzufügen, wird im 2. Parameter des Items die Spalte spezifiziert, an der eingefügt wird. Das Einfügen von Spalten unterscheidet sich von der normalen Schreibfunktion dadurch, dass zu der 1. Zeile mit den Daten für die neue Spalte eine 2. Zeile mit der Konstanten "LDX_TAD_INSERT_AXIS_ZOT_C" hinzukommt.

Um eine Spalte zu löschen, wird im 2. Parameter des Items die zu löschende Spalte angegeben. Ein anschließendes Write ohne Daten in der 1. Zeile und mit der Konstanten "LDX_TAD_DELETE_AXIS_ZOT_C" in der 2. Zeile löscht die Spalte.

Der Enumerator "LDX_TadWriteMode" kann folgende Werte annehmen:

Enumerator LDX_TadWriteMode

LDX_TadWriteMode	Beschreibung
LDX_TAD_WRITE_CACHE	Zugriff erfolgt über Cache
LDX_TAD_INSERT_AXIS_ZOT_C	Achse/Spalte einfügen
LDX_TAD_DELETE_AXIS_ZOT_C	Achse/Spalte löschen

Fehler, die beim Lesen oder Schreiben von Tabellen auftreten können, sind in Abschnitt [TAD-Fehler auf Seite 227](#) beschrieben.

Item Syntax NC.XmlTable_U,(1),(2)

1. Tabellenname (kompletter Pfad, Dateiname)
2. Knoten in der Tabelle (optional)

Aufruf asynchron

Zugriff Read

- Die gelesenen Daten werden als XML-Strings geliefert.

Write

- Hier ist die Länge des XML-Strings auf 8 kByte beschränkt.

zyklisch (opt.) nein

Datenformat NC.XmlTable_U, read

row, col	Typ	Beschreibung	NC.XmlTable_U,/database/DC1.dct,/DCT Lesen einer D-Korrektur-Tabelle
0,0	string	Tabellendaten	<DCT> ... </DCT>

NC.XmlTable_U, write

row, col	Typ	Beschreibung	NC.XmlTable_U,/database/ DC1.dct,/DCT/D/SD/L1 Schreiben eines Wertes einer D- Korrektur-Tabelle
0,0	string	neuer Wert	<L1>2.5</L1>

NC.XmlTable_U, write

row, col	Typ	Beschreibung	NC.XmlTable_U,/database/ ZO1.zot,/ZOT/Offset/Ax[4] Einfügen einer Spalte in eine NPV-Tabelle
0,0	string	Daten für die neue Spalte	<Ax><Name>B</ Name><Type>Lin</ Type><Set><G54>5.1</ G54>...</Ax>
1,0	string	Konstante	LDX_TAD_IN- SERT_AXIS_ZOT_C

NC.XmlTable_U, write

row, col	Typ	Beschreibung	NC.XmlTable_U,/database/ ZO1.zot,/ZOT/Offset/Ax[2] Löschen der 2. Spalte in einer NPV-Tabelle
0,0	string	Daten werden ignoriert	leer
1,0	string	Konstante	LDX_TAD_DELE- TE_AXIS_ZOT_C

NC.XmlTableCreate

Beschreibung Legt eine neue Tabelle an.

Anmerkung Fehler, die beim Anlegen von Tabellen auftreten können, sind in Abschnitt [TAD-Fehler auf Seite 227](#) beschrieben.

Item Syntax **NC.XmlTableEvent**

Aufruf synchron / asynchron

Zugriff Write

Es wird eine Versionsnummer und der geänderte Knoten in Xpath-Syntax geliefert.

zyklisch (opt.) nein

Datenformat **NC.XmlTableCreate, write**

row, col	Typ	Beschreibung	Anlegen der D-Korrektur-Tabelle sample.dct
0,0	string	Dateiname der neuen Tabelle	/sample.dct
0,1	string	XPath-Syntax der neuen Tabelle	/DCT
0,2	string	verwendete Vorlage	/mnt/template.dct
0,3	long	Kanalnummer	empty

NC.XmlTableCreate, write

row, col	Typ	Beschreibung	Anlegen der Achsoffset-Tabelle offsets1.aot
0,0	string	Dateiname der neuen Tabelle	/offsets1.aot
0,1	string	XPath-Syntax der neuen Tabelle	/AOT
0,2	string	verwendete Vorlage	
0,3	long	Kanalnummer	1

NC.XmlTableEvent

Beschreibung Meldet Schreibzugriffe auf die im Editor geöffnete Tabelle. Bei Anlegen des Items wird angegeben, für welche Tabelle Schreibzugriffe gemeldet werden sollen.

Anmerkung U.U. können der geänderte Knoten oder die neuen Daten nicht geliefert werden. In diesem Fall muss die Tabelle bzw. der Knoten neu gelesen werden.

Eine lückenlose Benachrichtigung bei Schreibvorgängen kann nicht gewährleistet werden. Daher ist auf Client-Seite eine Überprüfung der mitgelieferten Versionsnummer notwendig. Falls die Versionsnummern fortlaufend sind, ist die Benachrichtigung lückenlos. Andernfalls wurden nicht alle Schreibzugriffe erfasst und die Tabelle muss komplett gelesen werden.

Fehler, die im Zusammenhang mit diesem Item auftreten können, sind in Abschnitt [TAD-Fehler auf Seite 227](#) beschrieben.

Item Syntax **NC.XmlTableEvent,(1)**

1. Tabellenname (kompletter Pfad, Dateiname)

Aufruf synchron / asynchron

Zugriff Read

Es wird eine Versionsnummer und der geänderte Knoten in Xpath-Syntax geliefert.

zyklisch (opt.) ja, erforderlich!

Datenformat **NC.XmlTableEvent, read**

row, col	Typ	Beschreibung	NC.XmlTableEvent,/database/DC1.dct Ermitteln von Schreibzugriffen auf eine D-Korrektur-Tabelle
0,0	dword	Versionsnummer	213
1,0	string	geänderter Knoten	/DCT/Corr[3]/Rough/L1

4.7.5 Datenbankbasierte Tabellen**NC.DbTableDataRec_U**

Beschreibung Lesezugriff auf einen Datensatz. Beim Instanzieren des Items wird der gewünschte Datensatz adressiert (Element-Pfad, Sektor, Platz).

Anmerkung Durch den Element-Pfad wird die Tabelle eindeutig identifiziert, außerdem kann über diesen Parameter der komplette Datensatz oder eine Teilstruktur adressiert werden.

Beim Schreiben werden im Normalfall alle übergebenen Daten in den Datensatz geschrieben. Falls nur werkzeugspezifische oder platzspezifische Daten geschrieben werden sollen, wird ein entsprechender Modus gewählt. Hierfür kann der Enumerator "LDX_DbSetMode" folgende Werte annehmen:

Enumerator LDX_DbdSetMode

LDX_DbdSetMode	Beschreibung
LDX_DBD_SETMODE_TOOL_DATA_C	nur werkzeugspezifische Daten
LDX_DBD_SETMODE_PLACE_DATA_C	nur platzspezifische Daten

Fehler, die beim Suchen von Datensätzen auftreten können, sind in Abschnitt [DBD-Fehler auf Seite 228](#) beschrieben.

Item Syntax **NC.DbTableDataRec_U,(1),(2),(3)**

1. Element-Pfad (in Xpath-Syntax)
2. Key1 (z. B. Sektor)
3. Key2 (z. B. Platz)

Aufruf asynchron

Zugriff Read

Die gelesenen Daten werden als XML-Strings übergeben.

Write

Schreibt Daten in Datensatz

zyklisch (opt.) nein

Datenformat **NC.DbTableDataRec_U, read**

row, col	Typ	Beschreibung	NC.DbTableDataRec_U,/DBT1/Rec,1,1 Liest die Datenbanktabelle 1
0,0	string	Inhalt des Datensatzes	<Rec><Hd><K1>1</K1><K2>1</K2><SKQ></SKQ><IKQ1>0</IKQ1>...

NC.DbTableDataRecLocked_U

Beschreibung Lese- und Schreib-Zugriff auf einen Datensatz. Beim Instanzieren des Items wird der gewünschte Datensatz adressiert (Element-Pfad, Sektor, Platz).

Anmerkung Durch den Element-Pfad wird die Tabelle eindeutig identifiziert, außerdem kann über diesen Parameter der komplette Datensatz oder eine Teilstruktur adressiert werden.

- Der adressierte Datensatz wird durch erstmaliges Lesen des Items gesperrt.
- Mit dem ersten Schreib-Zugriff wird der Datensatz freigegeben.
- Bei Abbau des Items ohne voriges Schreiben wird der Datensatz automatisch freigegeben.
- Nach einem Schreibzugriff kann der Datensatz durch einen Lesezugriff erneut gesperrt werden.

Beim Schreiben werden im Normalfall alle übergebenen Daten in den Datensatz geschrieben. Falls nur werkzeugspezifische oder platzspezifische Daten geschrieben werden sollen, wird ein entsprechender Modus gewählt. Hierfür kann der Enumerator "LDX_DbdSetMode" folgende Werte annehmen:

Enumerator LDX_DbdSetMode

LDX_DbdSetMode	Beschreibung
LDX_DBD_SETMODE_TOOL_DATA_C	nur werkzeugspezifische Daten
LDX_DBD_SETMODE_PLACE_DATA_C	nur platzspezifische Daten

Fehler, die beim Suchen von Datensätzen auftreten können, sind in Abschnitt [DBD-Fehler auf Seite 228](#) beschrieben.

Item Syntax	NC.DbTableDataRecLocked_U,(1),(2),(3)
	1. Element-Pfad (in Xpath-Syntax)
	2. Key1 (z. B. Sektor)
	3. Key2 (z. B. Platz)
Aufruf	asynchron
Zugriff	Read
	- Die gelesenen Daten werden als XML-Strings übergeben.
	Write
	- Die zu schreibenden Daten werden als XML-Strings übergeben.
zyklisch (opt.)	nein
Datenformat	NC.DbTableDataRecLocked_U, read

row, col	Typ	Beschreibung	NC.DbTableDataRecLocked_U, / DBT1,1,3 Liest einen gesperrten Datensatz
0,0	string	Inhalt des Datensatzes	<DBT1><Rec><Hd><K1>1</K1><K2>3</K2><SKQ></SKQ>...

NC.DbTableDataRecLocked_U, write

row, col	Typ	Beschreibung	Schreibt den kompletten Datensatz
0,0	string	neuer Inhalt	<DBT1><Rec><Hd><K1>1</K1><K2>1</K2><SKQ></SKQ>...

NC.DbTableDataRecLocked_U, write

row, col	Typ	Beschreibung	Schreibt die werkzeugspezifischen Daten
0,0	string	neuer Inhalt	<DBT1><Rec><Hd><K1>1</K1><K2>3</K2><SKQ></SKQ>...
1,0	long	Modus	LDX_DBD_SETMODE_TOOL_DATA_C

NC.DbTableEvent

Beschreibung	Meldet Schreibzugriffe auf eine DB-Tabelle, auf eine Teilstruktur oder auf einen Datensatz der Tabelle.
Anmerkung	Wird nur der erste Parameter angegeben, so werden Schreibzugriffe auf die komplette Tabelle gemeldet. Wird der zweite Parameter angegeben, kann beispielsweise ein Sektor überwacht werden. Wird auch der dritte Parameter angegeben, werden nur Schreibzugriffe auf einen Datensatz überwacht. Eine lückenlose Benachrichtigung bei Schreibvorgängen kann nicht gewährleistet werden. Daher ist auf Client-Seite eine Überprüfung der mitgelieferten Versionsnummer notwendig. Falls die Versionsnummern fortlaufend sind, ist

die Benachrichtigung lückenlos. Andernfalls wurden nicht alle Schreibzugriffe erfasst und die DB-Tabelle muss neu gelesen werden.

Fehler, die beim Suchen von Datensätzen auftreten können, sind in Abschnitt [DBD-Fehler auf Seite 228](#) beschrieben.

Item Syntax **NC.DbTableEvent,(1),[(2),[(3)]]**

1. Root-Tag
2. Key1 (optional, z. B. Sektor)
3. Key2 (optional, z. B. Platz)

Aufruf synchron / asynchron

Zugriff Read

Liefert den zuletzt geänderten Datensatz der Tabelle (Key1, Key2).

zyklisch (opt.) ja, erforderlich!

Datenformat **NC.DbTableEvent, read**

row, col	Typ	Beschreibung	NC.DbTableEvent,/DBT1,0
0,0	dword	Versionsnummer des zuletzt geänderten Datensatzes	213
1,0	long	Key1 (z. B. Sektor)	0
2,0	long	Key2 (z. B. Platz)	14

NC.DbTableLoad

Beschreibung Liest Datensätze einer DB-Tabelle aus einer XML-Datei ein. Die eingelesenen Datensätze müssen bereits in der DB-Tabelle vorhanden sein, es werden keine neuen Datensätze angelegt.

Anmerkung Sind weder Key1 noch Key2 vorgegeben, werden alle Datensätze aus der XML-Datei eingelesen. Wird Key1 vorgegeben, werden alle Datensätze mit diesem Key1 (z. B. nur Datensätze eines Sektors) gelesen. Sind Key1 und Key2 vorgegeben, so wird der Datensatz, der durch dieses Wertepaar definiert ist, gelesen.

Durch einen optionalen Modus wird gesteuert, ob die Datensätze komplett eingelesen werden oder ob nur platzspezifische oder werkzeugspezifische Daten eingelesen werden. Diese Angabe erfolgt durch den Enumerator "LDX_DbdLoadMode":

Enumerator LDX_DbdLoadMode

LDX_DbdLoadMode	Beschreibung
LDX_DBD_LOADMODE_TOOL_DATA_C	nur werkzeugspezifische Daten
LDX_DBD_LOADMODE_PLACE_DATA_C	nur platzspezifische Daten

In der XML-Datei sollte jeder Datensatz nur einmal vorkommen.

Fehler, die beim Einlesen von Datensätzen auftreten können, sind in Abschnitt [DBD-Fehler auf Seite 228](#) beschrieben.

Item Syntax **NC.DbTableLoad**

Aufruf asynchron

Zugriff Write

Beim Schreiben müssen folgende Daten übergeben werden:

- Name der Datei, aus der eingelesen wird

- Root-Tag (spezifiziert eindeutig die DB-Tabelle)
- Key1 (optional)
- Key2 (optional)
- Modus (optional)

zyklisch (opt.) nein

Datenformat NC.DbTableLoad, write

row, col	Typ	Beschreibung	Einlesen aller Datensätze (nur werkzeugspezifische Daten)
0,0	string	Datei, aus der eingelesen wird	/mnt/tools.xml
0,1	string	Root-Tag	/DBT1
0,2	long	Key1, optional	
0,3	long	Key2, optional (Datensatz)	
1,0	long	Modus, optional	LDX_DBD_LOADMODE_TOOL_DATA_C

NC.DbTableQuery_U

Beschreibung Suche von Datensätzen nach einer vorgegebenen Suchbedingung. Die Suchbedingung wird über Parameter beim Instanzieren des Items angegeben.

Anmerkung Es werden zwei unterschiedliche Formate der Suchbedingung unterstützt:

Suchbedingung im XML-Format:

- Root-Tag des XML-Strings lautet "DBTCond".
- Für die Einzelbedingungen sind die Elementbezeichner aus dem Datensatz-Header zu verwenden.
- Zur Festlegung der Maske der relevanten Bits werden die Elementbezeichner BQ1Mask, BQ2Mask und BQ3Mask verwendet.
- *Die Werte für die Bit-Arrays (z. B. BQ1) und die Bitmasken (z. B. BQ1Mask) werden folgendermaßen interpretiert:*
 - als Hexadezimalzahl, wenn sie mit "0x" oder "0X" beginnen,
 - als Oktalzahl, wenn sie mit "0" beginnen,
 - ansonsten als Dezimalzahl.

Randbedingung:

Es dürfen nur vordefinierte Tag-Namen verwendet werden.

Beispiel:

```
<DBTCond><K1>1</K1><BQ2>1</BQ2><BQ2Mask>1</BQ2Mask></DBTCond>
```

Suchbedingung in Listenform:

- Die Liste enthält Einzelbedingungen, die alle erfüllt sein müssen.
- Die Einzelbedingungen werden durch Komma voneinander getrennt.
- Eine normale Einzelbedingung hat folgende Form: <Tagname><Vergleichsoperator><Wert> (z. B. SKQ=4711)
- *Folgende Vergleichsoperatoren sind erlaubt:*
 - "=" gleich
 - ">" größer

Items

- ">=" größer gleich
- "<" kleiner
- "<=" kleiner gleich
- "<>" ungleich

Einschränkung für SKQ: Es sind nur "=" und "<>" erlaubt.

- Eine Bitmasken-Einzelbedingung hat folgende Form: <Tagname><Wert><Maske> (z. B. BQ1=0x307000:0xF0FF00).
 - <Wert> enthält den Wert der relevanten Bits
 - <Maske> enthält die Maske der relevanten Bits

Achtung: Als Vergleichsoperator ist nur "=" erlaubt.
- Werte und Masken werden folgendermaßen interpretiert:
 - als Hexadezimalzahl, wenn sie mit "0x" oder "0X" beginnen,
 - als Oktalzahl, wenn sie mit "0" beginnen,
 - ansonsten als Dezimalzahl.
- Innerhalb von <Wert> werden vorlaufende und nachlaufende Leerzeichen nicht ausgewertet.

Soll ein String mit solchen Leerzeichen vorgegeben werden, so muss der Teilstring in Hochkommata eingeschlossen werden (z. B. SKQ=' ')
- Um in einem mit Hochkommata eingeschlossenen Teilstring ein Hochkomma zu verwenden, muss diesem das Fluchtzeichen "\" vorangestellt werden (z. B. SKQ='Achim\'s')

Randbedingung:

Es dürfen nur vordefinierte Tag-Namen verwendet werden.

Beispiel:

K1=1,IKQ2=3,BQ2=0x1:0x1

Fehler, die beim Suchen von Datensätzen auftreten können, sind in Abschnitt [DBD-Fehler auf Seite 228](#) beschrieben.

Item Syntax **NC.DbTableQuery_U,(1),...(n)**

1. Root-Tag

2. .. n. Suchbedingung im XML-Format oder Suchbedingung in Listenform

Aufruf asynchron

Zugriff Read

Liefert eine Liste aller Datensätze, die der Suchbedingung genügen. Pro Datensatz wird der komplette Header geliefert.

zyklisch (opt.) nein

Datenformat **NC.DbTableQuery_U, read - Suche im XML-Format**

row, col	Typ	Beschreibung	NC.DbTableQuery_U,/ DBT1,K1=1,IKQ2<>3005 Suche alle Datensätze mit (K1=1) und (IKQ2<>3005)
0,0	string	Suchergebnis	<DBT1> ... </DBT1>

NC.DbTableQuery_U, read - Suche in Listenform

row, col	Typ	Beschreibung	NC.DbTableQuery_U, DBT1,K1=1,IQ2=3 Suche alle Datensätze mit (K1=1) und (IQ2=3)
0,0	string	Suchergebnis	<DBT1> ... </DBT1>

NC.DbTableSave

Beschreibung Sichert Datensätze einer DB-Tabelle in eine XML-Datei. Es können die komplette DB-Tabelle, mehrere Datensätze (z.B. ein Sektor einer Werkzeug-tabelle) oder einzelne Datensätze gesichert werden. Falls die XML-Datei bereits besteht, können die zu sichernden Daten am Ende angehängt werden oder der Dateinhalt wird zuvor gelöscht.

Anmerkung Es ist möglich, nur die Datensätze eines Sektors (Key1) oder genau einen Datensatz (Key1 und Key2) zu sichern. Falls weder Key1 noch Key2 angegeben werden, wird die komplette DB-Tabelle gesichert.

Fehler, die beim Suchen von Datensätzen auftreten können, sind in Abschnitt [DBD-Fehler auf Seite 228](#) beschrieben.

Item Syntax NC.DbTableSave

Aufruf asynchron

Zugriff Write

Beim Schreiben enthalten die Daten folgende Angaben:

- Name der XML-Datei (inkl. Pfad und Extension)
- Root-Tag (spezifiziert eindeutig die DB-Tabelle)
- Anhängen Ja/Nein (Anhängen an eine bestehende Datei oder Überschreiben)
- Key1 (optional)
- Key2 (optional)

zyklisch (opt.) nein

Datenformat NC.DbTableSave, write

row, col	Typ	Beschreibung	Sichert einen Datensatz, Anhängen an /mnt/tools.xml
0,0	string	Name der XML-Datei	/mnt/tools.xml
0,1	string	Root-Tag	/DBT1
0,2	long	0 = Überschreiben 1 = Anhängen	1
0,3	long	Key1, optional (Sektor)	3
0,4	long	Key2, optional (Datensatz)	2

4.7.6 Maschinenparameter

NC.MacodaParameter

Beschreibung Liefert das Datum des angeforderten MACODA-Parameters eines Kanals oder einer Spindel.



MACODA-Zugriffe auf Spindelparameter erfolgen null-basiert!

Item Syntax	NC.MacodaParameter,(1),(2) 1. Parameternummer 2. Kanal- od. Spindelnummer (optional)
Aufruf	synchron / asynchron
Zugriff	Read Das Antwortformat variiert je nach Art des Parameters.
zyklisch (opt.)	nein
Datenformat	Zeilenanzahl abhängig von MACODA

NC.MacodaParameter

row, col	Typ	Beschreibung	NC.MacodaParameter, 600500060,1
n,0	string	Datum des MACODA-Parameters	G54.1

NC.CfgData_U

Beschreibung	Unterstützt den Lese- und Schreibzugriff auf die Konfigurationsdaten. Ein Parameter bestimmt die zugänglichen Konfigurationsdatenelemente.
Item Syntax	NC.CfgData_U,(1) 1. Konfigurationselement(e)
Aufruf	asynchron
Zugriff	Read - Beim Lesen werden die Item-Daten als XML-String zurückgegeben. Write - Beim Schreibzugriff werden die Item-Daten als XML-String übergeben. Die Datengröße ist auf 8 kByte beschränkt.
zyklisch (opt.)	nein
Datenformat	NC.CfgData_U,/NCO/Vel/Ch[2]/Fact, read

row, col	Typ	Beschreibung	Lesen des Konfigurationsdatenelementes /NCO/Vel/Ch[2]/Fact
0,0	string	Konfigurationselement(e)	<Fact><FactG94>1<\FactG94><FactRotG70>1<\FactRotG70><\Fact>

NC.CfgData_U,(1) write

row, col	Typ	Beschreibung	Setzt das Konfigurationsdatenelemente /NCO/Vel/Ch[2]/Fact/FactRotG70 auf 2
0,0	string	Konfigurationselement	<FactRotG70>2<\FactRotG70>

NC.CfgDataBackup

Beschreibung	Ablage und Übernahme von Konfigurationsdaten im Standard-Dateiformat.
Anmerkung	Der Enumerator "LDX_McaBackupOption" ermöglicht verschiedene Zugriffe und Ablagen auf Dateien:

Enumerator LDX_McaBackupOption

LDX_McaBackupOption	Beschreibung
LDX_MCA_UNLOAD_PAR_C	veränderte Parameter in Datei schreiben
LDX_MCA_RELOAD_PAR_C	Parameterdatei laden
LDX_MCA_UNLOAD_PAR_FEPROM_C	Parameter in "usrfep/mcadbpar.txt" sichern
LDX_MCA_RELOAD_PAR_IF_EMPTY_C	Parameter-Datei nur laden, wenn Standardeinstellung
LDX_MCA_UNLOAD_PAR_ALL_C	alle Parameter in Datei sichern



Beim Schreiben des Items kann eine Zielfeile angegeben werden. Wird keine Zielfeile angegeben, so werden alle Parameter unter "usrfep/mcadbpar.txt" gesichert.

Item Syntax	NC.CfgDataBackup
Aufruf	asynchron
Zugriff	Write
zyklisch (opt.)	nein
Datenformat	NC.CfgDataBackup, write

row, col	Typ	Beschreibung	Konfigurationsdaten unter "usrfep/mcadbpar.txt" ablegen
0,0	long	Zielfeile	
0,1	string	Enumerator "LDX_McaBackupOption"	

NC.CfgDataEvent

Beschreibung	Unterstützt die Benachrichtigung aufgrund veränderter Konfigurationsdaten. Ein Parameter (in Xpath-Syntax) bestimmt das Konfigurationsdatenelement oder den Knoten im Konfigurationsdatenbaum, für den die Meldungen notwendig sind.
Anmerkung	Wenn das Konfigurationsdatenelement oder ein Unterelement des genannten Knotens sich verändert, wird die Version inkrementiert. Das Item liefert die aktuelle Version und ein String, der das veränderte Element spezifiziert. Mit dem Item "NC.CfgData" können die aktuellen Konfigurationsdaten gelesen werden.
Item Syntax	NC.CfgDataEvent,(1) 1. Konfigurationselement
Aufruf	synchron / asynchron
Zugriff	Read
	Es wird die aktuelle Version und ein String in Xpath-Syntax geliefert.
zyklisch (opt.)	ja, erforderlich!

Datenformat NC.CfgDataEvent, read

row, col	Typ	Beschreibung	NC.CfgDataEvent,/AXSP Meldungsanfrage über veränderte Konfigurationsdaten von Achse oder Spindeln
0,0	dword	aktuelle Version	213
0,1	string	verändertes Element oder Unter-element	/AXSP/Dr[2]/AxFun/Pos/TravLim/PosLim

NC.CfgDataLoad

Beschreibung Lädt die Konfigurationsdaten aus einer XML-Datei. Alle Konfigurationsdateneinträge der Datei werden in der Konfigurationsdatenbank gespeichert.

Anmerkung *Folgende Werte werden benötigt:*

- Name der zu lesenden XML-Datei (Pfad, Name, Erweiterung)
- Root-Tag
- Modus

Der Enumerator "LDX_CfgDataMode" dient dem optionalen Abgleich der XML-Daten vor dem Laden:

Enumerator LDX_CfgDataMode

LDX_CfgDataMode	Beschreibung
0x0	Standard-Mode XML-Datei einlesen in MACODA
LDX_CFG_LOADMODE_LOAD_C (= 1)	in MACODA einlesen (= Standard-Mode)
LDX_CFG_LOADMODE_TEST_C (= 2)	prüft Gültigkeit des Inhalts
LDX_CFG_LOADMODE_SYNC_C (= 16)	binäre XML-Daten mit MACODA synchronisieren

Item Syntax NC.CfgDataLoad

Aufruf asynchron

Zugriff Write

Lädt Konfigurationsdaten aus XML-Datei.

zyklisch (opt.) nein

Datenformat NC.CfgDataLoad, write

row, col	Typ	Beschreibung	Laden der Konfigurationsdaten aus /mnt/Bsp_AX.xml
0,0	string	Name der XML-Datei	/mnt/Bsp_AX.xml
0,1	string	Root-Tag	/AX
0,2	long	Modus	LDX_CFG_LOADMODE_LOAD_C

NC.CfgDataSave

Beschreibung Speichert die Konfigurationsdaten in eine XML-Datei. Wenn die Datei existiert, wird sie überschrieben.

Anmerkung *Folgende Werte werden benötigt:*

- Name der XML-Datei (Pfad, Name, Erweiterung)

	• Root-Tag
Item Syntax	NC.CfgDataSave
Aufruf	asynchron
Zugriff	Write
	Es wird eine XML-Datei geladen.
zyklisch (opt.)	nein
Datenformat	NC.CfgDataSave, write

row, col	Typ	Beschreibung	Speichert die Konfigurationsdaten /AX in XML-Datei /mnt/ax.xml
0,0	string	Name der XML-Datei	/mnt/ax.xml
0,1	string	Root-Tag	/AX

4.7.7 Systemdaten

Funktion

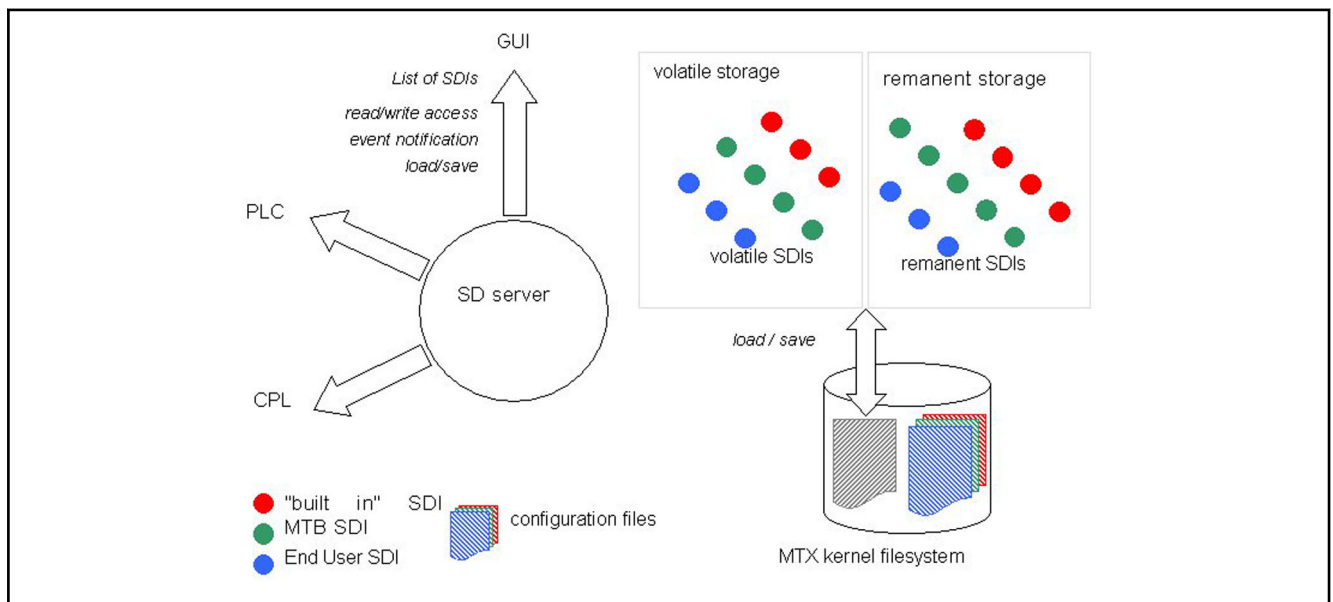


Abb. 4-1: SDI Übersicht

Der Begriff Systemdaten bezeichnet einen Satz von Daten-Items und ist folgendermaßen gekennzeichnet:

- Alle Systemdaten-Items (SDIs) werden vom Systemdaten-Server (SD-Server) innerhalb des MTX-Kerns verwaltet.
- Der SD-Server hat einen remanenten und einen volatilen Speicher.
- SDIs, die im volatilen Speicher sitzen, werden bei jeder Inbetriebnahme vom MTX-Kern gestartet, obwohl die SDIs im remanenten Speicher ihre Werte halten und nicht von Abschalten und Inbetriebnahme beeinflusst werden.
- Der MTX-Kern liefert einige "eingebaute" SDIs, die nicht konfiguriert werden müssen.
- Zusätzliche SDIs können vom Werkzeugmaschinenhersteller oder vom Endverbraucher definiert werden. Die Anzahl der SDIs, ihre Größe und Struktur sind frei konfigurierbar innerhalb der Auswahl der verfügbaren Speicher.

- Die Konfigurationsdateien "SDDefMTB.xml" und "SDDefUsr.xml" werden zur Definition von SDIs genutzt. Eingebaute SDIs werden in "SDDef.xml" definiert.
- *Jeder Eintrag in der Konfigurationsdatei beschreibt die folgenden Attribute eines SDI:*
 - Name (max. 16 Zeichen)
 - Anzahl der Datenelemente (z.B. ein Datenelement pro Kanal)
 - Größe (z.B. Größe der Arrays)
 - Struktur
 - Speicherklasse (remanent/volatil)
- SDIs sind von CPL, SPS-Programm und GUI aus zugänglich.
- Der GUI-Zugang basiert auf XML. In der Xpath-Bezeichnung wird ein String verwendet um ein spezielles SDI zu erreichen; der Datenaustausch basiert auf XML-Streams.
- Der SDI-Server liefert einen Ereignismechanismus für SDIs; ein Client kann z.B. die automatische Meldung bis zur Veränderung der SDIs anfragen.
- Es wird eine komplette Liste aller SDIs zur Verfügung gestellt.
- SDIs können in einer XML-Datei innerhalb des Kern-Dateisystems gespeichert oder von einer Datei geladen werden.
- Zukünftig wird der Zugang zu den SDIs durch Zugriffsrechte eingeschränkt.

Die hier dokumentierten SCP-Items unterstützen den XML-basierten Zugang zu den Systemdaten der MTX. Die folgenden Funktionen sind notwendig um den Zugang zu den Systemdaten zu ermöglichen.

Anmelden/Abmelden

Da der Zugang zu den Konfigurationsdaten geschützt werden soll, ist das An- und Abmelden beim MTX-Systemdatenserver notwendig. Beim Anmelden wird die aktuelle Benutzerebene übertragen und der Konfigurationsserver antwortet mit einer Client-ID, die für die folgenden Transaktionen genutzt wird.

NC.SystemData_U

Beschreibung Lese- und Schreibzugriff auf ein Systemdaten-Item

Anmerkung Bei der Instanziierung wird ein Parameter bereitgestellt, der das Systemdaten-Item oder eines seiner Unterelemente angibt.



Einschränkung: Das Item ist **nicht** für eine zyklische Abfrage optimiert!

Item Syntax **NC.SystemData_U,(1)**

1. Systemdaten-Item oder Unterelement (Xpath-Syntax, Stringlänge max. 99 Zeichen)

Aufruf asynchron

Zugriff Read

- Die Daten werden als Xml-Stream gelesen. Dabei wird die letzte Komponente des Xpath-Ausdrucks als TAG-Rahmen benutzt. Werden mehrdimensionale Arrays gelesen, werden die Daten in vordefinierte TAGs <SD></SD> eingeschlossen.

Write

- Die Daten werden als Xml-Stream übergeben. Es können einzelne Elemente als auch eine komplette Unterstruktur des Systemdatums geschrieben werden.

- Das Schreiben von vollständigen indizierten Strukturen ist im Gegensatz zum Lesen nicht möglich.

zyklisch (opt.)

nein

Datenformat

NC.SystemData_U, read (Einzelelement)

row, col	Typ	Beschreibung	Bsp.: NC.SystemData_U, Sys_VAR[1,2,2] Lesen des Strukturelementes [1,2,2] vom Systemdatum /Sys_VAR
0,0	string	Daten des Systemdaten-Items	<Sys_VAR Dimension="1,2,2">0.00000</Sys_VAR>

NC.SystemData_U, write (Einzelelement)

row, col	Typ	Beschreibung	Bsp.: NC.SystemData_U, Sys_VAR[1,2,2] Schreiben des Strukturelementes [1,2,2] von Systemdatum /Sys_VAR
0,0	string	Zu schreibende Daten des Systemdaten-Items	<Sys_VAR>1.00000</Sys_VAR>

NC.SystemData_U, read (Unterstruktur)

row, col	Typ	Beschreibung	Bsp.: NC.SystemData_U, Sys_VARArray[1] /ArrayBsp Lesen der Unterstruktur Sys_VARArray[1] /ArrayBsp vom Systemdatum / Sys_VARArray
0,0	string	Daten des Systemdaten-Items	<Sys_VARArray Dimension=="1"><ArrayBsp>ArrayElem1>1</ArrayElem1><ArrayElem2>2</ArrayElem2><ArrayElem3>3</ArrayElem3></ArrayBsp></Sys_VARArray>

NC.SystemData_U, write (Unterstruktur)

row, col	Typ	Beschreibung	Bsp.: NC.SystemData_U, Sys_VARArray[1] /ArrayBsp Schreiben der Unterstruktur Sys_VARArray[1] /ArrayBsp von Systemdatum /Sys_VARArray
0,0	string	Daten des Systemdaten-Items	<Sys_VARArray Dimension=="1"><ArrayBsp><ArrayElem1>44</ArrayElem1><ArrayElem2>55</ArrayElem2><ArrayElem3>66</ArrayElem3></ArrayBsp></Sys_VARArray>

NC.SystemData_U, read (komplettes Systemdatum)

row, col	Typ	Beschreibung	Bsp.: NC.SystemData_U,/ Sys_VARRAY Lesen des vollständigen Systemdatums /Sys_VARRAY
0,0	string	Daten des Systemdaten-Items	<Sys_VARRAY Dimension=="1"><ArrayBsp><ArrayElem1>1</ArrayElem1><ArrayElem2>2</ArrayElem2><ArrayElem3>3</ArrayElem3></ArrayBsp></Sys_VARRAY> <Sys_VARRAY Dimension=="2"><ArrayBsp><ArrayElem1>21</ArrayElem1><ArrayElem2>22</ArrayElem2><ArrayElem3>23</ArrayElem3></ArrayBsp></Sys_VARRAY>

NC.SystemDataEvent

Beschreibung Meldet veränderte Systemdaten

Anmerkung Bei der Instanziierung wird ein Parameter bereitgestellt, der das Systemdaten-Item oder Unterelement festlegt, für welches Änderungen gemeldet werden sollen.

Wird das Systemdaten-Item oder ein Unterelement davon verändert, so wird die Version inkrementiert. Wurde die Version verändert, müssen die Systemdaten mit dem Item "NC.SystemData_U" gelesen werden.

Um über Schreibzugriffe auf beliebige Systemdaten informiert zu werden, benutzen Sie den Xpath "/".

Sie können nur den Ursprung ("/") oder ein Ursprungselement benutzen, um den zu beobachtenden Ort festzulegen. Maßnahmen an Unterelementen eines Systemdaten-Items werden nicht unterstützt.

Item Syntax **NC.SystemDataEvent,(1)**

1. Systemdaten-Item oder Ursprung (Xpath-Syntax, Stringlänge max. 99 Zeichen)

Aufruf synchron / asynchron

Zugriff Read

Die aktuelle Version wird zurückgegeben sowie ein String, der das Systemdaten-Item oder seine veränderten Unterelemente angibt.

zyklisch (opt.) ja, erforderlich!

Datenformat **NC.SystemDataEvent**

row, col	Typ	Beschreibung	Bsp.: NC.SystemDataEvent,/ MySDI
0,0	dword	aktuelle Version	213
1,0	string	verändertes Systemdaten-Item oder Unterelement	/MySDI/SubComp/IntVal

NC.SystemDataLoad

Beschreibung Liest die Systemdaten aus einer XML-Datei. Alle Systemdaten-Elemente dieser Datei werden in den Speicher des Systemdaten-Servers geladen.

Anmerkung Die Systemdaten-Elemente können entweder komplett oder selektiv aus der Datei in den Speicher des Systemdaten-Servers geladen werden.

Bei der Erzeugung des Items werden keine Parameter festgelegt.

Beim Schreiben werden folgende vier Werte benötigt:

- Name der zu lesenden XML-Datei (Pfad, Name, Erweiterung)
- Name eines oder mehrerer Systemdaten als XPath-Vorgabe (durch einzelne XPath-Elemente, die mit dem Trennzeichen '+' und ohne weitere Leerzeichen zu separieren sind, können ein oder mehrere SD selektiv geladen werden). Wird nur das Rootzeichen "/" angegeben, werden alle vorhandenen Systemdaten abgelegt.
- Modus
Der Enumerator "LDX_SDMode" ermöglicht eine weitere Selektion der gewünschten Systemdaten
- Benutzer-ID

Enumeratoren LDX_SDMode

LDX_SDMode	Beschreibung
0	Standardmodus

Item Syntax NC.SystemDataLoad

Aufruf asynchron

Zugriff Write

Lädt Systemdaten der angegebenen XML-Datei

zyklisch (opt.) nein

Datenformat NC.SystemDataLoad, write

row, col	Typ	Beschreibung	Bsp.
0,0	string	Name der XML-Datei	/mnt/sdis.xml
0,1	string	Name mehrerer Systemdaten als XPath-Vorgabe	/Foo+/myVel
0,2	long	Modus	0
0,3	long	Benutzer-ID	0

NC.SystemDataSave

Beschreibung Speichert die Systemdaten in eine XML-Datei. Wenn die Datei schon existiert, wird sie überschrieben.

Anmerkung Die Systemdaten können entweder komplett oder selektiv in eine Datei geschrieben werden.

Bei der Erzeugung des Items werden keine Parameter festgelegt.

Beim Schreiben werden folgende vier Werte benötigt:

- Name der XML-Datei (Pfad, Name, Erweiterung)
- Name eines oder mehrerer Systemdaten sowie evtl. Unterelemente als XPath-Vorgabe (durch einzelne XPath-Elemente, die mit dem Trennzei-

chen '+' und ohne weitere Leerzeichen zu separieren sind, können ein oder mehrere SD selektiv gespeichert werden). Wird nur das Rootzeichen "/" angegeben, werden alle vorhandenen Systemdaten abgelegt.

- Modus

Der Enumerator "LDX_SDMode" ermöglicht eine weitere Selektion der gewünschten Systemdaten.

- Benutzer-ID

Enumeratoren LDX_SDMode

LDX_SDMode	Beschreibung
0	Standardmodus

Item Syntax NC.SystemDataSave

Aufruf asynchron

Zugriff Write

Schreibt Systemdaten in angegebene XML-Datei

zyklisch (opt.) nein

Datenformat NC.SystemDataSave, write

row, col	Typ	Beschreibung	Bsp.
0,0	string	Name der XML-Datei	/mnt/bigdi.xml
0,1	string	Name mehrerer Systemdaten als XPath-Vorgabe	/BigSDI+/Foo+/myVel
0,2	long	Modus	0
0,3	long	Benutzer-ID	0

NC.SystemDataWatch_U

Beschreibung Unterstützt den Lese- und Schreibzugriff auf ein spezielles Systemdaten-Item.

Anmerkung Um einen XML-Stream ohne äußeres Start-Tag und End-Tag zu lesen oder zu schreiben, muss der Parameter-Mode mit "LDX_MODE_SD_NO_XML_TAGS_C" mittels logisch "oder" verknüpft werden. Dies ist nur beim Zugriff auf Einzelelemente eines Systemdatums zu empfehlen.

Item Syntax NC.SystemDataWatch_U,(1),[(2)],[(3)]

1. Systemdaten-Item oder Unterelement (XPath-Syntax, Stringlänge max. 99 Zeichen)

2. Modus, steuert das Verhalten des Items

3. Benutzer-ID, für künftigen Gebrauch

Aufruf synchron / asynchron

Zugriff Read

- Die Daten werden als XML-Stream gelesen.

- Die Datengröße ist auf 8 kByte beschränkt.



Einschränkung: Die Read-Funktion ist für eine zyklische Abfrage optimiert. Deshalb darf der Lesezugriff auf das Item nur innerhalb einer Gruppe erfolgen. Die ersten Lesezugriffe können "SCPQualityUncertain" liefern.

Write

- Die Daten werden als XML-Stream übergeben.
- Die Datengröße ist auf 8 kByte beschränkt.

zyklisch (opt.)

ja

Datenformat

NC.SystemDataWatch_U, read

row, col	Typ	Beschreibung	NC.SystemDataWatch_U/ Sys_Simulation[1], LDX_MO- DE_SD_NO_XML_TAGS_C
0,0	string	Daten des Systemdaten-Items	FALSE

4.7.8 Systemfunktionen

System.Archive

Beschreibung Erzeugt ein Archiv mit den Daten der Steuerung oder liest ein vorhandenes Archiv.

Anmerkung *Folgende Angaben sind notwendig:*

- Name des Archivs
- Name der Log-Datei
- Modus (Archiv einlesen oder Archiv erzeugen, gewünschte Teilarchive)

Der Enumerator "LDX_TarMode" gibt an, ob ein Archiv erzeugt oder eingelesen werden soll. Weiterhin wird dieser Enumerator zur Angabe der Teilarchive verwendet.

Enumerator LDX_TarMode

LDX_TarMode	Beschreibung
LDX_TAR_MODE_ALL_C (= 0)	alle Daten
LDX_TAR_MODE_RAM (= 1)	RAM-Filesystem
LDX_TAR_MODE_USRFEP_C (= 2)	Benutzer-FEPROM
LDX_TAR_MODE_MACODA_C (= 4)	MACODA
LDX_TAR_MODE_TOOLTABLE_C (= 8)	Werkzeugtabellen
LDX_TAR_MODE_DATAMODULES_C (= 16)	remanente Datenbausteine der SPS
LDX_TAR_MODE_MARKERS_C (= 32)	remanente Marker der SPS
LDX_TAR_MODE_CPLVARS_C (= 64)	permanente CPL-Variablen
LDX_TAR_MODE_TOOLTAB1_C (= 128)	1. Werkzeugtabelle (XML)
LDX_TAR_MODE_TOOLTAB2_C (= 256)	2. Werkzeugtabelle (XML)
LDX_TAR_MODE_SYSTEMDATA_C (= 512)	Systemdaten
LDX_TAR_MODE_CREATE_C (= 4096)	neues Archiv anlegen
LDX_TAR_MODE_EXTRACT_C (= 8192)	vorhandenes Archiv einlesen

Schwerwiegende Fehler, die zum Abbruch der Funktion führen (z.B. nicht genügend Speicher zum Einlesen), sind in Abschnitt [TAR-Fehler auf Seite 237](#) beschrieben.

Der Schreibvorgang ist erst nach vollständiger Archivierung beendet. Während der Archivierung werden Fehler protokolliert, die durch Lesen des Items "System.ArchiveLog" ermittelt werden können.

Item Syntax	System.Archive
Aufruf	asynchron
Zugriff	Read / Write
zyklisch (opt.)	nein
Datenformat	System.Archive, write

row, col	Typ	Beschreibung	Erzeugen eines kompletten Archivs
0,0	string	Name des Archivs	/mnt/complete.tar
0,1	string	Name der Log-Datei	/mnt/complete.log
0,2	long	Wert des Enumerators "LDX_TarMode"	LDX_TAR_MODE_CREATE_C LDX_TAR_MODE_ALL_C

System.Archive, write

row, col	Typ	Beschreibung	Einlesen aller Dateien des RAM-Filesystems aus dem Archiv
0,0	string	Name des Archivs	/mnt/complete.tar
0,1	string	Name der Log-Datei	/mnt/complete.log
0,2	long	Wert des Enumerators "LDX_TarMode"	LDX_TAR_MODE_EXTRACT_C LDX_TAR_MODE_RAM_C

System.ArchiveLog

Beschreibung Ermittelt Fehler, die beim Anlegen eines neuen Archivs oder Einlesen eines vorhandenen Archivs aufgetreten sind.

Das Item liefert eine Liste der archivierten Files bzw. Teilarchive. Jeder Listeneintrag besteht aus einem Namen, einem Fehlercode und einer englischen Erläuterung.

Anmerkung Zur Codierung der Fehler wird der Enumerator "LDX_TarLogError" verwendet. Er kann folgende Werte annehmen:

Enumerator LDX_TarLogError

LDX_TarLogError	Beschreibung
LDX_TAR_LOG_OK_C (= 0)	kein Fehler aufgetreten
LDX_TAR_LOG_NOT_YET_PROCESSED_C (= 1)	Datei noch nicht bearbeitet
LDX_TAR_LOG_STAT_FAILED_C (= 2)	Datei Status fehlerhaft
LDX_TAR_LOG_SKIPPED_C (= 3)	Datei nicht bearbeitet (kein Fehler)
LDX_TAR_LOG_PATH_NOT_OPENED_C (= 4)	Fehler beim Öffnen des Verzeichnisses
LDX_TAR_LOG_READING_C (= 5)	Fehler beim Lesen der Datei
LDX_TAR_LOG_EACCESS_C (= 6)	Zugriff auf Datei nicht möglich
LDX_TAR_LOG_EEXIST_C (= 7)	Datei existiert bereits
LDX_TAR_LOG_ENOENT_C (= 8)	Datei nicht vorhanden

LDX_TarLogError	Beschreibung
LDX_TAR_LOG_ENOMEM_C (= 9)	Kein Speicher vorhanden
LDX_TAR_LOG_EROFS_C (= 10)	Read-Only Filesystem
LDX_TAR_LOG_MODE_DIFF_C (= 11)	Schutz unterschiedlich
LDX_TAR_LOG_SIZE_DIFF_C (= 12)	Größe unterschiedlich
LDX_TAR_LOG_MTIME_DIFF_C (= 13)	Datum unterschiedlich
LDX_TAR_LOG_OTHER_DIFF_C (= 14)	andere Unterschiede
LDX_TAR_LOG_FILE_DATA_DIFF_C (= 15)	Dateidaten unterschiedlich
LDX_TAR_LOG_TIME_MODE_C (= 16)	Schutz/Zeit nicht wiederhergestellt
LDX_TAR_LOG_MKDIR_FAILED_C (= 17)	Fehler beim Anlegen des Verzeichnisses
LDX_TAR_LOG_ERR_UNKNOWN_C (= 18)	unbekannter Fehler (-> WERA)
LDX_TAR_LOG_LOADING_MACODA_C (= 19)	Fehler bei Übernahme der MACODA
LDX_TAR_LOG_LOADING_TABLE_C (= 20)	Fehler bei Übernahme der Tabellen
LDX_TAR_LOG_CORRUPTED_C (= 21)	Datei zerstört
LDX_TAR_LOG_WRITE_ENABLED_C (= 22)	Schreibschutz aufgehoben
LDX_TAR_LOG_DATAMODULES_C (= 23)	Fehler bei Übernahme der SPS-Daten
LDX_TAR_LOG_MARKERS_C (= 24)	Fehler bei Übernahme der SPS-Daten
LDX_TAR_LOG_CPLVARS_C (= 25)	Fehler bei Übernahme der CPL-Variablen
LDX_TAR_LOG_RAMFS_NOTFOUND_C (= 26)	RAM-Filesystem fehlt im Archiv
LDX_TAR_LOG_USRFEPFS_NOT-FOUND_C (= 27)	Benutzer-FEPROM fehlt im Archiv
LDX_TAR_LOG_MACODA_NOT-FOUND_C (= 28)	MACODA fehlt im Archiv
LDX_TAR_LOG_TABLE_NOTFOUND_C (= 29)	Tabellen fehlen im Archiv
LDX_TAR_LOG_DATAMODULES_NOT-FOUND_C (= 30)	remanente SPS-Daten fehlen
LDX_TAR_LOG_MARKERS_NOT-FOUND_C (= 31)	remanente SPS-Daten fehlen
LDX_TAR_LOG_CPLVARS_NOT-FOUND_C (= 32)	permanente CPL-Variablen fehlen
LDX_TAR_LOG_PLCSTATE_RUN_C (= 33)	SPS ist nicht im Zustand STOP
LDX_TAR_LOG_SYSTEMDATA_C (= 34)	Fehler bei Übernahme der Systemdaten
LDX_TAR_LOG_SYSTEMDATA_NOT-FOUND_C (= 35)	System-Daten fehlen
LDX_TAR_LOG_INVALID_K1_K2_C (= 36)	Datei enthält ungültigen Datensatz

LDX_TarLogError	Beschreibung
LDX_TAR_LOG_SCHEMA_CONFLICT_C (= 37)	Daten passen nicht zur Schema-Datei
NCS_TAR_LOG_LOA-DING_NC_CFG_DATA_C (= 38)	Übernahme der NC-Konfigurationsdaten fehlerhaft

Item Syntax **System.ArchiveLog**

Read

Datenformat Zeilenanzahl: Anzahl der Teilarchive

System.ArchiveLog, read

row, col	Typ	Beschreibung	Fehler der letzten Archivierung
n,0	string	Name des archivierten Files bzw. Teilarchivs	/
n,1	string	Erläuterung des Fehler	o.k.
n,2	string	Fehlercode	LDX_TAR_LOG_OK_C

System.ArchiveWin

Beschreibung Erzeugt ein Archiv mit den Daten der Steuerung oder liest vorhandenes Archiv ein, wobei das Archiv und das Log-File im Windows-Filesystem liegen.

Anmerkung *Folgende Angaben sind notwendig:*

- Name des Archivs
- Name der Log-Datei
- Modus (Archiv einlesen oder Archiv erzeugen, gewünschte Teilarchive)

Der Enumerator "LDX_TarMode" gibt an, ob ein Archiv erzeugt oder eingelesen werden soll. Weiterhin wird dieser Enumerator zur Angabe der Teilarchive verwendet.

Enumerator LDX_TarMode

LDX_TarMode	Beschreibung
LDX_TAR_MODE_ALL_C (= 0)	alle Daten
LDX_TAR_MODE_RAM (= 1)	RAM-Filesystem
LDX_TAR_MODE_USRFEP_C (= 2)	Benutzer-FEPROM
LDX_TAR_MODE_MACODA_C (= 4)	MACODA
LDX_TAR_MODE_TOOLTABLE_C (= 8)	Werkzeugtabellen
LDX_TAR_MODE_DATAMODULES_C (= 16)	remanente Datenbausteine der SPS
LDX_TAR_MODE_MARKERS_C (= 32)	remanente Marker der SPS
LDX_TAR_MODE_CPLVARS_C (= 64)	permanente CPL-Variablen
LDX_TAR_MODE_TOOLTAB1_C (= 128)	1. Werkzeugtabelle (XML)
LDX_TAR_MODE_TOOLTAB2_C (= 256)	2. Werkzeugtabelle (XML)
LDX_TAR_MODE_SYSTEMDATA_C (= 512)	Systemdaten

LDX_TarMode	Beschreibung
LDX_TAR_MODE_CREATE_C (= 4096)	neues Archiv anlegen
LDX_TAR_MODE_EXTRACT_C (= 8192)	vorhandenes Archiv einlesen

Schwerwiegende Fehler, die zum Abbruch der Funktion führen (z.B. nicht genügend Speicher zum Einlesen), sind in [TAR-Fehler auf Seite 237](#) beschrieben.

Der Schreibvorgang ist erst nach vollständiger Archivierung beendet. Während der Archivierung werden Fehler protokolliert, die durch Lesen des Items "System.ArchiveWinLog" ermittelt werden können.

Item Syntax	System.ArchiveWin
Aufruf	asynchron
Zugriff	Read / Write
zyklisch (opt.)	nein
Datenformat	System.ArchiveWin, write

row, col	Typ	Beschreibung	Erzeugen eines kompletten Archivs
0,0	string	Name des Archivs	c:\mnt\Archiv\all.tar
0,1	string	Name der Log-Datei	c:\mnt\Archiv\all.log
0,2	string	Wert des Enumerators "LDX_TarMode"	LDX_TAR_MODE_CREATE_C LDX_TAR_MODE_ALL_C

System.ArchiveWinLog

Beschreibung	Ermittelt Fehler, die beim Anlegen eines neuen Archivs oder Einlesen eines vorhandenen Archivs im Windows-Filesystem aufgetreten sind. Das Item liefert eine Liste der archivierten Files bzw. Teilarchive. Jeder Listeneintrag besteht aus einem Namen, einem Fehlercode und einer englischen Erläuterung.
Anmerkung	Zur Codierung der Fehler wird der Enumerator "LDX_TarLogError" verwendet. Er kann folgende Werte annehmen:

Enumerator LDX_TarLogError

LDX_TarLogError	Beschreibung
LDX_TAR_LOG_OK_C (= 0)	kein Fehler aufgetreten
LDX_TAR_LOG_NOT_YET_PROCESSED_C (= 1)	Datei noch nicht bearbeitet
LDX_TAR_LOG_STAT_FAILED_C (= 2)	Datei Status fehlerhaft
LDX_TAR_LOG_SKIPPED_C (= 3)	Datei nicht bearbeitet (kein Fehler)
LDX_TAR_LOG_PATH_NOT_OPENED_C (= 4)	Fehler beim Öffnen des Verzeichnisses
LDX_TAR_LOG_READING_C (= 5)	Fehler beim Lesen der Datei
LDX_TAR_LOG_EACCESS_C (= 6)	Zugriff auf Datei nicht möglich
LDX_TAR_LOG_EEXIST_C (= 7)	Datei existiert bereits
LDX_TAR_LOG_ENOENT_C (= 8)	Datei nicht vorhanden
LDX_TAR_LOG_ENOMEM_C (= 9)	Kein Speicher vorhanden
LDX_TAR_LOG_EROFS_C (= 10)	Read-Only Filesystem

Items

LDX_TarLogError	Beschreibung
LDX_TAR_LOG_MODE_DIFF_C (= 11)	Schutz unterschiedlich
LDX_TAR_LOG_SIZE_DIFF_C (= 12)	Größe unterschiedlich
LDX_TAR_LOG_MTIME_DIFF_C (= 13)	Datum unterschiedlich
LDX_TAR_LOG_OTHER_DIFF_C (= 14)	andere Unterschiede
LDX_TAR_LOG_FILE_DATA_DIFF_C (= 15)	Dateidaten unterschiedlich
LDX_TAR_LOG_TIME_MODE_C (= 16)	Schutz/Zeit nicht wiederhergestellt
LDX_TAR_LOG_MKDIR_FAILED_C (= 17)	Fehler beim Anlegen des Verzeichnisses
LDX_TAR_LOG_ERR_UNKNOWN_C (= 18)	unbekannter Fehler (-> WERA)
LDX_TAR_LOG_LOADING_MACODA_C (= 19)	Fehler bei Übernahme der MACODA
LDX_TAR_LOG_LOADING_TABLE_C (= 20)	Fehler bei Übernahme der Tabellen
LDX_TAR_LOG_CORRUPTED_C (= 21)	Datei zerstört
LDX_TAR_LOG_WRITE_ENABLED_C (= 22)	Schreibschutz aufgehoben
LDX_TAR_LOG_DATAMODULES_C (= 23)	Fehler bei Übernahme der SPS-Daten
LDX_TAR_LOG_MARKERS_C (= 24)	Fehler bei Übernahme der SPS-Daten
LDX_TAR_LOG_CPLVARS_C (= 25)	Fehler bei Übernahme der CPL-Variablen
LDX_TAR_LOG_RAMFS_NOTFOUND_C (= 26)	RAM-Filesystem fehlt im Archiv
LDX_TAR_LOG_USRFEPFS_NOT-FOUND_C (= 27)	Benutzer-FEPROM fehlt im Archiv
LDX_TAR_LOG_MACODA_NOT-FOUND_C (= 28)	MACODA fehlt im Archiv
LDX_TAR_LOG_TABLE_NOTFOUND_C (= 29)	Tabellen fehlen im Archiv
LDX_TAR_LOG_DATAMODULES_NOT-FOUND_C (= 30)	remanente SPS-Daten fehlen
LDX_TAR_LOG_MARKERS_NOT-FOUND_C (= 31)	remanente SPS-Daten fehlen
LDX_TAR_LOG_CPLVARS_NOT-FOUND_C (= 32)	permanente CPL-Variablen fehlen
LDX_TAR_LOG_PLCSTATE_RUN_C (= 33)	SPS ist nicht im Zustand STOP
LDX_TAR_LOG_SYSTEMDATA_C (= 34)	Fehler bei Übernahme der Systemdaten
LDX_TAR_LOG_SYSTEMDATA_NOT-FOUND_C (= 35)	System-Daten fehlen
LDX_TAR_LOG_INVALID_K1_K2_C (= 36)	Datei enthält ungültigen Datensatz

LDX_TarLogError	Beschreibung
LDX_TAR_LOG_SCHEMA_CONFLICT_C (= 37)	Daten passen nicht zur Schema-Datei
NCS_TAR_LOG_LOA- DING_NC_CFG_DATA_C (= 38)	Übernahme der NC-Konfigurationsdaten fehlerhaft

Item Syntax **System.ArchiveWinLog**

Read

Datenformat Zeilenanzahl: Anzahl der Teilarchive

System.ArchiveWinLog, read

row, col	Typ	Beschreibung	Fehler der letzten Archivierung
n,0	string	Name des archivierten Files bzw. Teilarchivs	/
n,1	string	Erläuterung des Fehler	o.k.
n,2	string	Fehlercode	LDX_TAR_LOG_OK_C

System.HardwareDescription

Beschreibung Liefert die Hardwarebezeichnung der eingesetzten Steuerung, sowie deren Leiterplatten-, Hardware- und Seriennummer.

Item Syntax **System.HardwareDescription**

Aufruf synchron / asynchron

Zugriff Read

zyklisch (opt.) nein

Datenformat **System.HardwareDescription, read**

row, col	Typ	Beschreibung	Bsp. Daten des Zielsystems
0,0	string	Bezeichnung	CMP60_2
0,1	long	LP-Nummer	170645
0,2	long	Hardware-Version	101
0,3	long	Seriennummer	5969810

System.HardwareDescription2

Beschreibung Liefert die Typenschild-Information der eingesetzten Steuerungshardware.

Anmerkung: Verfügbar ab Firmwareversion: sys Release 23.00.00

Für ältere Versionen muss das Item **System.HardwareDescription** verwendet werden.

Item Syntax **System.HardwareDescription2**

Aufruf synchron / asynchron

Zugriff Read

zyklisch (opt.) nein

Datenformat System.HardwareDescription2, read

row, col	Typ	Beschreibung	Bsp. Daten des Zielsystems
0,0	string	Bezeichnung	XM42A_1
0,1	string	Materialnummer	R911394337
0,2	string	Hardware-Index	16710
0,3	string	Seriennummer	403209584
0,4	string	Typenschlüssel	XM4203.21-21-31-63-301-NN-NNNN-MTXANNNNN

System.Language

Beschreibung Ermittelt die eingestellte Landessprache oder schaltet die Sprache um.

Item Syntax **System.Language,(1),(2)**

1. Numerische Landessprachenkennung (3-stellig) z.B. 049 für "deutsch"
2. Landessprachenkennung im Klartext (z.B. "Deutsch", "Englisch",...)



Beim Schreiben des Items **muss** die Vorgabe im Parameter 2 einer zweistelligen Länderkennung (ISO 639-1) entsprechen!

Beispiele:

de - für deutsch

en - für englisch

fr - für französisch

Aufruf synchron / asynchron

Zugriff Read / Write

zyklisch (opt.) nein

Datenformat **System.Language, read**

row, col	Typ	Beschreibung	Ermitteln der aktuellen Sprache
0,0	string	aktuelle Sprache, Nummer	049
0,1	string	aktuelle Sprache, Name	Deutsch

System.Language, write

row, col	Typ	Beschreibung	Sprachumschaltung
0,0	string	neue Sprache, Nummer	033
0,1	string	neue Sprache, Name	fr

System.MainReleases

Beschreibung Liefert Informationen über die generelle Software-Version des MTX-Kerns und eine Copyright-Meldung (Version der Basis-Software und der Kundenerweiterungen).

Item Syntax **System.MainReleases**

Aufruf synchron / asynchron

Zugriff Read

zyklisch (opt.) nein

Datenformat System.MainReleases, read

row, col	Typ	Beschreibung	Bsp.
0,0	string	Software-Version	sys Release 100.0.3, 10-03-03
0,1	string	Information zum Copyright	(C) Copyright Bosch Rexroth AG 1992 – 2003
1,0	string	Software-Version	kns Release 100.0.3, 10-03-03
1,1	string	Information zum Copyright	(C) Copyright Bosch Rexroth AG 1992 – 2003
2,0	string	Software-Version	knb Release 100.0.3, 10-03-03
2,1	string	Information zum Copyright	(C) Copyright Bosch Rexroth AG 1992 – 2003

System.PcReleases

Beschreibung Liefert Versionsinformationen zu den Windows-Softwarekomponenten, die unter Windows die Kommunikationsschnittstelle zum MTX-Kern bereitstellen. Dazu gehören die NCS-Schnittstelle (ncstyp3.dll), die Textverwaltung (tsvtyp3.dll), das Logical Device MTX (SCPBrickDevice.dll, SCPBrickComponent.dll) und FIS (FisServer.dll).

Pro Komponente werden folgende Daten geliefert:

- Name des Subsystems
- Release-Nummern (Haupt-Release.Release.Sub-Release.Patch-Release)
- Zeitstempel (UTC)

Item Syntax System.PcReleases

Aufruf synchron / asynchron

Zugriff Read

zyklisch (opt.) nein

Datenformat Zeilenanzahl: Anzahl der Komponenten

System.PcReleases, read

row, col	Typ	Beschreibung	Bsp.
0,0	string	Name der Komponente	ncstyp3.dll
0,1	string	Release-Nummern	100.0.3.0
0,2	date	Zeitstempel	10.03.03 23:51:12

System.ProVi.KaFreeze

Beschreibung Kriterienanalyse für den eingefrorenen Status einer ProVi Meldung.

Item Syntax System.ProVi.KaFreeze,(1)

1 - eindeutige Meldungs-ID; Wert der bei System.MessageStore in der Spalte 8 zurückgegeben wurde.

Aufruf asynchron

Zugriff Read

zyklisch (opt.) nein

Datenformat Die Anzahl der Zeilen richtet sich nach der Größe des Codes. Die einzelnen Zeilen haben unterschiedliche Anzahl von Spalten.

System.ProVi.KaFreeze, read

row, col	Typ	Beschreibung	Bsp.
0,0	long	Typ der POE, in der die Meldung programmiert ist 1 = Programm 2 = Funktionsblock	
0,1	string	Name der POE, in der die Meldung programmiert ist	
0,2	long	Typ des Details, in dem die Meldungen programmiert sind 1 = Implementation 2 = Aktion 3 = Transition	
0,3	string	Name des Details, leer bei Implementation	
0,4	long	Netzwerknummer oder Zeile, in der die Meldung programmiert ist	
0,5	string	Variable, die als Zuweisung für die ProVi-Meldung verwendet wird	
0,6	string	POE Instanzen-Name	
0,7	long	Art der ProVi Meldung 1 = nicht negiert, nicht speichernd 2 = negiert, nicht speichernd 3 = nicht negiert, speichernd 4 = nicht negiert, speichernd	
0,8	long	IEC Sprache 1 = ST 2 = AWL	
0,9	bool	TRUE - Kriterienanalyse konnte berechnet werden FALSE - Fehler bei der Kriterienanalyseberechnung	
0,10	bool	TRUE - IndraLogic 2G FALSE - IndraLogic 1G	

Die weiteren Zeilen sind abhängig von den gelieferten Werten:

0,8 = 2 (AWL) und 0,9 = TRUE

row, col	Typ	Beschreibung	Bsp.
1+x,0	string	Label	
1+x,1	string	Operator	

row, col	Typ	Beschreibung	Bsp.
1+x,2	string	Operand	
1+x,3	string	Status	
1+x,4	string	IO-Adresse	
1+x,5	string	Kommentar	
1+x,6	long	Variablentyp	
1+x,7	bool	TRUE - relevant, bei der reduzierten Anzeige darstellen FALSE - nicht relevant, nur bei der Gesamtanzeige darstellen	
1+x,8	string	Ersatzoperator	
1+x,9	bool	TRUE - bei der kompletten Anzeige darstellen FALSE - bei der kompletten Anzeige nicht darstellen	

0,8 = 2 (AWL) und 0,9 = FALSE

row, col	Typ	Beschreibung	Bsp.
1+x,0	string	Label	
1+x,1	string	Operator	
1+x,2	string	Operand	
1+x,3	string	Status	
1+x,4	string	IO-Adresse	
1+x,5	string	Kommentar	
1+x,6	long	Variablentyp	
1+x,7	bool	TRUE - diese Zeile enthält einen Fehler, der dazu geführt hat, dass die Berechnung nicht durchgeführt werden konnte FALSE - diese Zeile enthält keinen Fehler	
1+x,8	string	Wenn Spalte 7 = TRUE Beschreibung des Fehlers, sonst leer	
1+x,9	bool	TRUE - bei der kompletten Anzeige darstellen FALSE - bei der kompletten Anzeige nicht darstellen	

0,8 = 1 (ST)

row, col	Typ	Beschreibung	Bsp.
1+x,0	string	Code in ST	
1+x,1	string	Beschreibung des Fehlers	

System.ProVi.KaOnline

Beschreibung Ermitteln der Kriterienanalysedaten der übergebenen ProVi Meldung für den Online-Zustand. Die Berechnung erfolgt bei jeder Änderung der beteiligten Variablen der ProViMeldung.

Item Syntax **System.ProVi.KaOnline,(1)**

1 - eindeutige Meldungs-ID; Wert der bei System.MessageStore in der Spalte 8 zurückgegeben wurde.

Aufruf asynchron

Zugriff Read

zyklisch (opt.) nein

Datenformat Die Anzahl der Zeilen richtet sich nach der Größe des Codes. Die einzelnen Zeilen haben unterschiedliche Anzahl von Spalten.

System.ProVi.KaOnline, read

row, col	Typ	Beschreibung	Bsp.
0,0	long	Typ der POE, in der die Meldung programmiert ist 1 = Programm 2 = Funktionsblock	
0,1	string	Name der POE, in der die Meldung programmiert ist	
0,2	long	Typ des Details, in dem die Meldungen programmiert sind 1 = Implementation 2 = Aktion 3 = Transition	
0,3	string	Name des Details, leer bei Implementation	
0,4	long	Netzwerknummer oder Zeile, in der die Meldung programmiert ist	
0,5	string	Variable, die als Zuweisung für die ProVi-Meldung verwendet wird	
0,6	string	POE Instanzen-Name	
0,7	long	Art der ProVi Meldung 1 = nicht negiert, nicht speichernd 2 = negiert, nicht speichernd 3 = nicht negiert, speichernd 4 = nicht negiert, speichernd	
0,8	long	IEC Sprache 1 = ST 2 = AWL	

row, col	Typ	Beschreibung	Bsp.
0,9	bool	TRUE - Kriterienanalyse konnte berechnet werden FALSE - Fehler bei der Kriterienanalyseberechnung	
0,10	bool	TRUE - IndraLogic 2G FALSE - IndraLogic 1G	

Die weiteren Zeilen sind abhängig von den gelieferten Werten:

0,8 = 2 (AWL) und 0,9 = TRUE

row, col	Typ	Beschreibung	Bsp.
1+x,0	string	Label	
1+x,1	string	Operator	
1+x,2	string	Operand	
1+x,3	string	Status	
1+x,4	string	IO-Adresse	
1+x,5	string	Kommentar	
1+x,6	long	Variablentyp	
1+x,7	bool	TRUE - relevant, bei der reduzierten Anzeige darstellen FALSE - nicht relevant, nur bei der Gesamtanzeige darstellen	
1+x,8	string	Ersatzoperator	
1+x,9	bool	TRUE - bei der kompletten Anzeige darstellen FALSE - bei der kompletten Anzeige nicht darstellen	

0,8 = 2 (AWL) und 0,9 = FALSE

row, col	Typ	Beschreibung	Bsp.
1+x,0	string	Label	
1+x,1	string	Operator	
1+x,2	string	Operand	
1+x,3	string	Status	
1+x,4	string	IO-Adresse	
1+x,5	string	Kommentar	
1+x,6	long	Variablentyp	
1+x,7	bool	TRUE - diese Zeile enthält einen Fehler, der dazu geführt hat, dass die Berechnung nicht durchgeführt werden konnte FALSE - diese Zeile enthält keinen Fehler	

row, col	Typ	Beschreibung	Bsp.
1+x,8	string	Wenn Spalte 7 = TRUE Beschreibung des Fehlers, sonst leer	
1+x,9	bool	TRUE - bei der kompletten Anzeige darstellen FALSE - bei der kompletten Anzeige nicht darstellen	

0,8 = 1 (ST)

row, col	Typ	Beschreibung	Bsp.
1+x,0	string	Code in ST	
1+x,1	string	Beschreibung des Fehlers	

System.ProVi.Active

Beschreibung	Gibt für die gewählte ProVi-Meldung an, ob die Meldung noch ansteht
Item Syntax	System.ProVi.Active,(1) 1 - eindeutige Meldungs-ID; Wert der bei System.MessageStore in der Spalte 8 zurückgegeben wurde.
Aufruf	asynchron
Zugriff	Read
zyklisch (opt.)	nein
Datenformat	Eine Zeile, eine Spalte.

System.ProVi.Active, read

row, col	Typ	Beschreibung	Bsp.
0,0	bool	TRUE - steht noch an FALSE - steht nicht an	

System.Sfc.KaFreeze

Beschreibung	Kriterienanalyse für den eingefrorenen Status einer Schrittkettenmeldung. Bei Zeitfehlern können mehrere Details für den Fehler verantwortlich sein. Wird kein Detail angegeben, so wird das Detail, in dem der Fehler liegt, automatisch berechnet. Ansonsten wird die Kriterienanalyse für das übergebene Detail berechnet.
Item Syntax	System.Sfc.KaFreeze,(1),(2),(3)] 1 - eindeutige Meldungs-ID; Wert, der bei System.MessageStore in der Spalte 8 zurückgegeben wurde. 2 - [optional] Typ des Details, für den die Kriterienanalyse angefordert wird. 2 = Aktion, 3 = Transition. Nur bei Zeitfehlern. 3 - [optional] Name des Details, für den die Kriterienanalyse angefordert wird. Nur bei Zeitfehlern.
Aufruf	asynchron
Zugriff	Read

zyklisch (opt.) nein

Datenformat Die Anzahl der Zeilen richtet sich nach der Größe des Codes. Die einzelnen Zeilen haben unterschiedliche Anzahl von Spalten.

System.Sfc.KaFreeze, read

row, col	Typ	Beschreibung	Bsp.
0,0	long	Typ der POE, in der die Meldung programmiert ist 1 = Programm 2 = Funktionsblock	
0,1	string	Name der POE, in der die Meldung programmiert ist	
0,2	long	Typ des Details, in dem die Meldungen programmiert sind 1 = Implementation 2 = Aktion 3 = Transition	
0,3	string	Name des Details, leer bei Implementation	
0,4	long	Netzwerknummer oder Zeile, in der die Meldung programmiert ist Nur bei Monitorfehler, sonst leer	
0,5	long	Meldungsart 3 = Monitorfehler 4 = Zeitfehler 5 = Startvoraussetzungs-Meldung 6 = Grundstellungs-Meldung	
0,6	string	Name des Schritts, in dem die Meldung programmiert ist, nur bei Zeitfehler, sonst leer	
0,7	string	Instanzname der Schrittkette	
0,8	long	IEC Sprache 1 = ST 2 = AWL	
0,9	bool	TRUE - Kriterienanalyse konnte berechnet werden FALSE - Fehler bei der Kriterienanalyseberechnung	
0,10	long	Art des Monitorfehler 0 = Fehler bei VKE = false 1 = Fehler bei VKE = true nur bei Monitorfehler, sonst undefiniert	

Items

row, col	Typ	Beschreibung	Bsp.
0,11	bool	TRUE - IndraLogic 2G FALSE - IndraLogic 1G	
0,12	long	Nur bei Zeitfehler und wenn Typ des Detail Aktion ist, sonst undefiniert 1 = Action wird weder mit ON noch mit OFF aufgerufen 2 = Aufruf der Action mit ON 3 = Aufruf der Action mit OFF	

Die weiteren Zeilen sind abhängig von den gelieferten Werten:

0,8 = 2 (AWL) und 0,9 = TRUE

row, col	Typ	Beschreibung	Bsp.
1+x,0	string	Label	
1+x,1	string	Operator	
1+x,2	string	Operand	
1+x,3	string	Status	
1+x,4	string	IO-Adresse	
1+x,5	string	Kommentar	
1+x,6	long	Variablentyp	
1+x,7	bool	TRUE - relevant, bei der reduzierten Anzeige darstellen FALSE - nicht relevant, nur bei der Gesamtanzeige darstellen	
1+x,8	string	Ersatzoperator	
1+x,9	bool	TRUE - bei der kompletten Anzeige darstellen FALSE - bei der kompletten Anzeige nicht darstellen	

0,8 = 2 (AWL) und 0,9 = FALSE

row, col	Typ	Beschreibung	Bsp.
1+x,0	string	Label	
1+x,1	string	Operator	
1+x,2	string	Operand	
1+x,3	string	Status	
1+x,4	string	IO-Adresse	
1+x,5	string	Kommentar	
1+x,6	long	Variablentyp	

row, col	Typ	Beschreibung	Bsp.
1+x,7	bool	TRUE - diese Zeile enthält einen Fehler, der dazu geführt hat, dass die Berechnung nicht durchgeführt werden konnte FALSE - diese Zeile enthält keinen Fehler	
1+x,8	string	Wenn Spalte 7 = TRUE Beschreibung des Fehlers, sonst leer	
1+x,9	bool	TRUE - bei der kompletten Anzeige darstellen FALSE - bei der kompletten Anzeige nicht darstellen	

0,8 = 1 (ST)

row, col	Typ	Beschreibung	Bsp.
1+x,0	string	Code in ST	
1+x,1	string	Beschreibung des Fehlers	

System.Sfc.KaOnline

Beschreibung	Kriterienanalyse für den Online-Status einer Schrittkettenmeldung. Bei Zeitfehlern können mehrere Details für den Fehler verantwortlich sein. In diesem Fall muss das Detail (Parameter 2,3) angegeben werden, für das die KA berechnet werden soll.
Item Syntax	System.Sfc.KaOnline,(1)[,(2),(3)] 1 - eindeutige Meldungs-ID; Wert der bei System.MessageStore in der Spalte 8 zurückgegeben wurde 2 - [bei Zeitfehler obligatorisch] Typ des Details, für den die Kriterienanalyse angefordert wird. 2 = Aktion, 3 = Transition 3 - [bei Zeitfehler obligatorisch] Name des Details, für den die Kriterienanalyse angefordert wird
Aufruf	asynchron
Zugriff	Read
zyklisch (opt.)	nein
Datenformat	Die Anzahl der Zeilen richtet sich nach der Größe des Codes. Die einzelnen Zeilen haben unterschiedliche Anzahl von Spalten.

System.Sfc.KaOnline, read

row, col	Typ	Beschreibung	Bsp.
0,0	long	Typ der POE, in der die Meldung programmiert ist 1 = Programm 2 = Funktionsblock	
0,1	string	Name der POE, in der die Meldung programmiert ist	
0,2	long	Typ des Details, in dem die Meldungen programmiert sind 2 = Aktion 3 = Transition Nur bei Monitorfehler oder Zeitfehler, sonst leer	
0,3	string	Name des Details, leer bei Implementation	
0,4	long	Netzwerknummer oder Zeile, in der die Meldung programmiert ist Nur bei Monitorfehler, sonst leer	
0,5	long	Meldungsart 3 = Monitorfehler 4 = Zeitfehler 5 = Startvoraussetzungs-Meldung 6 = Grundstellungs-Meldung	
0,6	string	Name des Schritts, in dem die Meldung programmiert ist, nur bei Zeitfehler, sonst leer	
0,7	string	Instanzname der Schrittkette	
0,8	long	IEC Sprache 1 = ST 2 = AWL	
0,9	bool	TRUE - Kriterienanalyse konnte berechnet werden FALSE - Fehler bei der Kriterienanalyseberechnung	
0,10	long	Art des Monitorfehler 0 = Fehler bei VKE = false 1 = Fehler bei VKE = true nur bei Monitorfehler, sonst undefiniert	

row, col	Typ	Beschreibung	Bsp.
0,11	bool	TRUE - IndraLogic 2G FALSE - IndraLogic 1G	
0,12	long	Nur bei Zeitfehler und wenn Typ des Detail Aktion ist, sonst unde- finiert 1 = Action wird weder mit ON noch mit OFF aufgerufen 2 = Aufruf der Action mit ON 3 = Aufruf der Action mit OFF	

Die weiteren Zeilen sind abhängig von den gelieferten Werten:

0,8 = 2 (AWL) und 0,9 = TRUE

row, col	Typ	Beschreibung	Bsp.
1+x,0	string	Label	
1+x,1	string	Operator	
1+x,2	string	Operand	
1+x,3	string	Status	
1+x,4	string	IO-Adresse	
1+x,5	string	Kommentar	
1+x,6	long	Variablentyp	
1+x,7	bool	TRUE - relevant, bei der redu- zierten Anzeige darstellen FALSE - nicht relevant, nur bei der Gesamtanzeige darstellen	
1+x,8	string	Ersatzoperator	
1+x,9	bool	TRUE - bei der kompletten An- zeige darstellen FALSE - bei der kompletten An- zeige nicht darstellen	

0,8 = 2 (AWL) und 0,9 = FALSE

row, col	Typ	Beschreibung	Bsp.
1+x,0	string	Label	
1+x,1	string	Operator	
1+x,2	string	Operand	
1+x,3	string	Status	
1+x,4	string	IO-Adresse	
1+x,5	string	Kommentar	
1+x,6	long	Variablentyp	

Items

row, col	Typ	Beschreibung	Bsp.
1+x,7	bool	TRUE - diese Zeile enthält einen Fehler, der dazu geführt hat, dass die Berechnung nicht durchgeführt werden konnte FALSE - diese Zeile enthält keinen Fehler	
1+x,8	string	Wenn Spalte 7 = TRUE Beschreibung des Fehlers, sonst leer	
1+x,9	bool	TRUE - bei der kompletten Anzeige darstellen FALSE - bei der kompletten Anzeige nicht darstellen	

0,8 = 1 (ST)

row, col	Typ	Beschreibung	Bsp.
1+x,0	string	Code in ST	
1+x,1	string	Beschreibung des Fehlers	

System.Sfc.Mode

Beschreibung	Liefert die Betriebsart einer Schrittkette.
Item Syntax	System.Sfc.Mode,(1) 1 - Instanz der Schrittkette
Aufruf	asynchron
Zugriff	Read
zyklisch (opt.)	nein
Datenformat	Die Anzahl der Zeilen richtet sich nach der Größe des Codes. Die einzelnen Zeilen haben unterschiedliche Anzahl von Spalten.

System.Sfc.Mode, read

row, col	Typ	Beschreibung	Bsp.
0,0	long	Betriebsart der Schrittkette 0 = unbekannt 1 = Manual 2 = Auto 3 = Tipp 4 = SemiAuto 5 = Stop 6 = Fehler 7 = deaktiviert	

System.Sfc.StepInfo

Beschreibung	Gibt für den gewählten Schritt die fehlerhaften Aktionen und Transitionen an.
---------------------	---

Item Syntax	System.Sfc.StepInfo,(1),(2) 1 - Instanz der Schrittkette 2 - Name des Schrittes
Aufruf	asynchron
Zugriff	Read
zyklisch (opt.)	nein
Datenformat	Die Anzahl der Zeilen entspricht der Anzahl der Aktionen und Transitionen.

System.Sfc.StepInfo, read

row, col	Typ	Beschreibung	Bsp.
x,0	long	Typ des Details, in dem die Meldungen programmiert sind 2 = Aktion, 3 = Transition	
x,1	string	Name des Details	
x,2	bool	TRUE - Aktion oder Transition ist fehlerhaft FALSE - Aktion oder Transition ist nicht fehlerhaft	

System.Sfc.Active

Beschreibung	Gibt für die gewählte Sfc-Meldung an, ob die Meldung noch ansteht
Item Syntax	System.Sfc.Active,(1) 1 - eindeutige Meldungs-ID; Wert der bei System.MessageStore in der Spalte 8 zurückgegeben wurde
Aufruf	asynchron
Zugriff	Read
zyklisch (opt.)	nein
Datenformat	Die Anzahl der Zeilen richtet sich nach der Größe des Codes. Die einzelnen Zeilen haben unterschiedliche Anzahl von Spalten.

System.Sfc.Active, read

row, col	Typ	Beschreibung	Bsp.
0,0	bool	TRUE - steht noch an FALSE - steht nicht an	

System.Sfc.Execute

Beschreibung	Ausführbarkeitsanzeige berechnen.
Item Syntax	System.Sfc.Execute,(1) {(2),(3),(4)} 1 - Instanz der Schrittkette 2 - 1 = Ausführbarkeitsanzeige für einen Schritt berechnen, 2 = Ausführbarkeitsanzeige für eine Aktion berechnen 3 - Name des Schritts oder der Aktion 4 - 1 = Force, 2 = Force on, 3 = Force off >>> Bei Ausführbarkeitsanzeige für einen Schritt ist nur "1" zulässig.

Items

Hinweis: Das Parametertripel 2, 3 und 4 kann beliebig oft übergeben werden. Für jedes wird dann die Ausführbarkeitsanzeige berechnet.

Aufruf	asynchron
Zugriff	Read
zyklisch (opt.)	nein
Datenformat	Die Anzahl der Zeilen richtet sich nach Anzahl der übergebenen Parameter. Für jede angeforderte Berechnung gibt es eine Zeile

System.Sfc.Execute, read

row, col	Typ	Beschreibung	Bsp.
x,0	bool	TRUE - die Bewegung kann ausgeführt werden FALSE - die Bewegung kann nicht ausgeführt werden	
x,1	bool	TRUE - Fehler bei der Berechnung der Ausführbarkeit. In diesem Fall liefert die Spalte 0 auf jeden fall TRUE, oder Berechnung wurde noch nicht durchgeführt. FALSE - kein Fehler bei der Berechnung der Ausführbarkeit	
x,2	string	Fehlerhafte Variablen (durch Komma getrennt), nur wenn Spalte 2 TRUE liefert, sonst leer	

Der erste Leseaufruf für dieses Item liefert immer:

Spalte 0: TRUE, Spalte 1: TRUE, Spalte 2: leer

Erst wenn die Berechnung durchgeführt wurde, werden richtige Ergebnisse geliefert.

System.Sfc.KaExecute

Beschreibung	Kriterienanalyse für die Ausführbarkeitsanzeige berechnen.
Item Syntax	System.Sfc.KaExecute,(1),(2),(3),(4) 1 - Instanz der Schrittkette 2 - 1 = Ausführbarkeitsanzeige für einen Schritt berechnen, 2 = Ausführbarkeitsanzeige für eine Aktion berechnen 3 - Name des Schritts oder der Aktion 4 - 1 = Force, 2 = Force on, 3 = Force off >>> Bei Ausführbarkeitsanzeige für einen Schritt ist nur "1" zulässig.
Aufruf	asynchron
Zugriff	Read
zyklisch (opt.)	nein
Datenformat	Die Anzahl der Zeilen richtet sich nach der Größe des Codes. Die einzelnen Zeilen haben unterschiedliche Anzahl von Spalten.

System.Sfc.KaExecute, read

row, col	Typ	Beschreibung	Bsp.
0,0	long	Typ der POE, in der die Schrittkette programmiert ist 1 = Program 2 = Funktionsblock	
0,1	string	Name der POE in der die Meldung programmiert ist	
0,2	long	IEC Sprache 1 = ST 2 = AWL	
0,3	bool	TRUE - Kriterienanalyse konnte berechnet werden FALSE - Fehler bei der Kriterienanalyseberechnung	

Die weiteren Zeilen sind abhängig von den gelieferten Werten:

0,2 = 2 (AWL) und 0,3 = TRUE

row, col	Typ	Beschreibung	Bsp.
1 + x,0	string	Label	
1 + x,1	string	Operator	
1 + x,2	string	Operand	
1 + x,3	string	Status	
1 + x,4	string	IO-Adresse	
1 + x,5	string	Kommentar	
1 + x,6	long	Variablentyp	
1 + x,7	bool	TRUE - Relevant, bei der reduzierten Anzeige darstellen FALSE - nicht relevant, nur bei der Gesamtanzeige darstellen	
1 + x,8	string	Ersatzoperator	
1 + x,9	bool	TRUE - bei der kompletten Anzeige darstellen FALSE - bei der kompletten Anzeige nicht darstellen	

0,2 = 2 (AWL) und 0,3 = FALSE

row, col	Typ	Beschreibung	Bsp.
1 + x,0	string	Label	
1 + x,1	string	Operator	
1 + x,2	string	Operand	
1 + x,3	string	Status	

row, col	Typ	Beschreibung	Bsp.
1 + x,4	string	IO-Adresse	
1 + x,5	string	Kommentar	
1 + x,6	long	Variablentyp	
1 + x,7	bool	TRUE - Diese Zeile enthält einen Fehler der dazu geführt hat das die Berechnung nicht durchgeführt werden konnte FALSE - diese Zeile enthält keinen Fehler	
1 + x,8	string	Wenn Spalte 7 = TRUE Beschreibung des Fehlers, sonst leer	
1 + x,9	bool	TRUE - bei der kompletten Anzeige darstellen FALSE - bei der kompletten Anzeige nicht darstellen	

0,2 = 1 (ST)

row, col	Typ	Beschreibung	Bsp.
1 + x,0	string	Code in ST	
1 + x,1	string	Beschreibung des Fehlers	

System.SubsystemReleases

Beschreibung Liefert Informationen über Software-Versionen der Subsysteme im MTX-Kerns.

Pro Subsystem werden folgende Daten geliefert:

- Name des Subsystems
- Release-Nummern (Haupt-Release.Release.Sub-Release.Patch-Release)
- Zeitstempel (UTC)

Item Syntax **System.SubsystemReleases**

Aufruf synchron / asynchron

Zugriff Read

zyklisch (opt.) nein

Datenformat Zeilenanzahl: Anzahl der Subsysteme

System.SubsystemReleases, read

row, col	Typ	Beschreibung	Bsp.
0,0	string	Name des Subsystems	cmp
0,1	string	Release-Nummern	100.1.2.0
0,2	date	Zeitstempel	10.03.03 23:49:12

System.SupportedLanguages

Beschreibung Ermitteln der verfügbaren Fremdsprachen. Das Item liefert die Liste aller unterstützten Sprachen.

Item Syntax **System.SupportedLanguages**

Aufruf synchron / asynchron

Zugriff Read

zyklisch (opt.) nein

Datenformat Zeilenanzahl: Anzahl der verfügbaren Fremdsprachen

System.SupportedLanguages, read

row, col	Typ	Beschreibung	Bsp.
0,0	string	1. Fremdsprache, Nummer	049
0,1	string	1. Fremdsprache, Name	Deutsch
1,0	string	2. Fremdsprache, Nummer	044
1,1	string	2. Fremdsprache, Name	English

System.SystemTime

Beschreibung Liefert oder ändert die aktuelle Uhrzeit.

Item Syntax **System.SystemTime**

keine Parameter

Aufruf synchron / asynchron

Zugriff Read / Write

zyklisch (opt.) nur lesend

Datenformat **System.SystemTime, read**

row, col	Typ	Beschreibung	Bsp.
0,0	date	aktuelle Zeit	03.03.07 11:50:12

System.TimeSync

Beschreibung Liest den Status der Zeitsynchronisation und konfiguriert die Zeitsynchronisation.

Bei aktiver Zeitsynchronisation synchronisiert die Steuerung zyklisch ihre Zeit mit einem als Zeitquelle arbeitenden PC.

Item Syntax **System.TimeSync**

Aufruf synchron / asynchron

Zugriff Read

- Liest den Status der Zeitsynchronisation

Write

- Konfiguriert die Zeitsynchronisation (einschließlich aktivieren / deaktivieren)

- Aktiviert / deaktiviert die Zeitsynchronisation

zyklisch (opt.) nein

Datenformat System.TimeSync, read

row, col	Typ	Beschreibung	Bsp.
0,0	bool	Zeitsynchronisation aktiv	TRUE
1,0	string	IP-Adresse der Zeitquelle (max. 39 Zeichen)	10.110.248.230
1,1	long	Port der Zeitquelle (0=Default Port)	0
1,2	long	Timeout (Millisekunden)	20
1,3	long	UpdateRate	3600
2,0	date	aktuelle Zeit (UTC)	31.07.2012 12:37:12
2,1	date	letzter Synchronisationszeitpunkt (UTC)	31.07.2012 12:00:00
2,2	long	Anzahl aufgetretener Timeouts	0

System.TimeSync, write

row, col	Typ	Beschreibung	Konfigurieren der Zeitsynchronisation
0,0	bool	Zeitsynchronisation aktiv	TRUE
1,0	string	IP-Adresse der Zeitquelle (max. 39 Zeichen)	10.110.248.230
1,1	long	Port der Zeitquelle (0=Default Port)	0
1,2	long	Timeout (Millisekunden)	20
1,3	long	UpdateRate	3600

System.TimeSync, write

row, col	Typ	Beschreibung	Aktivieren der Zeitsynchronisation
0,0	bool	Zeitsynchronisation aktiv	TRUE

System.TimeZone

Beschreibung Liefert oder ändert die Einstellungen für die Zeitzone.

Item Syntax **System.TimeZone**

Aufruf synchron / asynchron

Zugriff Read / Write

zyklisch (opt.) nein

Datenformat **System.TimeZone, write**

row, col	Typ	Beschreibung	Bsp.
0,0	string	Name der Zeitzone für Normalzeit	MEZ
1,0	long	Normalzeit vor Greenwich-Zeit in sec	3600

row, col	Typ	Beschreibung	Bsp.
2,0	string	Name der Zeitzone für Sommerzeit	MESZ
3,0	long	Sommerzeit vor Greenwich-Zeit in sec	7200
4,0	long	Sommerzeit Start am Sonntag (00) der letzten Woche (05) im März (03)	0x00030500
5,0	long	Zeitpunkt der Umstellung gerechnet ab Mitternacht in sec	7200
6,0	long	Sommerzeit Ende am Sonntag (00) der letzten Woche (05) im Oktober (0a)	0x000a0500
7,0	long	Zeitpunkt der Umstellung gerechnet ab Mitternacht in sec	10800

4.8 NC - kanalabhängig

4.8.1 Allgemeine Kanalinformationen

NC.Chan.BalancedConditionState

Beschreibung	Liefert die Information, ob ein Abgleich von Achsen bzw. Spindeln stattfindet.
Anmerkung	Der bitcodierte Wert ist "1", solange der Abgleich läuft. Andernfalls meldet das Item "0" zurück.
Item Syntax	NC.Chan.BalancedConditionState,(1) 1. Kanalnummer - Integer Format, gültige Kanalnummer
Aufruf	synchron / asynchron
Zugriff	Read
zyklisch (opt.)	ja
Datenformat	NC.Chan.BalancedConditionState, read

row, col	Typ	Beschreibung	Bsp.
0,0	long	Abgleichstatus	1

NC.Chan.ChanHidden

Beschreibung	Gibt an, ob der Kanal in der Anzeige ausgeblendet wird. Einstellung erfolgt über MACODA (600500010).
Item Syntax	NC.Chan.ChanHidden,(1) 1. Kanalnummer
Aufruf	synchron / asynchron
Zugriff	Read
zyklisch (opt.)	nein

Datenformat NC.Chan.ChanHidden, read

row, col	Typ	Beschreibung	NC.Chan.ChanHidden,2
0,0	bool	Kanal ausgeblendet	TRUE

NC.Chan.ChanState

Beschreibung Gibt den aktuellen Kanalzustand wieder.

Anmerkung Neben dem Kanalzustand werden auch die Zustände der Satzverarbeitung und des Interpolators gemeldet. Für die Codierung aller Zustände werden die Enumeratoren "LDX_ChanState", "LDX_SavState" und "LDX_IpoState" verwendet. Sie können folgende Werte annehmen:

Kanalzustand: Enumerator LDX_ChanState

LDX_ChanState	Kanalzustand
LDXChanInactive	nicht aktiv
LDXChanReady	bereit
LDXChanRunning	läuft
LDXChanStopped	angehalten
LDXChanResetting	Grundstellung wird ausgeführt
LDXChanCreating	Übersetzung und Prüfung eines Teileprogramms
LDXChanReadyForNextBlock	bereit für nächste Vorgabe unter "MDI"
LDXChanError	Fehler

Satzverarbeitungszustand: Enumerator LDX_SavState

LDX_SavState	Zustand der Satzverarbeitung
LDX_SAV_OP_STATE_INACTIV_C	inaktiv - Programm kann angewählt werden
LDX_SAV_OP_STATE_READY_C	bereit - Programm kann gestartet werden
LDX_SAV_OP_STATE_RUN_C	Ein Programm oder ein NC-Satz wird abgearbeitet.
LDX_SAV_OP_STATE_WAIT_C	reserviert
LDX_SAV_OP_STATE_HOLD_C	reserviert
LDX_SAV_OP_STATE_ERROR_C	Fehler - kann nur durch Grundstellung oder Programm-Abwahl behoben werden
LDX_SAV_OP_STATE_SUSPEND_C	reserviert für interne Zwecke
LDX_SAV_OP_STATE_RESETTING_C	Grundstellung wird ausgeführt - es dürfen keine Aufträge angenommen werden, bis der Zustand auf "inaktiv" übergeht
LDX_SAV_OP_STATE_CREATING_C	Ein Programm ist angewählt und wird vorbereitet (z. B. gelinkt).
LDX_SAV_OP_STATE_CANCEL_DIST_PREP_C	"Restweg löschen" wurde ausgelöst, Aufräumarbeiten sind im Gange
LDX_SAV_OP_STATE_CANCEL_DIST_RUN_C	

LDX_SavState	Zustand der Satzverarbeitung
LDX_SAV_OP_STATE_READY_FOR_PROGSTART_C	bereit - Programm steht am Programmanfang und kann gestartet werden
LDX_SAV_OP_STATE_READY_FOR_NEXT_BLOCK_C	Bei der gepufferten NC-Satzvorgabe sind alle Sätze abgearbeitet. Es wird auf die nächste Vorgabe gewartet

Interpolatorzustand: Enumerator LDX_IpoState

LDX_IpoState	Zustand des Interpolators
LDX_IPO_OP_STATE_RUN_C	läuft
LDX_IPO_OP_STATE_SLOPEDOWN_C	fährt herunter aufgrund von Vorschub Halt
LDX_IPO_OP_STATE_STOPPED_C	hat Achsen angehalten

Item Syntax NC.Chan.ChanState,(1)

1. Kanalnummer

Aufruf synchron / asynchron**Zugriff** Read

Es wird der Zustand des Kanals, der Satzverarbeitung und des Interpolators im Kanal geliefert.

zyklisch (opt.) ja**Datenformat** NC.Chan.ChanState, read

row, col	Typ	Beschreibung	Bsp
0,0	long	Zustand des Kanals	LDXChanReady
0,1	long	Zustand der Satzverarbeitung	LDX_SAV_OP_STATE_READY_FOR_PROGSTART_C
0,2	long	Zustand des Interpolators	LDX_IPO_OP_STATE_STOPPED_C

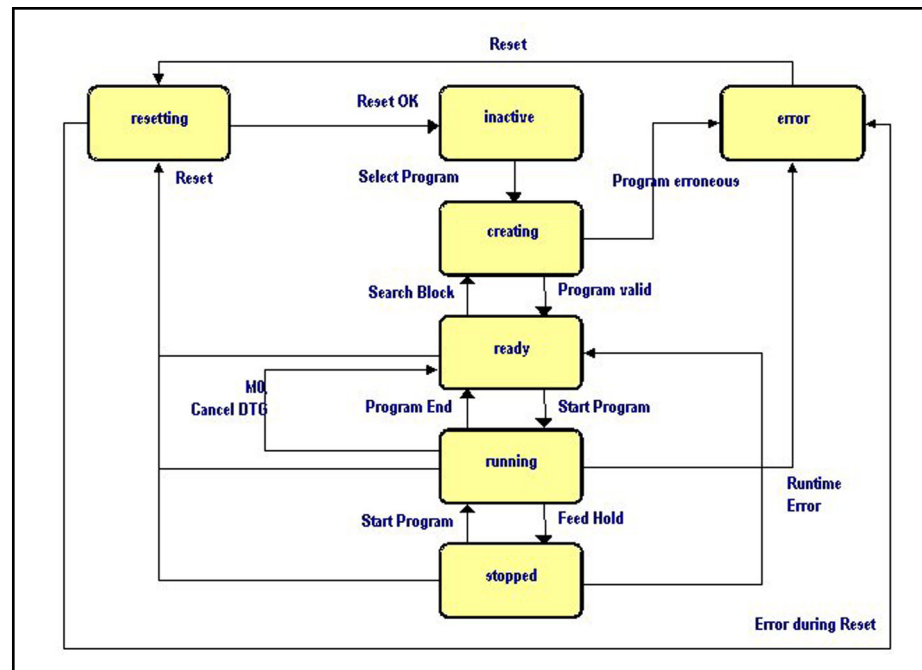


Abb. 4-2: Zustandsdiagramm bei Ausführung eines Teileprogramms

**Betriebsart MDI**

- Bei Vorgabe von Handeingabesätzen erfolgt ein Zustandsübergang von "inactive" nach "ready".
- Nach Betätigen von "Start" erfolgt ein Zustandsübergang von "ready" nach "running".
- Nach Ausführung der Handeingabesätze erfolgt der Zustandsübergang von "running" nach "ready for next block" oder ggf. "error".

NC.Chan.DryRunFeed

Beschreibung	Liefert die eingangs- und ausgangsseitig anliegenden Signalzustände der Steuerung für den Testvorschub.
Item Syntax	NC.Chan.DryRunFeed,(1) 1. Kanalnummer
Aufruf	synchron / asynchron
Zugriff	Read
zyklisch (opt.)	ja

Datenformat NC.Chan.DryRunFeed, read

row, col	Typ	Beschreibung	Bsp.
0,0	long	NC-Ausgang meldet Steuerungszustand (Mitteilung an die SPS) 0 = Testvorschub inaktiv 1 = Testvorschub aktiv	0
0,1	long	Zustand des NC-Eingangs (Aktivierungsangabe der SPS) 0 = kein Testvorschub 1 = Testvorschub aktivieren	0

NC.Chan.DryRunMode

Beschreibung Liefert den aktuellen Status der Kanalsperre.

Anmerkung Zzt. unterstützt der NC-Kern nur eine globale Gesamtsperre, die über alle Kanäle wirkt. Daher entspricht der Wert des Items "NC.Chan.DryRunMode" dem Wert des Items "NC.DryRunMode".

Item Syntax **NC.Chan.DryRunMode,(1)**

1. Kanalnummer

Aufruf synchron / asynchron

Zugriff Read

zyklisch (opt.) ja

Datenformat **NC.Chan.DryRunMode, read**

row, col	Typ	Beschreibung	Bsp.
0,0	long	Status der Kanalsperre 0 = kein Testbetrieb 1 = Testbetrieb	1

NC.Chan.DryRunRapid

Beschreibung Liefert die eingangs- und ausgangsseitig anliegenden Signalzustände der Steuerung für den Testeilgang.

Item Syntax **NC.Chan.DryRunRapid,(1)**

1. Kanalnummer

Aufruf synchron / asynchron

Zugriff Read

zyklisch (opt.) ja

Datenformat NC.Chan.DryRunRapid, read

row, col	Typ	Beschreibung	Bsp.
0,0	long	NC-Ausgang meldet Steuerungszustand (Mittellung an die SPS) 0 = Testeilgang inaktiv 1 = Testeilgang aktiv	0
0,1	long	Zustand des NC-Eingangs (Aktivierungsangabe der SPS) 0 = kein Testeilgang 1 = Testeilgang aktivieren	0

NC.Chan.FunctionBaseConfiguration

Beschreibung Liefert eine Liste aller in der Steuerung bekannten Gruppen und der dazugehörigen Funktionen.

Item Syntax **NC.Chan.FunctionBaseConfiguration,(1)**

1. Kanalnummer (optional)



Ohne Vorgabe eines Parameters liefert das Item die Information, die während des Hochlaufs vom Logischen Device ermittelt wurde. Die Angabe einer Kanalnummer führt bei jedem Aufruf zum erneuten Lesen dieser Steuerungsinformationen.

Aufruf synchron / asynchron

Zugriff Read

zyklisch (opt.) ohne Parameter ja
mit Parameter nein

Datenformat Die Liste hat momentan eine Länge von 129 Zeilen.
Die Daten der Spalte 1 sind vom Typ long.
Die Daten der Spalte 2 sind vom Typ string.
Das Blank wird als Trennzeichen zwischen mehreren NC-Funktionen genutzt.

NC.Chan.FunctionBaseConfiguration, read

row, col	Typ	Beschreibung	Bsp.
0,0	long	Gruppennummer	0
0,1	string	Funktionen der Gruppe 0	CTA
1,0	long	Gruppennummer	1
1,1	string	Funktionen der Gruppe 1	SME SMD
2,0	long	Gruppennummer	2
2,1	string	Funktionen der Gruppe 2	G80 G81 G82 G83 G84 G85 G86
n,0	long	...	...
n,1	string	...	...
128,0	long	Gruppennummer	3
128,1	string	Funktionen der Gruppe 128	KNBG1 KNBG0

NC.Chan.ProgramLinkInfo

Beschreibung Liefert das aktive Programm mit Pfadnamen und dessen Anwahlkriterien aus dem angeforderten Kanal.

Anmerkung Die Anwahlkriterien stehen als bitcodierte Enumeratoren unter "LDX_Sav-Flags" zur Verfügung. Dieser Aufzählungstyp kann folgende Werte annehmen:

Enumerator LDX_Sav_Flag

LDX_Sav_Flag	Beschreibung
LDX_SAV_FLAG_LINK_BEFORE_READY_C	Ist dieses Flag gesetzt, wird während der Anwahl ein Linklauf durchgeführt. Dabei wird eine Linktabelle für das Hauptprogramm angelegt. Diese Tabelle ist notwendig, wenn das Programm Unterprogrammaufrufe oder CPL-Elemente enthält.
LDX_SAV_FLAG_RETURN_BY_READY_C	Normalerweise wird bei der Anwahl anhand der Betriebsart und des NC-Zustands geprüft, ob die Anwahl zulässig ist. Danach wird die Anwahl quittiert. Ist dieses Flag gesetzt, erfolgt die Quittung erst nach Abschluss der Anwahl, d. h. wenn der NC-Status "bereit" ist.
LDX_SAV_FLAG_DA_C	reserviert
LDX_SAV_FLAG_REMOTE_C	reserviert
LDX_SAV_FLAG_SUPERVISOR_C	reserviert
LDX_SAV_FLAG_MDI_C	Dieses Flag muss gesetzt sein, wenn in der Betriebsart "Handeingabe" ein Satz abgearbeitet werden soll.
LDX_SAV_FLAG_PROCESSDELETE_C	Zuerst wird ein aktives Programm oder eine aktive Handeingabe abgewählt.
LDX_SAV_FLAG_AUTOSTART_C	In Verbindung mit LDX_SAV_FLAG_MDI_C: Der eingegebene Satz wird sofort gestartet. Ist der angegebene Kanal nicht aktiv, wird der Satz in diesem Kanal gestartet. Alle NC-Sätze sind dann zugelassen. Ist der angegebene Kanal aktiv, wird der Satz wie eine Schaltfunktion behandelt. Einschränkungen: -> LDX_SAV_FLAG_MACHINE_C
LDX_SAV_FLAG_JOG_C	Dieses Flag muss gesetzt sein, wenn in der Betriebsart "Tippbetrieb" Achsenverfahren werden sollen oder wenn in der Betriebsart "Einrichten" manuelle Werkstückkoordinaten-Bewegungen stattfinden sollen.
LDX_SAV_FLAG_REFPT_C	Dieses Flag muss gesetzt sein, wenn in der Betriebsart "Referenzpunkt Anfahren" Achsen gestartet werden sollen.

LDX_Sav_Flag	Beschreibung
LDX_SAV_FLAG_MACHINE_C	Schaltfunktion wirkt in Verbindung mit LDX_SAV_FLAG_MDI_C. Der Satz wird parallel zum angegebenen Kanal ausgeführt. Mit Schaltfunktion sind allerdings nur Hilfsfunktionen und asynchrone Achsbewegungen zulässig.
LDX_SAV_FLAG_PROCESSCHANGE_C	Ist dieses Flag gesetzt, wird ein bereits aktives Programm am Programmstart nicht durch Abwahl abgebrochen, sondern durch das neu angewählte ersetzt, die modalen Zustände bleiben erhalten. Bei Handeingabe wird die alte Zeichenfolge durch die neue ersetzt.
LDX_SAV_FLAG_BUFFERED_INPUT_C	Gepufferte NC-Satzvorgabe wirkt in Verbindung mit LDX_SAV_FLAG_MDI_C. Während Sätze noch abgearbeitet werden, können weitere vorgegeben werden.
LDX_SAV_FLAG_NO_CHAN_CHANGE_C	Der vorgegebene Kanal wird beibehalten, es erfolgt kein impliziter Wechsel auf den Autostart-Kanal.
LDX_SAV_FLAG_RSP_IMMEDIATELY_C	Der Auftrag wird nach Bearbeitung quittiert und nicht erst wenn Satz aktiv ist.

Item Syntax **NC.Chan.ProgramLinkInfo,(1)**

1. Kanalnummer

Aufruf synchron / asynchron

Zugriff Read

zyklisch (opt.) ja

Datenformat **NC.Chan.ProgramLinkInfo, read**

row, col	Typ	Beschreibung	Bsp.
0,0	string	aktives Programm	/mnt/anlauf-testphase/ Beispiel.npg
0,1	long	Aufrufstatus	LDX_SAV_FLAG_LINK_BEFORE_READY_C

NC.Chan.WaitingStates

Beschreibung Liefert die aktiven Wartezustände eines Kanals.

Anmerkung Für Wartezustände steht der Enumerator "LDX_McoWaitingState" zur Verfügung. Er kann folgende Werte annehmen:

Enumerator LDX_McoWaitingState

LDX_McoWaitingState	Beschreibung
LDX_McoDwellTime_C	Verweilzeit
LDX_McoAuxiliaryFunction_C	Quittungspflichtige Hilfsfunktion
LDX_McoBlockTransferInhibit_C	Einleesperre
LDX_McoChannelFeedrate0_C	Vorschub im Kanal gleich 0

LDX_McoWaitingState	Beschreibung
LDX_McoProgramStop_C	Programm Halt mit M0/M1
LDX_McoFeedHold_C	Vorschub Halt im Kanal
LDX_McoFeedInhibit_C	Vorschubsperrung in Kanal oder Kanalachse
LDX_McoCustomerStop_C	vom Kunden vorgegebene Einlesesperre
LDX_McoSynchronizedMovementStop_C	Synchronisierter Bewegungsstopp zwischen Kanälen (ASTOP, ...)
LDX_McoWaitingForAxisTransfer_C	Warten auf Achse bei Achsentauch (G511)
LDX_McoWaitingForPermanentVariable_C	Warten auf permanente Variable (WPV)
LDX_McoWaitingForIFSignalActive_C	Warten auf Interface-Signal zum aktiven Zeitpunkt (WAITA, ...)
LDX_McoWaitingForIFSignalPreparation_C	Warten auf Interface-Signal (WAIT(IC...)) oder auf eine festgelegte Zeitspanne (WAIT(,ZEIT%)) in der Satzvorbereitung
LDX_McoMotionControlDataService_C	Motion Control Datendienst (MCODS(...))
LDX_MCoWaitingForInpos_C	Warten auf Inpos der Kanalachsen
LDX_MCoFeedProgramming_C	Warten auf G95-Programmierung
LDX_MCoWaitingForSpindleG33_C	Warten bis Hauptspindel dreht G33
LDX_MCoNcProgramRestart_C	Warten auf Satzvorlauf
LDX_MCoWaitingForSpindle_C	Warten auf Drehzahl erreicht, Spindel steht, ...
LDX_MCoWaitingForEvent_C	Warten auf Event >>> Funktion WEV
LDX_MCoWaitingForInternalResource_C	Warten auf interne Ressource (z. B. Speicher)

Item Syntax **NC.Chan.WaitingStates,(1)**

1. Kanalnummer

Aufruf synchron / asynchron

Zugriff Read

zyklisch (opt.) ja

Datenformat **NC.Chan.WaitingStates, read**

row, col	Typ	Beschreibung	Bsp.
0,0	long	aktiver Wartezustand im abgefragten Kanal: "Vorschub Halt"	32 - LDX_MCoFeedHold_C

4.8.2 Achspositionen

NC.Chan.AxisPosAcs

Beschreibung Liefert die aktuellen Achspositionen bezogen auf das Achskoordinatensystem.

Item Syntax **NC.Chan.AxisPosAcs,(1)**

1. Kanalnummer

Aufruf synchron / asynchron

Zugriff Read

In den ersten sieben Spalten werden Zusatzinformationen pro Achse geliefert. Die achte Spalte enthält den Positionswert der Achse.

zyklisch (opt.) ja

Datenformat Zeilenanzahl: entspricht der Anzahl der Achsen

NC.Chan.AxisPosAcs, read

row, col	Typ	Beschreibung	Bsp.
n,0	long	phys. Achs- oder Kanalnummer	2
n,1	long	Kanalzugehörigkeit	1
n,2	string	logischer Achs- oder Kanalname	Y
n,3	long	Achse in Position 0 = FALSE 1 = TRUE	1
n,4	long	Referenzpunkt bekannt	1
n,5	long	Maßeinheit	LDX_McoUnitMetric_C
n,6	long	Durchmesserprogrammierung aktiv	0
n,7	double	Positionswert	

NC.Chan.AxisPosAcsCoupl

Beschreibung Liefert die aktuellen Achspositionen bezogen auf das Achskoordinatensystem bei Systemachskopplung.

Item Syntax NC.Chan.AxisPosAcsCoupl,(1)

1. Kanalnummer

Aufruf synchron / asynchron

Zugriff Read

In den ersten sieben Spalten werden Zusatzinformationen pro Achse geliefert. Die achte Spalte enthält den Positionswert der Achse.

zyklisch (opt.) ja

Datenformat Zeilenanzahl: entspricht der Anzahl der Achsen

NC.Chan.AxisPosAcsCoupl, read

row, col	Typ	Beschreibung	Bsp.
n,0	long	phys. Achs- oder Kanalnummer	2
n,1	long	Kanalzugehörigkeit	1
n,2	string	logischer Achs- oder Kanalname	Y
n,3	long	Achse in Position 0 = FALSE 1 = TRUE	1
n,4	long	Referenzpunkt bekannt	1
n,5	long	Maßeinheit	LDX_McoUnitMetric_C

row, col	Typ	Beschreibung	Bsp.
n,6	long	Durchmesserprogrammierung aktiv	0
n,7	double	Positionswert	

NC.Chan.AxisPosAcsDrv

Beschreibung Liefert die aktuellen Ist-Positionen der Achsen bezogen auf das Achskoordinatensystem. Die Positionen werden direkt aus den Antriebsdaten geliefert.

Item Syntax **NC.Chan.AxisPosAcsDrv,(1)**

1. Kanalnummer

Aufruf synchron / asynchron

Zugriff Read

In den ersten sieben Spalten werden Zusatzinformationen pro Achse geliefert. Die achte Spalte enthält den Positionswert der Achse.

zyklisch (opt.) ja

Datenformat Zeilenanzahl: entspricht der Anzahl der Achsen

NC.Chan.AxisPosAcsDrv, read

row, col	Typ	Beschreibung	Bsp.
n,0	long	phys. Achs- oder Kanalnummer	2
n,1	long	Kanalzugehörigkeit	1
n,2	string	logischer Achs- oder Kanalname	Y
n,3	long	Achse in Position 0 = FALSE 1 = TRUE	1
n,4	long	Referenzpunkt bekannt	1
n,5	long	Maßeinheit	LDX_McoUnitMetric_C
n,6	long	Durchmesserprogrammierung aktiv	0
n,7	double	Positionswert	

NC.Chan.AxisPosAcsDrvAdj

Beschreibung Liefert die aktuellen Ist-Positionen der Achsen bezogen auf das Achskoordinatensystem, korrigiert um Lsec, CComp und TempComp. Die Ist-Positionen werden direkt aus den Antriebsdaten geliefert.

Item Syntax **NC.Chan.AxisPosAcsDrvAdj,(1)**

1. Kanalnummer

Aufruf synchron / asynchron

Zugriff Read

In den ersten sieben Spalten werden Zusatzinformationen pro Achse geliefert. Die achte Spalte enthält den Positionswert der Achse.

zyklisch (opt.) ja

Datenformat Zeilenanzahl: entspricht der Anzahl der Achsen

NC.Chan.AxisPosAcsDrvAdj, read

row, col	Typ	Beschreibung	Bsp.
n,0	long	phys. Achs- oder Kanalnummer	2
n,1	long	Kanalzugehörigkeit	1
n,2	string	logischer Achs- oder Kanalname	Y
n,3	long	Achse in Position 0 = FALSE 1 = TRUE	1
n,4	long	Referenzpunkt bekannt	1
n,5	long	Maßeinheit	LDX_McoUnitMetric_C
n,6	long	Durchmesserprogrammierung aktiv	0
n,7	double	Positionswert	4.123

NC.Chan.AxisPosBcs

Beschreibung Liefert die aktuellen Achspositionen bezogen auf das Basiskoordinatensystem.

Item Syntax **NC.Chan.AxisPosBcs,(1)**

1. Kanalnummer

Aufruf synchron / asynchron

Zugriff Read

In den ersten sieben Spalten werden Zusatzinformationen pro Achse geliefert. Die achte Spalte enthält den Positionswert der Achse.

zyklisch (opt.) ja

Datenformat Zeilenanzahl: entspricht der Anzahl der Achsen

NC.Chan.AxisPosBcs, read

row, col	Typ	Beschreibung	Bsp.
n,0	long	phys. Achs- oder Kanalnummer	2
n,1	long	Kanalzugehörigkeit	1
n,2	string	logischer Achs- oder Kanalname	Y
n,3	long	Achse in Position 0 = FALSE 1 = TRUE	1
n,4	long	Referenzpunkt bekannt	1
n,5	long	Maßeinheit	LDX_McoUnitMetric_C
n,6	long	Durchmesserprogrammierung aktiv	0
n,7	double	Positionswert	

NC.Chan.AxisPosBcsDrv

Beschreibung Liefert die aktuellen Ist-Positionen der Achsen bezogen auf das Basiskoordinatensystem. Die Positionen werden direkt aus den Antriebsdaten geliefert.

Item Syntax	NC.Chan.AxisPosBcsDrv,(1) 1. Kanalnummer
Aufruf	synchron / asynchron
Zugriff	Read In den ersten sieben Spalten werden Zusatzinformationen pro Achse geliefert. Die achte Spalte enthält den Positionswert der Achse.
zyklisch (opt.)	ja
Datenformat	Zeilenanzahl: entspricht der Anzahl der Achsen

NC.Chan.AxisPosBcsDrv, read

row, col	Typ	Beschreibung	Bsp.
n,0	long	phys. Achs- oder Kanalnummer	2
n,1	long	Kanalzugehörigkeit	1
n,2	string	logischer Achs- oder Kanalname	Y
n,3	long	Achse in Position 0 = FALSE 1 = TRUE	1
n,4	long	Referenzpunkt bekannt	1
n,5	long	Maßeinheit	LDX_McoUnitMetric_C
n,6	long	Durchmesserprogrammierung aktiv	0
n,7	double	Positionswert	

NC.Chan.AxisPosDistAcs

Beschreibung	Liefert den Restweg bezogen auf das Achskoordinatensystem.
Item Syntax	NC.Chan.AxisPosAcsCoupl,(1) 1. Kanalnummer
Aufruf	synchron / asynchron
Zugriff	Read In den ersten sieben Spalten werden Zusatzinformationen pro Achse geliefert. Die achte Spalte enthält den Positionswert der Achse.
zyklisch (opt.)	ja
Datenformat	Zeilenanzahl: entspricht der Anzahl der Achsen

NC.Chan.AxisPosDistAcs, read

row, col	Typ	Beschreibung	Bsp.
n,0	long	phys. Achs- oder Kanalnummer	2
n,1	long	Kanalzugehörigkeit	1
n,2	string	logischer Achs- oder Kanalname	Y
n,3	long	Achse in Position 0 = FALSE 1 = TRUE	1
n,4	long	Referenzpunkt bekannt	1

row, col	Typ	Beschreibung	Bsp.
n,5	long	Maßeinheit	LDX_McoUnitMetric_C
n,6	long	Durchmesserprogrammierung aktiv	0
n,7	double	Restweg	

NC.Chan.AxisPosDistMcs

Beschreibung Liefert den Restweg bezogen auf das Maschinenkoordinatensystem.

Item Syntax **NC.Chan.AxisPosDistMcs,(1)**

1. Kanalnummer

Aufruf synchron / asynchron

Zugriff Read

In den ersten Spalten werden Zusatzinformationen pro Achse geliefert. Die letzte Spalte enthält den Restweg.

zyklisch (opt.) ja

Datenformat Zeilenanzahl: entspricht der Anzahl der Achsen

NC.Chan.AxisPosDistMcs, read

row, col	Typ	Beschreibung	Bsp.
n,0	long	phys. Achs- oder Kanalnummer	2
n,1	long	Kanalzugehörigkeit	1
n,2	string	logischer Achs- oder Kanalname	Y
n,3	long	Achse in Position 0 = FALSE 1 = TRUE	1
n,4	long	Referenzpunkt bekannt	1
n,5	long	Maßeinheit	LDX_McoUnitMetric_C
n,6	long	Durchmesserprogrammierung aktiv	0
n,7	double	Restweg	123.456

NC.Chan.AxisPosDistWcs

Beschreibung Liefert den Restweg bezogen auf das Werkstückkoordinatensystem.

Item Syntax **NC.Chan.AxisPosDistWcs,(1)**

1. Kanalnummer

Aufruf synchron / asynchron

Zugriff Read

In den ersten sieben Spalten werden Zusatzinformationen pro Achse geliefert. Die achte Spalte enthält den Restweg.

zyklisch (opt.) ja

Datenformat Zeilenanzahl: entspricht der Anzahl der Achsen

NC.Chan.AxisPosDistWcs, read

row, col	Typ	Beschreibung	Bsp.
n,0	long	phys. Achs- oder Kanalnummer	2
n,1	long	Kanalzugehörigkeit	1
n,2	string	logischer Achs- oder Kanalname	Y
n,3	long	Achse in Position 0 = FALSE 1 = TRUE	1
n,4	long	Referenzpunkt bekannt	1
n,5	long	Maßeinheit	LDX_McoUnitMetric_C
n,6	long	Durchmesserprogrammierung aktiv	0
n,7	double	Restweg	

NC.Chan.AxisPosEndWcs

Beschreibung Liefert die Endpositionen der Achsen bezogen auf das Werkstückkoordinatensystem. Die Positionen beinhalten Korrekturen, die zu dem programmierten Endwert aktiv sind.

Item Syntax **NC.Chan.AxisPosEndWcs,(1)**

1. Kanalnummer

Aufruf synchron / asynchron

Zugriff Read

In den ersten sieben Spalten werden Zusatzinformationen pro Achse geliefert. Die achte Spalte enthält den Positionswert der Achse.

zyklisch (opt.) ja

Datenformat Zeilenanzahl: entspricht der Anzahl der Achsen

NC.Chan.AxisPosEndWcs, read

row, col	Typ	Beschreibung	Bsp.
n,0	long	phys. Achs- oder Kanalnummer	2
n,1	long	Kanalzugehörigkeit	1
n,2	string	logischer Achs- oder Kanalname	Y
n,3	long	Achse in Position 0 = FALSE 1 = TRUE	1
n,4	long	Referenzpunkt bekannt	1
n,5	long	Maßeinheit	LDX_McoUnitMetric_C
n,6	long	Durchmesserprogrammierung aktiv	0
n,7	double	Endposition	

NC.Chan.AxisPosLag

Beschreibung Liefert den Nachlauf der Achsen.

Item Syntax	NC.Chan.AxisPosLag,(1) 1. Kanalnummer
Aufruf	synchron / asynchron
Zugriff	Read In den ersten sieben Spalten werden Zusatzinformationen pro Achse geliefert. Die achte Spalte enthält den Nachlauf der Achse.
zyklisch (opt.)	ja
Datenformat	Zeilenanzahl: entspricht der Anzahl der Achsen

NC.Chan.AxisPosLag, read

row, col	Typ	Beschreibung	Bsp.
n,0	long	phys. Achs- oder Kanalnummer	2
n,1	long	Kanalzugehörigkeit	1
n,2	string	logischer Achs- oder Kanalname	Y
n,3	long	Achse in Position 0 = FALSE 1 = TRUE	1
n,4	long	Referenzpunkt bekannt	1
n,5	long	Maßeinheit	LDX_McoUnitMetric_C
n,6	long	Durchmesserprogrammierung aktiv	0
n,7	double	Nachlauf	

NC.Chan.AxisPosMcs

Beschreibung	Liefert die aktuellen Achspositionen bezogen auf das Maschinenkoordinatensystem der Achsen.
---------------------	---



Bei aktiver Achstransformation (Wirkstelle 1) werden Koordinatenwerte statt Achspositionen geliefert! Die Unterscheidung ist am negativen Zahlenwert (entspr. Maschinenparameter 7080 00020 von -6 bis -1) in Spalte n,0 erkennbar.

Item Syntax	NC.Chan.AxisPosMcs,(1) 1. Kanalnummer
Aufruf	synchron / asynchron
Zugriff	Read In den ersten sieben Spalten werden Zusatzinformationen pro Achse geliefert. Die achte Spalte enthält den Positionswert der Achse.
zyklisch (opt.)	ja
Datenformat	Zeilenanzahl: entspricht der Anzahl der Achsen

NC.Chan.AxisPosMcs, read

row, col	Typ	Beschreibung	Bsp.
n,0	long	phys. Achs- oder Kanalnummer (1 bis 64) bzw. bei aktiver Achstransformation die MCS-Koordinaten (-6 bis -1)	2
n,1	long	Kanalzugehörigkeit	1
n,2	string	logischer Achs- oder Kanalname	Y
n,3	long	Achse in Position 0 = FALSE 1 = TRUE	1
n,4	long	Referenzpunkt bekannt	1
n,5	long	Maßeinheit	LDX_McoUnitMetric_C
n,6	long	Durchmesserprogrammierung aktiv	0
n,7	double	Positionswert	5.462

NC.Chan.AxisPosProg

Beschreibung Liefert die programmierten Achspositionen.

Item Syntax **NC.Chan.AxisPosProg,(1)**

1. Kanalnummer

Aufruf synchron / asynchron

Zugriff Read

In den ersten sieben Spalten werden Zusatzinformationen pro Achse geliefert. Die achte Spalte enthält den Positionswert der Achse.

zyklisch (opt.) ja

Datenformat Zeilenanzahl: entspricht der Anzahl der Achsen

NC.Chan.AxisPosProg, read

row, col	Typ	Beschreibung	Bsp.
n,0	long	phys. Achs- oder Kanalnummer	2
n,1	long	Kanalzugehörigkeit	1
n,2	string	logischer Achs- oder Kanalname	Y
n,3	long	Achse in Position 0 = FALSE 1 = TRUE	1
n,4	long	Referenzpunkt bekannt	1
n,5	long	Maßeinheit	LDX_McoUnitMetric_C
n,6	long	Durchmesserprogrammierung aktiv	0
n,7	double	Positionswert	

NC.Chan.AxisPosWcs

Beschreibung Liefert die aktuellen Achspositionen bezogen auf das Werkstückkoordinatensystem der Achsen.



Bei aktiver Achstransformation (Wirkstelle 1 oder 2) werden Koordinatenwerte statt Achspositionen geliefert! Die Unterscheidung ist am negativen Zahlenwert (entspr. Maschinenparameter 7080 00010 von -12 bis -7) in Spalte n,0 erkennbar.

Item Syntax NC.Chan.AxisPosWcs,(1)

1. Kanalnummer

Aufruf synchron / asynchron

Zugriff Read

In den ersten sieben Spalten werden Zusatzinformationen pro Achse geliefert. Die achte Spalte enthält den Positionswert der Achse.

zyklisch (opt.) ja

Datenformat Zeilenanzahl: entspricht der Anzahl der Achsen

NC.Chan.AxisPosWcs, read

row, col	Typ	Beschreibung	Bsp.
n,0	long	phys. Achs- oder Kanalnummer (1 bis 64) bzw. bei aktiver Achstransformation die WCS-Koordinaten (-12 bis -7)	2
n,1	long	Kanalzugehörigkeit	1
n,2	string	logischer Achs- oder Kanalname	Y
n,3	long	Achse in Position 0 = FALSE 1 = TRUE	1
n,4	long	Referenzpunkt bekannt	1
n,5	long	Maßeinheit	LDX_McoUnitMetric_C
n,6	long	Durchmesserprogrammierung aktiv	0
n,7	double	Positionswert	3.456

NC.Chan.AxisPosWcsDrv

Beschreibung Liefert die kartesischen Achs-Istpositionen bezogen auf das Werkstückkoordinatensystem der Achsen.

Item Syntax NC.Chan.AxisPosWcsDrv,(1)

1. Kanalnummer

Aufruf synchron / asynchron

Zugriff Read

In den ersten sieben Spalten werden Zusatzinformationen pro Achse geliefert. Die achte Spalte enthält den Positionswert der Achse.

zyklisch (opt.) ja

Datenformat Zeilenanzahl: entspricht der Anzahl der Achsen

NC.Chan.AxisPosWcsDrv, read

row, col	Typ	Beschreibung	Bsp.
n,0	long	phys. Achs- oder Kanalnummer (1 bis 64) bzw. bei aktiver Achstransformation die WCS-Koordinaten (-12 bis -7)	1
n,1	long	Kanalzugehörigkeit	1
n,2	string	logischer Achs- oder Kanalname	X
n,3	long	Achse in Position 0 = FALSE 1 = TRUE	1
n,4	long	Referenzpunkt bekannt	1
n,5	long	Maßeinheit	LDX_McoUnitMetric_C
n,6	long	Durchmesserprogrammierung aktiv	0
n,7	double	Positionswert	12.456

4.8.3 Achskorrekturen**NC.Chan.ActAxisOffsetTable**

Beschreibung Liefert den Namen und den Pfad der aktiven Achsoffset-Tabelle.

Item Syntax **NC.Chan.ActAxisOffsetTable,(1)**

1. Kanalnummer

Aufruf synchron / asynchron

Zugriff Read

zyklisch (opt.) ja

Datenformat **NC.Chan.ActAxisOffsetTable, read**

row, col	Typ	Beschreibung	Bsp.
0,0	string	Name und Pfad der Tabelle	/database/achsoffsets.aot

NC.Chan.ActDCorrectionNmb

Beschreibung Liefert die aktive D-Korrekturnummer.

Item Syntax **NC.Chan.ActDCorrectionNmb,(1)**

1. Kanalnummer

Aufruf synchron / asynchron

Zugriff Read

zyklisch (opt.) ja

Datenformat **NC.Chan.ActDCorrectionNmb, read**

row, col	Typ	Beschreibung	Bsp.
0,0	long	aktive D-Korrekturnummer	2

NC.Chan.ActDCorrectionTable

Beschreibung Liefert den Namen und den Pfad der aktiven D-Korrekturtabelle.

Item Syntax NC.Chan.ActDCorrectionTable,(1)
 1. Kanalnummer
Aufruf synchron / asynchron
Zugriff Read
zyklisch (opt.) ja
Datenformat NC.Chan.ActDCorrectionTable, read

row, col	Typ	Beschreibung	Bsp.
0,0	string	Name und Pfad der Tabelle	/database/sample.dct

NC.Chan.ActDCorrectionValues

Beschreibung Liefert die aktiven D-Korrekturwerte, d. h. die aktiven Werte für Längen- und Radiuskorrekturen sowie die Schneidenlage (Orientierung).
Item Syntax NC.Chan.ActDCorrectionValues,(1)
 1. Kanalnummer
Aufruf synchron / asynchron
Zugriff Read
zyklisch (opt.) ja
Datenformat NC.Chan.ActDCorrectionValues, read

row, col	Typ	Beschreibung	Bsp.
0,0	double	Längenkorrektur 1	2.65
0,1	double	Längenkorrektur 2	20.5
0,2	double	Längenkorrektur 3	0.109
0,3	double	Radiuskorrektur	9.04
0,4	double	Orientierung	0

NC.Chan.ActOnlineCorrValue

Beschreibung Liefert für alle Achsen des Kanals die aktiven Online-Korrekturwerte (WCS).
Anmerkung In den ersten sieben Spalten werden Zusatzdaten geliefert. Die achte Spalte enthält den Korrekturwert.
Item Syntax NC.Chan.ActOnlineCorrValue,(1)
 1. Kanalnummer
Aufruf synchron / asynchron
Zugriff Read
zyklisch (opt.) ja
Datenformat Zeilenanzahl: entspricht der Achsenanzahl

NC.Chan.ActOnlineCorrValue, read

row, col	Typ	Beschreibung	Bsp.
n,0	long	physikalische Achs- oder Koordinatennummer	2
n,1	long	Kanalzugehörigkeit	1

row, col	Typ	Beschreibung	Bsp.
n,2	string	Achs- oder Koordinatenname	Y
n,3	bool	Achse in Position	TRUE
n,4	bool	Referenzpunkt bekannt	TRUE
n,5	long	Maßeinheit	LDX_MCoUnitMetric_C
n,6	bool	Durchmesserprogrammierung	FALSE
n,7	double	aktiver Online-Korrekturwert(e)	12.345

NC.Chan.ActPlacementBcrCorr

Beschreibung	Liefert die aktiven Verschiebungs- und Winkelwerte zur Werkstücklagenkorrektur.
Anmerkung	Ist keine Werkstücklagenkorrektur aktiv, wird jeweils 0 zurückgegeben.
Item Syntax	NC.Chan.ActPlacementBcrCorr,(1) 1. Kanalnummer
Aufruf	synchron / asynchron
Zugriff	Read
zyklisch (opt.)	ja
Datenformat	NC.Chan.ActPlacementBcrCorr, read

row, col	Typ	Beschreibung	Bsp.
0,0	double	X-Offset	40
0,1	double	Y-Offset	70.6
0,2	double	Z-Offset	-5.1
0,3	double	Phi-Winkel	75
0,4	double	Theta-Winkel	45
0,5	double	Psi-Winkel	35

NC.Chan.ActPlacementData

Beschreibung	Liefert die aktiven Placement-Daten von fünf Bänken mit den zugehörigen drei Koordinatenwerten und drei Winkelwerten.
Anmerkung	Ist für eine Bank (Ebene) kein Placement aktiv, wird jeweils 0 zurückgegeben.
Item Syntax	NC.Chan.ActPlacementData,(1) 1. Kanalnummer
Aufruf	synchron / asynchron
Zugriff	Read
zyklisch (opt.)	ja
Datenformat	Zeilenanzahl: 5 (entspricht der Anzahl der Bänke)

NC.Chan.ActPlacementData, read

row, col	Typ	Beschreibung	Bsp.
n,0	double	X-Koordinate	4
n,1	double	Y-Koordinate	0.6
n,2	double	Z-Koordinate	1
n,3	double	Phi-Winkel	90
n,4	double	Theta-Winkel	45
n,5	double	Psi-Winkel	135

NC.Chan.ActPlacementSumData

Beschreibung Liefert die aktiven Summenwerte der Placements und der Werkstücklagenkorrektur.

Anmerkung Sind keine Placements und keine Werkstücklagenkorrektur aktiv, wird 0 zurückgegeben.

Item Syntax **NC.Chan.ActPlacementSumData,(1)**

1. Kanalnummer

Aufruf synchron / asynchron

Zugriff Read

zyklisch (opt.) ja

Datenformat **NC.Chan.ActPlacementSumData, read**

row, col	Typ	Beschreibung	Bsp.
0,0	double	X-Summe	25
0,1	double	Y-Summe	10.6
0,2	double	Z-Summe	0.0
0,3	double	Phi-Summe	45
0,4	double	Theta-Summe	12
0,5	double	Psi-Summe	135

NC.Chan.ActPlacementTable

Beschreibung Liefert den aktiven Tabellennamen der Placements mit Pfadangabe.

Anmerkung Die maximale Zeichenlänge beträgt 100 Zeichen (incl. Pfad).

Ist keine Tabelle aktiv, werden 3 Leerzeichen geliefert.

Item Syntax **NC.Chan.ActPlacementTable,(1)**

1. Kanalnummer

Aufruf synchron / asynchron

Zugriff Read

zyklisch (opt.) ja

Datenformat **NC.Chan.ActPlacementTable, read**

row, col	Typ	Beschreibung	Bsp.
0,0	string	Name und Pfad der Tabelle	/mnt/beispiel.pmt

NC.Chan.ActProgCoordATrans

Beschreibung	Liefert die additiven Programmkoordinatenverschiebungswerte aller Achsen des selektierten Kanals.
Anmerkung	In den ersten sieben Spalten werden Zusatzinformationen pro Achse geliefert. Jeweils die achte Spalte enthält den Verschiebungswert der Achse.
Item Syntax	NC.Chan.ActProgCoordATrans,(1) 1. Kanalnummer - Integer Format, gültige Kanalnummer
Aufruf	synchron / asynchron
Zugriff	Read
zyklisch (opt.)	ja
Datenformat	Zeilenanzahl entspricht der Achsenanzahl

NC.Chan.ActProgCoordATrans, read

row, col	Typ	Beschreibung	Bsp.
n,0	long	phys. Achs- oder Kanalnummer	1
n,1	long	Kanalzugehörigkeit	1
n,2	string	logischer Achs- oder Kanalname	X
n,3	long	Achse in Position 0 = nein 1 = ja	0
n,4	long	Referenzpunkt bekannt 0 = nein 1 = ja	1
n,5	long	Maßeinheit	LDX_McoUnitMetric_C
n,6	long	Durchmesserprogrammierung aktiv 0 = nein 1 = ja	0
n,7	double	Programmkoordinatenverschiebungswert	5.4

NC.Chan.ActProgCoordTrans

Beschreibung	Liefert die Programmkoordinatenverschiebungswerte aller Achsen des selektierten Kanals.
Anmerkung	In den ersten sieben Spalten werden Zusatzinformationen pro Achse geliefert. Jeweils die achte Spalte enthält den Verschiebungswert der Achse.
Item Syntax	NC.Chan.ActProgCoordTrans,(1) 1. Kanalnummer - Integer Format, gültige Kanalnummer
Aufruf	synchron / asynchron
Zugriff	Read
zyklisch (opt.)	ja
Datenformat	siehe NC.Chan.ActProgCoordATrans

NC.Chan.ActProgCoordTransSum

Beschreibung	Liefert die Summe der Programmkoordinatenverschiebungswerte aus TRANS und ATRANS aller Achsen des selektierten Kanals.
Anmerkung	In den ersten sieben Spalten werden Zusatzinformationen pro Achse geliefert. Jeweils die achte Spalte enthält den Summenverschiebungswert der Achse.
Item Syntax	NC.Chan.ActProgCoordTransSum,(1) 1. Kanalnummer - Integer Format, gültige Kanalnummer
Aufruf	synchron / asynchron
Zugriff	Read
zyklisch (opt.)	ja
Datenformat	siehe NC.Chan.ActProgCoordATrans

NC.Chan.ActToolCorrectionValues

Beschreibung	Liefert die aktiven Werkzeugkorrekturwerte.
Anmerkung	Es werden die Werte für Längen- und Radiuskorrektur geliefert, sowie Schneidenlage, Eulerwinkel (Orientierung im Raum) und Schneidenummer.
Item Syntax	NC.Chan.ActToolCorrectionValues,(1) 1. Kanalnummer
Aufruf	synchron / asynchron
Zugriff	Read
zyklisch (opt.)	ja
Datenformat	NC.Chan.ActToolCorrectionValues, read

row, col	Typ	Beschreibung	Bsp.
0,0	double	Längenkorrektur 1	2.65
0,1	double	Längenkorrektur 2	20.5
0,2	double	Längenkorrektur 3	0.109
0,3	double	Radiuskorrektur	9.04
0,4	double	Orientierung	0.0
0,5	double	Φ	0.0
0,6	double	ϑ	0.0
0,7	double	ψ	0.0
0,8	double	Schneidenummer	4.0

NC.Chan.ActZeroOffsetSummary

Beschreibung	Liefert für alle Achsen des Kanals die Summe der aktiven Nullpunktverschiebungswerte aller NPV-Gruppen.
Anmerkung	In den ersten sieben Spalten werden Zusatzdaten geliefert. Die achte Spalte enthält die Summe der Verschiebungswerte.
Item Syntax	NC.Chan.ActZeroOffsetSummary,(1) 1. Kanalnummer
Aufruf	synchron / asynchron
Zugriff	Read
zyklisch (opt.)	ja

Datenformat Zeilenanzahl: entspricht der Achsenanzahl

NC.Chan.ActZeroOffsetSummary, read

row, col	Typ	Beschreibung	Bsp.
n,0	long	physikalische Achs- oder Koordinatennummer	2
n,1	long	Kanalzugehörigkeit	1
n,2	string	Achs- oder Koordinatenname	Y
n,3	bool	Achse in Position	TRUE
n,4	bool	Referenzpunkt bekannt	TRUE
n,5	long	Maßeinheit	LDX_MCoUnitMetric_C
n,6	bool	Durchmesserprogrammierung	FALSE
n,7	double	Summe der Verschiebungswerte	13.5678

NC.Chan.ActZeroOffsetValues

Beschreibung Liefert für alle Achsen des Kanals die aktiven Nullpunktverschiebungswerte aller NPV-Gruppen.

Anmerkung In den ersten sieben Spalten werden Zusatzdaten geliefert.

Die achte bis zwölfte Spalte enthält die Werte der Basisgruppe sowie die Werte der ersten bis vierten additiven Gruppe.

Item Syntax **NC.Chan.ActZeroOffsetValues,(1)**

1. Kanalnummer

Aufruf synchron / asynchron

Zugriff Read

zyklisch (opt.) ja

Datenformat Zeilenanzahl: entspricht der Achsenanzahl

NC.Chan.ActZeroOffsetValues, read

row, col	Typ	Beschreibung	Bsp.
n,0	long	physikalische Achs- oder Koordinatennummer	2
n,1	long	Kanalzugehörigkeit	1
n,2	string	Achs- oder Koordinatenname	Y
n,3	bool	Achse in Position	TRUE
n,4	bool	Referenzpunkt bekannt	TRUE
n,5	long	Maßeinheit	LDX_MCoUnitMetric_C
n,6	bool	Durchmesserprogrammierung	FALSE
n,7	double	Verschiebungswert der Basisgruppe	0.04
n,8	double	Verschiebungswert der 1. additiven Gruppe	0.5
n,9	double	Verschiebungswert der 2. additiven Gruppe	0.44

row, col	Typ	Beschreibung	Bsp.
n,10	double	Verschiebungswert der 3. additiven Gruppe	0.0
n,11	double	Verschiebungswert der 4. additiven Gruppe	0.0

NC.Chan.AxisShift

Beschreibung Das Item liest die Werte, um die eine programmierte Kontur parallel zu den Achsen des Programmkoordinatensystems verschoben wurde.

Anmerkung In den ersten sieben Spalten werden Zusatzinformationen pro Achse geliefert. Jeweils die achte Spalte enthält den Verschiebungswert der Achse.

Item Syntax **NC.Chan.AxisShift,(1)**

1. Kanalnummer - Integer Format, gültige Kanalnummer

Aufruf synchron / asynchron

Zugriff Read

zyklisch (opt.) ja

Datenformat **NC.Chan.AxisShift, read**

row, col	Typ	Beschreibung	Bsp.
n,0	long	phys. Achs- oder Kanalnummer	2
n,1	long	Kanalzugehörigkeit	1
n,2	string	logischer Achs- oder Kanalname	Y
n,3	long	Achse in Position 0 = nein 1 = ja	1
n,4	long	Referenzpunkt bekannt 0 = nein 1 = ja	1
n,5	long	Maßeinheit 0 = inch 1 = metrisch 2 = Grad 4 = Platz	1
n,6	long	Durchmesserprogrammierung aktiv 0 = nein 1 = ja	0
n,7	double	Verschiebungswert	

NC.Chan.CompCrossData

Beschreibung Liefert für alle Achsen des Kanals die aktiven Kreuzkompensationswerte.

Anmerkung In den ersten sieben Spalten werden Zusatzinformationen pro Achse geliefert. Jeweils die achte Spalte enthält den Kompensationswert der Achse.

Item Syntax NC.Chan.CompCrossData,(1)
 1. Kanalnummer - Integer Format, gültige Kanalnummer
Aufruf synchron / asynchron
Zugriff Read
zyklisch (opt.) ja
Datenformat NC.Chan.CompCrossData, read

row, col	Typ	Beschreibung	Bsp.
n,0	long	phys. Achs- oder Kanalnummer	2
n,1	long	Kanalzugehörigkeit	1
n,2	string	logischer Achs- oder Kanalname	Y
n,3	long	Achse in Position 0 = nein 1 = ja	1
n,4	long	Referenzpunkt bekannt 0 = nein 1 = ja	1
n,5	long	Maßeinheit 0 = inch 1 = metrisch 2 = Grad 4 = Platz	1
n,6	long	Durchmesserprogrammierung aktiv 0 = nein 1 = ja	0
n,7	double	aktive Kreuzkompensationswerte	0.123

NC.Chan.CompLeadScrewErrData

Beschreibung Liefert für alle Achsen des Kanals die aktive Spindelsteigungsfehlerkompensation.
Anmerkung In den ersten sieben Spalten werden Zusatzinformationen pro Achse geliefert. Jeweils die achte Spalte enthält den Kompensationswert der Achse.
Item Syntax NC.Chan.CompLeadScrewErrData,(1)
 1. Kanalnummer - Integer Format, gültige Kanalnummer
Aufruf synchron / asynchron
Zugriff Read
zyklisch (opt.) ja
Datenformat NC.Chan.CompLeadScrewErrData, read

row, col	Typ	Beschreibung	Bsp.
n,0	long	phys. Achs- oder Kanalnummer	2
n,1	long	Kanalzugehörigkeit	1
n,2	string	logischer Achs- oder Kanalname	Y

Items

row, col	Typ	Beschreibung	Bsp.
n,3	long	Achse in Position 0 = nein 1 = ja	1
n,4	long	Referenzpunkt bekannt 0 = nein 1 = ja	1
n,5	long	Maßeinheit 0 = inch 1 = metrisch 2 = Grad 4 = Platz	1
n,6	long	Durchmesserprogrammierung aktiv 0 = nein 1 = ja	0
n,7	double	aktive Spindelsteigungsfehlerkompensation	4.123

NC.Chan.CompOffsetData

Beschreibung Liefert für alle Achsen des Kanals den Summenwert aller Achskompensationen.

Anmerkung In den ersten sieben Spalten werden Zusatzinformationen pro Achse geliefert. Jeweils die achte Spalte enthält den Kompensationswert der Achse.

Item Syntax **NC.Chan.CompOffsetData,(1)**

1. Kanalnummer - Integer Format, gültige Kanalnummer

Aufruf synchron / asynchron

Zugriff Read

zyklisch (opt.) ja

Datenformat **NC.Chan.CompOffsetData, read**

row, col	Typ	Beschreibung	Bsp.
n,0	long	phys. Achs- oder Kanalnummer	2
n,1	long	Kanalzugehörigkeit	1
n,2	string	logischer Achs- oder Kanalname	Y
n,3	long	Achse in Position 0 = nein 1 = ja	1
n,4	long	Referenzpunkt bekannt 0 = nein 1 = ja	1

row, col	Typ	Beschreibung	Bsp.
n,5	long	Maßeinheit 0 = inch 1 = metrisch 2 = Grad 4= Platz	1
n,6	long	Durchmesserprogrammierung aktiv 0 = nein 1 = ja	0
n,7	double	Summenwert aller Achskompensationen	24.089

NC.Chan.CompTempData

Beschreibung	Liefert für alle Achsen des Kanals die aktiven temperaturabhängigen Kompensationswerte.
Anmerkung	In den ersten sieben Spalten werden Zusatzinformationen pro Achse geliefert. Jeweils die achte Spalte enthält den Kompensationswert der Achse.
Item Syntax	NC.Chan.CompTempData,(1) 1. Kanalnummer - Integer Format, gültige Kanalnummer
Aufruf	synchron / asynchron
Zugriff	Read
zyklisch (opt.)	ja
Datenformat	NC.Chan.CompTempData, read

row, col	Typ	Beschreibung	Bsp.
n,0	long	phys. Achs- oder Kanalnummer	2
n,1	long	Kanalzugehörigkeit	1
n,2	string	logischer Achs- oder Kanalname	Y
n,3	long	Achse in Position 0 = nein 1 = ja	1
n,4	long	Referenzpunkt bekannt 0 = nein 1 = ja	1
n,5	long	Maßeinheit 0 = inch 1 = metrisch 2 = Grad 4= Platz	1
n,6	long	Durchmesserprogrammierung aktiv 0 = nein 1 = ja	0
n,7	double	aktiver temperaturabhängiger Kompensationswert	4.589

4.8.4 Abgleichfunktionen

NC.Chan.AuxActual

Beschreibung	Liefert die Liste der aktuellen modalen Hilfsfunktionen.
Item Syntax	NC.Chan.AuxActual,(1) 1. Kanalnummer
Aufruf	synchron / asynchron
Zugriff	Read
zyklisch (opt.)	ja
Datenformat	Jede Hilfsfunktion wird als String in einer eigenen Zeile ausgegeben.

NC.Chan.AuxActual, read

row, col	Typ	Beschreibung	Bsp.
0,0	string	Aktuelle modale Hilfsfunktion 1	M05
1,0	string	Aktuelle modale Hilfsfunktion 2	M08
n,0	string	Aktuelle modale Hilfsfunktion n	M48

NC.Chan.AuxAdjustment

Beschreibung	Liefert die Abgleichsinformation modaler Hilfsfunktionen nach Satzvorlauf.
Anmerkung	Für jede modale Funktionsgruppe wird ein Byte geliefert. Der Wert 0 steht für eine abgegichene Gruppe während Wert 1 für eine nicht abgegichene Gruppe steht.
Item Syntax	NC.Chan.AuxAdjustment,(1) 1. Kanalnummer
Aufruf	synchron / asynchron
Zugriff	Read
zyklisch (opt.)	ja
Datenformat	NC.Chan.AuxAdjustment, read

row, col	Typ	Beschreibung	Bsp.
0,0	long	Abgleichinformation	1

NC.Chan.AuxReference

Beschreibung	Liefert die Liste der Sollzustände modaler Hilfsfunktionen nach Satzvorlauf.
Item Syntax	NC.Chan.AuxReference,(1) 1. Kanalnummer
Aufruf	synchron / asynchron
Zugriff	Read
zyklisch (opt.)	ja
Datenformat	Jede Hilfsfunktion wird als String in einer eigenen Zeile ausgegeben.

NC.Chan.AuxReference, read

row, col	Typ	Beschreibung	Bsp.
0,0	string	Aktuelle modale Hilfsfunktion 1	M05
1,0	string	Aktuelle modale Hilfsfunktion 2	M14
n,0	string	Aktuelle modale Hilfsfunktion n	M40

NC.Chan.FctActual

Beschreibung Liefert die Liste der aktuellen modalen NC-Funktionen.

Item Syntax **NC.Chan.FctActual,(1)**

1. Kanalnummer

Aufruf synchron / asynchron

Zugriff Read

zyklisch (opt.) ja

Datenformat Jede NC-Funktion wird als String in einer eigenen Zeile ausgegeben.

NC.Chan.FctActual, read

row, col	Typ	Beschreibung	Bsp.
0,0	string	Aktuelle NC-Funktion	G80
1,0	string	Aktuelle NC-Funktion	G01
2,0	string	Aktuelle NC-Funktion	G17
3,0	string	Aktuelle NC-Funktion	G91
4,0	string	Aktuelle NC-Funktion	G94
n,0	string	Aktuelle NC-Funktion	OVE

NC.Chan.FctActualSorted

Beschreibung Liefert die Liste von ausgewählten aktuellen modalen NC-Funktionen.

Item Syntax **NC.Chan.FctActualSorted,(1)**

1. Kanalnummer

Aufruf synchron / asynchron

Zugriff Read

zyklisch (opt.) ja

Datenformat Jede NC-Funktion wird als String in einer eigenen Zeile ausgegeben.

NC.Chan.FctActualSorted, read

row, col	Typ	Beschreibung	Bsp.
0,0	string	Aktuelle NC-Funktion	G80
1,0	string	Aktuelle NC-Funktion	G17
2,0	string	Aktuelle NC-Funktion	G91
n,0	string	Aktuelle NC-Funktion	G8

NC.Chan.FctAdjustment

Beschreibung	Liefert die Abgleichsinformationen aller modalen NC-Funktionen nach Satzvorlauf.
Anmerkung	Für jede modale Funktionsgruppe wird ein Byte geliefert. Der Wert "0" steht für eine abgegliche Gruppe, während Wert "1" für eine nicht abgegliche Gruppe steht.
Item Syntax	NC.Chan.FctAdjustment,(1) 1. Kanalnummer
Aufruf	synchron / asynchron
Zugriff	Read
zyklisch (opt.)	ja
Datenformat	NC.Chan.FctAdjustment, read

row, col	Typ	Beschreibung	Bsp.
0,0	long	Aktueller Wert	0
1,0	long	Aktueller Wert	1
2,0	long	Aktueller Wert	1
n,0	long	Aktueller Wert	0

NC.Chan.FctAdjustmentSorted

Beschreibung	Liefert die Abgleichsinformationen ausgewählter modaler NC-Funktionen nach Satzvorlauf.
Anmerkung	Für jede modale Funktionsgruppe wird ein Byte geliefert. Der Wert "0" steht für eine abgegliche Gruppe, während Wert "1" für eine nicht abgegliche Gruppe steht.
Item Syntax	NC.Chan.FctAdjustmentSorted,(1) 1. Kanalnummer
Aufruf	synchron / asynchron
Zugriff	Read
zyklisch (opt.)	ja
Datenformat	NC.Chan.FctAdjustmentSorted, read

row, col	Typ	Beschreibung	Bsp.
0,0	long	Aktueller Wert	1
1,0	long	Aktueller Wert	0
2,0	long	Aktueller Wert	1
n,0	long	Aktueller Wert	0

NC.Chan.FctReference

Beschreibung	Liefert die Liste der Sollzustände modaler NC-Funktionen nach Satzvorlauf.
Item Syntax	NC.Chan.FctReference,(1) 1. Kanalnummer
Aufruf	synchron / asynchron
Zugriff	Read

zyklisch (opt.) ja

Datenformat Jede NC-Funktion wird als String in einer eigenen Zeile ausgegeben.

NC.Chan.FctReference, read

row, col	Typ	Beschreibung	Bsp.
0,0	string	Aktuelle NC-Funktion	G80
1,0	string	Aktuelle NC-Funktion	G00
2,0	string	Aktuelle NC-Funktion	G17
3,0	string	Aktuelle NC-Funktion	G90
4,0	string	Aktuelle NC-Funktion	G94
n,0	string	Aktuelle NC-Funktion	OVE

NC.Chan.FctReferenceSorted

Beschreibung Liefert die Liste der Sollzustände ausgewählter modaler NC-Funktionen nach Satzvorlauf.

Item Syntax **NC.Chan.FctReferenceSorted,(1)**

1. Kanalnummer

Aufruf synchron / asynchron

Zugriff Read

zyklisch (opt.) ja

Datenformat Jede NC-Funktion wird als String in einer eigenen Zeile ausgegeben.

NC.Chan.FctReferenceSorted, read

row, col	Typ	Beschreibung	Bsp.
0,0	string	Aktuelle NC-Funktion	G80
1,0	string	Aktuelle NC-Funktion	G00
2,0	string	Aktuelle NC-Funktion	G17
3,0	string	Aktuelle NC-Funktion	G90
n,0	string	Aktuelle NC-Funktion	OVE

4.8.5 Spindeldaten**NC.Chan.ActSpindleAutoGears**

Beschreibung Zeigt an, für welche Spindel im selektierten Kanal eine automatische Getriebestufenanwahl aktiv ist.

Anmerkung *Zustand der Getriebestufenanwahl:*

- 0 = Getriebestufenanwahl manuell
- 1 = automatische Getriebestufenanwahl aktiv

Die im Kanal angezeigten Spindeln werden über MACODA Parameter 6005 0030 konfiguriert. Dieser Parameter legt kanalspezifisch fest, welche Spindel an welcher Position in der Spindelanzeige dargestellt wird.

Item Syntax **NC.Chan.ActSpindleAutoGears,(1)**

1. Kanalnummer

Aufruf synchron / asynchron

Zugriff Read

Als Zusatzinformationen werden Systemspindelnummer, Spindelname und Achsnummer bereitgestellt.

zyklisch (opt.) ja

Datenformat Zeilenanzahl: abhängig von angezeigten Spindeln, max. 8

NC.Chan.ActSpindleAutoGears, read

row, col	Typ	Beschreibung	Bsp.
n,0	long	Zustand Getriebestufenwahl	1
n,1	long	Systemnummer	2
n,2	string	Spindelname	SSP02
n,3	long	Achsnummer	6

NC.Chan.ActSpindleGears

Beschreibung Liefert die aktive Getriebestufe einer Spindel.

Anmerkung Die im Kanal angezeigten Spindeln werden über MACODA-Parameter 6005 0030 konfiguriert. Dieser Parameter legt kanalspezifisch fest, welche Spindel an welcher Position in der Spindelanzeige dargestellt wird.

Zur Decodierung der Getriebestufen kann der Enumerator "LDX_McoSpindleGearRange" verwendet werden, der wie folgt definiert ist:

Enumerator LDX_McoSpindleGearRange

LDX_McoSpindleGearRange	Beschreibung
LDX_McoSpindleGearRange1_C	Getriebestufe 1
LDX_McoSpindleGearRange2_C	Getriebestufe 2
LDX_McoSpindleGearRange3_C	Getriebestufe 3
LDX_McoSpindleGearRange4_C	Getriebestufe 4
LDX_McoSpindleGearIdleMode_C	Getriebestufe ausgerückt

Item Syntax **NC.Chan.ActSpindleGears,(1)**

1. Kanalnummer

Aufruf synchron / asynchron

Zugriff Read

Als Zusatzinformationen werden Systemspindelnummer, Spindelname und Achsnummer bereitgestellt.

zyklisch (opt.) ja

Datenformat Zeilenanzahl: abhängig von angezeigten Spindeln, max. 8

NC.Chan.ActSpindleGears, read

row, col	Typ	Beschreibung	Bsp.
n,0	long	Aktive Getriebestufe der Spindel	LDX_McoSpindleGearRange1_C
n,1	long	Systemnummer	2
n,2	string	Spindelname	SSP02
n,3	long	Achsnummer	6

NC.Chan.ActSpindleMovements

- Beschreibung** Liefert die Bewegungsfunktionen der Spindeln.
- Anmerkung** Die im Kanal angezeigten Spindeln werden über MACODA-Parameter 6005 0030 konfiguriert. Dieser Parameter legt kanalspezifisch fest, welche Spindel an welcher Position in der Spindelanzeige angezeigt wird.
- Zur Codierung der Getriebestufen wird der Enumerator "LDX_McoSpindleMovements" verwendet, der folgende Werte annehmen kann:

Enumerator LDX_McoSpindleMovements

LDX_McoSpindleMovements	Beschreibung
LDX_McoSpindleNotDefined_C	Spindel nicht definiert
LDX_McoSpindleRight_C	Rechtslauf ohne Kühlmittel
LDX_McoSpindleRightCool_C	Rechtslauf mit Kühlmittel
LDX_McoSpindleLeft_C	Linkslauf ohne Kühlmittel
LDX_McoSpindleLeftCool_C	Linkslauf mit Kühlmittel
LDX_McoSpindleStop_C	Spindel Stopp
LDX_McoSpindleOrientate_C	Spindel richten

- Item Syntax** **NC.Chan.ActSpindleMovements,(1)**
1. Kanalnummer
- Aufruf** synchron / asynchron
- Zugriff** Read
- Als Zusatzinformationen werden Systemspindelnummer, Spindelname und Achsnummer bereitgestellt.
- zyklisch (opt.)** ja
- Datenformat** Zeilenanzahl: abhängig von angezeigten Spindeln, max. 8

NC.Chan.ActSpindleMovements, read

row, col	Typ	Beschreibung	Bsp.
n,0	long	Bewegungsfunktion der Spindel	LDX_McoSpindleStop_C
n,1	long	Systemnummer	2
n,2	string	Spindelname	SSP02
n,3	long	Achsnummer	6

NC.Chan.ActSpindleOverride

- Beschreibung** Liefert den aktuellen Wert des Spindel Overrides.
- Item Syntax** **NC.Chan.ActSpindleOverride,(1)**
1. Kanalnummer
- Aufruf** synchron / asynchron
- Zugriff** Read
- Das Ergebnis enthält den Overridewert in Prozent.
- Außerdem werden der max. einstellbare Overridewert, die Systemnummer, der Spindelname und die Achsnummer bereitgestellt.
- zyklisch (opt.)** ja
- Datenformat** Zeilenanzahl: abhängig von angezeigten Spindeln, max. 8

NC.Chan.ActSpindleOverride, read

row, col	Typ	Beschreibung	Bsp.
n,0	double	aktiver Wert in %	40.0
n,1	double	Maximalwert	50.0
n,2	long	Systemnummer	1
n,3	string	Spindelname	S1 (SSP1)
n,4	long	Achsnummer	6

NC.Chan.ActSpindlePosition

Beschreibung Liefert die aktive Spindelposition in Grad.

Item Syntax **NC.Chan.ActSpindlePosition,(1)**

1. Kanalnummer

Aufruf synchron / asynchron

Zugriff Read

Als Zusatzinformationen werden Systemspindelnummer, Spindelname und Achsnummer bereitgestellt.

zyklisch (opt.) ja

Datenformat Zeilenanzahl: abhängig von angezeigten Spindeln, max. 8

NC.Chan.ActSpindlePosition, read

row, col	Typ	Beschreibung	Bsp.
n,0	double	aktive Ist-Position der Spindel	246.3700
n,1	long	Systemnummer	1
n,2	string	Spindelname	S1
n,3	long	Achsnummer	6

NC.Chan.ActSpindlePower

Beschreibung Liefert die aktive und maximale Spindelleistung in Watt.

Item Syntax **NC.Chan.ActSpindlePower,(1)**

1. Kanalnummer

Aufruf synchron / asynchron

Zugriff Read

Als Zusatzinformationen werden Systemspindelnummer, Spindelname und Achsnummer bereitgestellt.

zyklisch (opt.) ja

Datenformat Zeilenanzahl: abhängig von angezeigten Spindeln, max. 8

NC.Chan.ActSpindlePower, read

row, col	Typ	Beschreibung	Bsp.
n,0	long	aktiver Wert in Watt	80
n,1	long	Maximalwert in Watt	120
n,2	long	Systemnummer	1

row, col	Typ	Beschreibung	Bsp.
n,3	string	Spindelname	S1
n,4	long	Achsnummer	6

NC.Chan.ActSpindleSpeed

Beschreibung Liefert die aktive(n) Spindelsolldrehzahl(en).

Item Syntax **NC.Chan.ActSpindleSpeed,(1)**

1. Kanalnummer

Aufruf synchron / asynchron

Zugriff Read

Als Zusatzinformationen werden Systemspindelnummer, Spindelname und Achsnummer bereitgestellt. Bei aktivem "G196" werden die Schnittgeschwindigkeiten in m/min bzw. feet/min geliefert.

zyklisch (opt.) ja

Datenformat Zeilenanzahl: abhängig von angezeigten Spindeln, max. 8

NC.Chan.ActSpindleSpeed, read

row, col	Typ	Beschreibung	Bsp.
n,0	double	Spindelsolldrehzahl	450.0
n,1	long	Systemnummer	1
n,2	string	Spindelname	S1 (SSP1)
n,3	long	Achsnummer	6

NC.Chan.ActSpindleSpeedDrv

Beschreibung Liefert die aktive(n) Spindelstdrehzahl(en).

Item Syntax **NC.Chan.ActSpindleSpeedDrv,(1)**

1. Kanalnummer

Aufruf synchron / asynchron

Zugriff Read

Als Zusatzinformationen werden Systemspindelnummer, Spindelname und Achsnummer bereitgestellt. Bei aktivem "G196" werden die Schnittgeschwindigkeiten in m/min bzw. feet/min geliefert.

zyklisch (opt.) ja

Datenformat Zeilenanzahl: abhängig von angezeigten Spindeln, max. 8

NC.Chan.ActSpindleSpeedDrv, read

row, col	Typ	Beschreibung	Bsp.
n,0	double	Spindelstdrehzahl	2000.0
n,1	long	Systemnummer	1
n,2	string	Spindelname	S1 (SSP1)
n,3	long	Achsnummer	6

NC.Chan.ActSpindleSwitchGears

Beschreibung	Zeigt an, ob zur Zeit ein Getriebestufenwechsel erfolgt.
Anmerkung	<i>Zustand der Getriebestufenwechsels:</i> 0 = kein Getriebestufenwechsel 1 = Getriebestufenwechsel erfolgt Die im Kanal angezeigten Spindeln werden über MACODA-Parameter 6005 0030 konfiguriert. Dieser Parameter legt kanalspezifisch fest, welche Spindel an welcher Position in der Spindelanzeige angezeigt wird.
Item Syntax	NC.Chan.ActSpindleSwitchGears,(1) 1. Kanalnummer
Aufruf	synchron / asynchron
Zugriff	Read
	Als Zusatzinformationen werden Systemspindelnummer, Spindelname und Achsnummer bereitgestellt.
zyklisch (opt.)	ja
Datenformat	Zeilenanzahl: abhängig von angezeigten Spindeln, max. 8

NC.Chan.ActSpindleSwitchGears, read

row, col	Typ	Beschreibung	Bsp.
n,0	long	Getriebestufenwechsel	0
n,1	long	Systemnummer	2
n,2	string	Spindelname	SSP02
n,3	long	Achsnummer	6

NC.Chan.ActSpindleTorque

Beschreibung	Liefert das aktive und maximale Spindeldrehmoment als Prozentwert.
Item Syntax	NC.Chan.ActSpindleTorque,(1) 1. Kanalnummer
Aufruf	synchron / asynchron
Zugriff	Read
	Als Zusatzinformationen werden Systemspindelnummer, Spindelname und Achsnummer bereitgestellt.
zyklisch (opt.)	ja
Datenformat	Zeilenanzahl: abhängig von angezeigten Spindeln, max. 8

NC.Chan.ActSpindleTorque, read

row, col	Typ	Beschreibung	Bsp.
n,0	double	aktiver Wert in %	5.5
n,1	double	Maximalwert in %	8.3
n,2	long	Systemnummer	1
n,3	string	Spindelname	S1
n,4	long	Achsnummer	6

NC.Chan.ProgrammedSpindleSpeed

Beschreibung	Liefert die programmierte(n) Spindeldrehzahl(en).
Anmerkung	Die im Kanal angezeigten Spindeln werden über MACODA Parameter 6005 0030 konfiguriert. Dieser Parameter legt kanalspezifisch fest, welche Spindel an welcher Position in der Spindelanzeige angezeigt wird.
Item Syntax	NC.Chan.ProgrammedSpindleSpeed,(1) 1. Kanalnummer
Aufruf	synchron / asynchron
Zugriff	Read Als Zusatzinformationen werden Systemspindelnummer, Spindelname und Achsnummer bereitgestellt.
zyklisch (opt.)	ja
Datenformat	Zeilenanzahl: abhängig von angezeigten Spindeln, max. 8

NC.Chan.ProgrammedSpindleSpeed, read

row, col	Typ	Beschreibung	Bsp.
n,0	double	programmierte Spindeldrehzahl	2000.0
n,1	long	Systemnummer	1
n,2	string	Spindelname	S1 (SSP1)
n,3	long	Achsnummer	6

4.8.6 Prozessdaten

NC.Chan.ActCallChain

Beschreibung	Liefert die aktuelle Unterprogrammschachtelung.
Anmerkung	Es werden jeweils der Programmname mit Pfad und die Satznummer geliefert. Die Unterprogrammschachtelung wird relativ zum aktiven Satz geliefert. Die Unterprogrammschachtelung relativ zum vorbereiteten Satz steht zzt. nicht zur Verfügung.
Item Syntax	NC.Chan.ActCallChain,(1) 1. Kanalnummer
Aufruf	synchron / asynchron
Zugriff	Read
zyklisch (opt.)	ja
Datenformat	Zeilenanzahl: abhängig von Anzahl der Schachtelung

NC.Chan.ActCallChain, read

row, col	Typ	Beschreibung	Bsp.
n,0	string	Programmname mit Pfad	/mnt/Testprogramm.cnc
n,1	string	Satznummer	N235

NC.Chan.ActMainProgram

Beschreibung	Liefert den Namen und den Pfad des Hauptprogramms.
---------------------	--

Item Syntax NC.Chan.ActMainProgram,(1)
 1. Kanalnummer
Aufruf synchron / asynchron
Zugriff Read
zyklisch (opt.) ja
Datenformat NC.Chan.ActMainProgram, read

row, col	Typ	Beschreibung	Bsp.
0,0	string	Name & Pfad des Hauptprogramms	/usr/user/sekt.cnc

NC.Chan.ActNcBlock

Beschreibung Liefert die Bezeichnung des aktuellen Satzes.
Anmerkung Falls keine Satznummer programmiert ist, wird ein Leerstring zurückgeliefert.
Item Syntax NC.Chan.ActNcBlock,(1)
 1. Kanalnummer
Aufruf synchron / asynchron
Zugriff Read
zyklisch (opt.) ja
Datenformat NC.Chan.ActNcBlock, read

row, col	Typ	Beschreibung	Bsp.
0,0	string	Bezeichnung des aktuellen Satzes	N150

NC.Chan.ActSubProgram

Beschreibung Liefert den Namen des aktuellen Unterprogramms.
Anmerkung Das Ergebnis enthält den Namen und den Pfad des Unterprogramms.
 Bei lesegeschützten bzw. verschlüsselten Unterprogrammen und Zyklen wird der Name des aufrufenden Programms geliefert.
Item Syntax NC.Chan.ActSubProgram,(1)
 1. Kanalnummer
Aufruf synchron / asynchron
Zugriff Read
zyklisch (opt.) ja
Datenformat NC.Chan.ActSubProgram, read

row, col	Typ	Beschreibung	Bsp.
0,0	string	Name & Pfad des Unterprogramms	/usr/user/sub01.cnc

NC.Chan.AxisMirror

Beschreibung Mit dem Item kann der aktuelle Zustand des Spiegels ermittelt werden. Ist das Spiegeln für eine Achse innerhalb des Kanals eingeschaltet, liefert das

Item für diese Achse den Wert 1 zurück. Bei ausgeschaltetem Spiegeln wird die 0 zurückgegeben. Neben den Spiegelzuständen liefert das Item 7 weitere Zusatzwerte, die in den Spalten 0 - 6 dargestellt werden.

Anmerkung In den ersten sieben Spalten werden Zusatzinformationen pro Achse geliefert. Die achte Spalte enthält den aktuellen Zustandswert "Spiegeln aktiv" der Achse.

Item Syntax **NC.Chan.AxisMirror,(1)**

1. Kanalnummer

Aufruf synchron / asynchron

Zugriff Read

zyklisch (opt.) ja

Datenformat **NC.Chan.AxisMirror, read**

row, col	Typ	Beschreibung	Bsp.
0,0	long	phys. Achs- oder Kanalnummer	2
0,1	long	Kanalzugehörigkeit	1
0,2	string	logischer Achs- oder Kanalname	Y
0,3	long	Achse in Position 0 = nein 1 = ja	1
0,4	long	Referenzpunkt bekannt 0 = nein 1 = ja	1
0,5	long	Maßeinheit 0 = inch 1 = metrisch 2 = Grad 4 = Platz	1
0,6	long	Durchmesserprogrammierung aktiv 0 = nein 1 = ja	0
0,7	double	Spiegelzustandswert 0 = ausgeschaltet 1 = eingeschaltet	1

NC.Chan.AxisPoleSet

Beschreibung Das Item liefert die absolute Position des Pols für die Eingabehilfen Spiegeln, Skalieren und Drehen. Neben dem Spiegeldrehpunkt liefert das Item 7 weitere Zusatzwerte, die in den Spalten 0 - 6 dargestellt werden.

Anmerkung In den ersten sieben Spalten werden Zusatzinformationen pro Achse geliefert. Jeweils die achte Spalte enthält die absolute Position des Pols einer Achse.

Item Syntax **NC.Chan.AxisPoleSet,(1)**

1. Kanalnummer

Aufruf synchron / asynchron
Zugriff Read
zyklisch (opt.) ja
Datenformat NC.Chan.AxisPoleSet, read

row, col	Typ	Beschreibung	Bsp.
n,0	long	phys. Achs- oder Kanalnummer	2
n,1	long	Kanalzugehörigkeit	1
n,2	string	logischer Achs- oder Kanalname	X
n,3	long	Achse in Position 0 = nein 1 = ja	1
n,4	long	Referenzpunkt bekannt 0 = nein 1 = ja	1
n,5	long	Maßeinheit 0 = inch 1 = metrisch 2 = Grad 4 = Platz	1
n,6	long	Durchmesserprogrammierung aktiv 0 = nein 1 = ja	0
n,7	double	absoluter Positionswert des Pols	12

NC.Chan.AxisRotate

Beschreibung Das Item liefert den Winkel, um den eine programmierte Kontur in der aktiven Ebene gedreht wurde.
Anmerkung
Item Syntax NC.Chan.AxisRotate,(1)
 1. Kanalnummer - Integer Format, gültige Kanalnummer
Aufruf synchron / asynchron
Zugriff Read
zyklisch (opt.) ja
Datenformat NC.Chan.AxisRotate, read

row, col	Typ	Beschreibung	Bsp.
0,0	double	Winkelwert	5.0

NC.Chan.AxisScale

Beschreibung Mit dem Item können die Skalierungswerte gelesen werden, mit denen eine programmierte Kontur beaufschlagt wurde. Neben den Skalierungswerten liefert das Item 7 weitere Zusatzwerte, die in den Spalten 0 - 6 dargestellt werden.

Anmerkung	In den ersten sieben Spalten werden Zusatzinformationen pro Achse geliefert. Jeweils die achte Spalte enthält den Skalierungswert einer Achse.
Item Syntax	NC.Chan.AxisScale,(1) 1. Kanalnummer - Integer Format, gültige Kanalnummer
Aufruf	synchron / asynchron
Zugriff	Read
zyklisch (opt.)	ja
Datenformat	NC.Chan.AxisScale, read

row, col	Typ	Beschreibung	Bsp.
n,0	long	phys. Achs- oder Kanalnummer	2
n,1	long	Kanalzugehörigkeit	1
n,2	string	logischer Achs- oder Kanalname	X
n,3	long	Achse in Position 0 = nein 1 = ja	1
n,4	long	Referenzpunkt bekannt 0 = nein 1 = ja	1
n,5	long	Maßeinheit 0 = inch 1 = metrisch 2 = Grad 4 = Platz	1
n,6	long	Durchmesserprogrammierung aktiv 0 = nein 1 = ja	0
n,7	double	Skalierungswert	1.2

NC.Chan.EqualizeState

Beschreibung	Das Item liefert den Status des Satzvorlaufabgleichs. Nach der Initialisierung wird im Kern ein Eventmechanismus aktiviert, der anspricht, sobald eine Zustandsänderung des Satzvorlaufabgleichs erfolgt.
Anmerkung	Gültige Daten erscheinen erst, nachdem die Steuerung den ersten Event abgesetzt hat. Vor dem ersten Event sind die gelieferten Daten ungültig und das Quality-Flag liefert solange "Uncertain". Die erste Datenzeile enthält die Versionsnummer des Events. Sie wird mit jedem stattgefundenen Event um 1 erhöht. Die zweite Datenzeile enthält eine Bitmaske, in der die entsprechenden Zustände des Abgleichs markiert sind. Bit 1 zeigt den Status des Satzvorlaufabgleichs. Ist dieses Bit gesetzt, so findet der Abgleich statt.
Item Syntax	NC.Chan.EqualizeState,(1) 1. Kanalnummer - Integer Format, gültige Kanalnummer
Aufruf	synchron / asynchron
Zugriff	Read

zyklisch (opt.) ja
 Datenformat NC.Chan.EqualizeState, read

row, col	Typ	Beschreibung	Bsp.
0,0	ulong	Versionsnummer des Events	5
1,0	ulong	Bitmaske modaler Blockstatus Bit 0 = zeigt, ob der im Interpolator befindliche Block durch den Satzvorlauf erzeugt wurde Bit 1 = zeigt, ob ein Abgleich stattfindet	3

NC.Chan.MessageWindow

Beschreibung Liefert die aktuellen programmierten Hinweise.

Anmerkung Es werden kanalspezifische und kanalunabhängige Hinweise berücksichtigt, ebenso CPL-Ausgaben.
 Die Fenstergröße, d. h. die Anzahl der gelieferten Hinweise, kann vorgegeben werden, sie ist jedoch durch NC-interne Ressourcen beschränkt (max. 5 kanalspezifische und 5 kanalunabhängige Hinweise).



Einschränkung: Parameter Fenstergröße ist zzt. wirkungslos!

Item Syntax NC.Chan.MessageWindow,(1),(2)

1. Kanalnummer
2. Fenstergröße

Aufruf synchron / asynchron

Zugriff Read

Es werden geliefert:

- kanalunabhängige Hinweise (GMSG ...)
- kanalspezifische Hinweise (MSG ...)
- CPL-Ausgaben PRINT#(0, ...)

zyklisch (opt.) ja

Datenformat Zeilenanzahl: abhängig von Anzahl der Hinweise

NC.Chan.MessageWindow, read

row, col	Typ	Beschreibung	Bsp.: NC.Chan.MessageWindow,1,20
n,0	string	Hinweis	Gewinde wird gebohrt

NC.Chan.ModalAuxFunctions

Beschreibung Liefert eine Liste der modalen Hilfsfunktionen. Die Reihenfolge und Anzahl der gelieferten Funktionen ist über MACODA (6005 00050) konfigurierbar.

Anmerkung Die Liste kann Leerstrings enthalten, falls in einer modalen Gruppe keine Hilfsfunktion aktiv ist.

Bei fehlerhaften Eingaben in der Konfiguration (MACODA Parameter hat ungültigen Wert) werden Einträge als ungültig gekennzeichnet. Der Listeneintrag hat dann den Wert "?".

Item Syntax NC.Chan.ModalAuxFunctions,(1)

1. Kanalnummer

Aufruf synchron / asynchron
Zugriff Read
zyklisch (opt.) ja
Datenformat Zeilenanzahl: abhängig von MACODA

NC.Chan.ModalAuxFunctions, read

row, col	Typ	Beschreibung	Bsp.
n,0	string	aktive NC-Funktion	M10

NC.Chan.ModalAxisOffsetFunctions

Beschreibung Liefert eine Liste der modalen NC-Funktionen für Achsnullpunktverschiebung. Die Reihenfolge und Anzahl der gelieferten modalen Funktionen ist über MACODA (6005 00070) konfigurierbar.

Anmerkung Die Liste kann Leerstrings enthalten, falls in einer modalen Gruppe keine NC-Funktion aktiv ist.

Bei fehlerhaften Eingaben in der Konfiguration (MACODA Parameter hat ungültigen Wert) werden Einträge als ungültig gekennzeichnet. Der Listeneintrag hat dann den Wert "?".

Item Syntax NC.Chan.ModalAxisOffsetFunctions,(1)

1. Kanalnummer

Aufruf synchron / asynchron

Zugriff Read

zyklisch (opt.) ja

Datenformat Zeilenanzahl: abhängig von MACODA

NC.Chan.ModalAxisOffsetFunctions, read

row, col	Typ	Beschreibung	Bsp.
n,0	string	aktive NC-Funktion	G55.3

NC.Chan.ModalFunction

Beschreibung Liest eine einzelne modale Gruppe aus.

Die auszulesende modale Gruppe wird spezifiziert durch eine beliebige modale Funktion, die in dieser Gruppe wirksam sein kann.

Item Syntax NC.Chan.ModalFunction,(1),(2)

1. Kanalnummer

2. modale Funktion

Aufruf asynchron

Zugriff Read

zyklisch (opt.) nein

Datenformat NC.Chan.ModalFunction, read

row, col	Typ	Beschreibung	NC.Chan.ModalFunction,2,G70
0,0	string	Ergebnis einer modalen Gruppe	G71

NC.Chan.ModalNcFunctions

Beschreibung	Liefert eine Liste der modalen NC-Funktionen. Die Reihenfolge und Anzahl der gelieferten modalen Funktionen ist über MACODA (6005 00040) konfigurierbar.
Anmerkung	Die Liste kann Leerstrings enthalten, falls in einer modalen Gruppe keine NC-Funktion aktiv ist. Bei fehlerhaften Eingaben in der Konfiguration (MACODA Parameter hat ungültigen Wert) werden Einträge als ungültig gekennzeichnet. Der Listeneintrag hat dann den Wert "?".
Item Syntax	NC.Chan.ModalNcFunctions,(1) 1. Kanalnummer
Aufruf	synchron / asynchron
Zugriff	Read
zyklisch (opt.)	ja
Datenformat	Zeilenanzahl: abhängig von MACODA

NC.Chan.ModalNcFunctions, read

row, col	Typ	Beschreibung	Bsp.
n,0	string	aktive NC-Funktion	G18

NC.Chan.ModalTransformationFunctions

Beschreibung	Liefert eine Liste der modalen NC-Funktionen für Werkstücktransformationen. Die Reihenfolge und Anzahl der gelieferten modalen Funktionen ist über MACODA (6005 00060) konfigurierbar.
Anmerkung	Die Liste kann Leerstrings enthalten, falls in einer modalen Gruppe keine NC-Funktion aktiv ist. Bei fehlerhaften Eingaben in der Konfiguration (MACODA Parameter hat ungültigen Wert) werden Einträge als ungültig gekennzeichnet. Der Listeneintrag hat dann den Wert "?".
Item Syntax	NC.Chan.ModalTransformationFunctions,(1) 1. Kanalnummer
Aufruf	synchron / asynchron
Zugriff	Read
zyklisch (opt.)	ja
Datenformat	Zeilenanzahl: abhängig von MACODA

NC.Chan.ModalTransformationFunctions, read

row, col	Typ	Beschreibung	Bsp.
n,0	string	aktive NC-Funktion	G154.3

NC.Chan.ModeOfOperation

Beschreibung	Liefert die Betriebsart und die Nebenbetriebsart eines Kanals.
Anmerkung	Für die Codierung der genannten Betriebsarten werden die Enumeratoren "LDX_SavOpModes" und "LDX_SavOpSubModes" verwendet. Diese können folgende Werte annehmen:

Enumerator LDX_SavOpModes

LDX_SavOpModes	Beschreibung
LDX_SAV_OP_MODE_EMPTY_C	keine Betriebsart aktiv
LDX_SAV_OP_MODE_JOGGING_C	Tippbetrieb
LDX_SAV_OP_MODE_REFERENCE_C	Referenzpunkt Anfahren
LDX_SAV_OP_MODE_SEARCH-BLOCK_C	Satzvorlauf
LDX_SAV_OP_MODE_MDI_C	Handeingabe
LDX_SAV_OP_MODE_AUTOMATIC_C	Automatik: Teileprogramme können in Folgesatz abgearbeitet werden
LDX_SAV_OP_MODE_SINGLE_BLOCK_C	Programmsatz: Mit "NC-Start" werden einzelne Sätze wie sie im Teileprogramm stehen abgearbeitet. Die Satzvorbereitung startet erst mit "NC-Start".
LDX_SAV_OP_MODE_SINGLE_STEP_C	Einzelschritt: Mit "NC-Start" wird ein einzelner Block an den Interpolator zur Bearbeitung weitergereicht.
LDX_SAV_OP_MODE_TEACH_C	reserviert
LDX_SAV_OP_MODE_SUSPEND_C	reserviert
LDX_SAV_OP_MODE_SINGLE_ACT_BLOCK_C	Einzelsatz: Mit "NC-Start" werden alle Blöcke, die aufgrund eines einzelnen NC-Satzes im Teileprogramm generiert und fertig vorbereitet wurden, an den Interpolator zur Bearbeitung weitergereicht.
LDX_SAV_OP_MODE_RETURN_TO_PATH_C	Wiederanfahren: Achsen können manuell von der Kontur weggefahren und automatisch oder manuell wieder herangefahren werden.
LDX_SAV_OP_MODE_DEBUG_STEP_C	CPL-Debugger: Es werden einzelne Sätze wie sie im Teileprogramm stehen abgearbeitet.
LDX_SAV_OP_MODE_DEBUG_CONT_C	CPL-Debugger: Es werden alle Sätze bis zum nächsten Breakpoint abgearbeitet.
LDX_SAV_OP_MODE_JOGGING_COORDS_C	Einrichten Manuell: Bewegung in Werkstückkoordinaten

Enumerator LDX_SavOpSubModes

LDX_SavOpSubModes	Beschreibung
LDX_SAV_OP_SUBMODE_EMPTY_C	keine Nebenbetriebsart aktiv
LDX_SAV_OP_SUBMODE_OVERSTORE_C	Überspeichern aktiv. D. h. in den Automatikbetriebsarten können Verfahrssätze übergeben werden.

Item Syntax NC.Chan.ModeOfOperation,(1)

1. Kanalnummer

Aufruf synchron / asynchron

Zugriff Read

zyklisch (opt.) ja

Datenformat NC.Chan.ModeOfOperation, read

row, col	Typ	Beschreibung	Bsp.: NC.Chan.ModeOfOperation,1
0,0	long	Aktive Betriebsart im Kanal 1	LDX_SAV_OP_MODE_AUTOMATIC_C
0,1	long	Aktive Nebenbetriebsart im Kanal 1	LDX_SAV_OP_SUBMODE_OVERSTORE_C

NC.Chan.OptionalStop

Beschreibung Liefert den aktuellen Status von "Wahlweise Halt", so, wie es am NC-Interface-Eingang und am NC-Interface-Ausgang ansteht.

Item Syntax NC.Chan.OptionalStop,(1)

1. Kanalnummer

Aufruf synchron / asynchron

Zugriff Read

zyklisch (opt.) ja

Datenformat NC.Chan.OptionalStop, read

row, col	Typ	Beschreibung	Bsp.: NC.Chan.OptionalStop,1
0,0	long	Status von "Wahlweise Halt" am Eingang	0
0,1	long	Status von "Wahlweise Halt" am Ausgang	0

NC.Chan.ProgramPosition

Beschreibung Liefert die aktuelle File-Position im jeweiligen Haupt- oder Unterprogramm.

Anmerkung Falls sich der Kanal im Fehlerzustand befindet, wird die File-Position geliefert, an der der Fehler aufgetreten ist.



Dieses Item ist nicht für die zyklische Abfrage innerhalb einer Gruppe geeignet, da bei jedem Lesezugriff mit der NC kommuniziert wird.

Item Syntax NC.Chan.ProgramPosition,(1)

1. Kanalnummer

Aufruf	synchron / asynchron
Zugriff	Read
	Es wird der Dateiname und komplette Pfad des Programms, sowie die File-Position geliefert.
zyklisch (opt.)	nein
Datenformat	NC.Chan.ProgramPosition, read

row, col	Typ	Beschreibung	Bsp.
0,0	string	Dateiname inkl. Pfad	/usr/user/sekt.cnc
0,1	long	File-Position	34

NC.Chan.ProgramWindow

Beschreibung	Liefert einen Ausschnitt aus dem aktuellen Haupt- oder Unterprogramm.
Anmerkung	Die Fenstergröße des Programmausschnitts kann parametrierbar werden, außerdem die Anzahl der NC-Sätze vor dem aktiven Satz. Bei lesegeschützten bzw. verschlüsselten Unterprogrammen und Zyklen wird der Ausschnitt aus dem aufrufenden Programm geliefert.



Einschränkung: Zzt. wird noch nicht die Information geliefert, welches der letzte vorbereitete Satz ist. Das Feld wird in den Antwortdaten vorgesehen, enthält aber noch keinen gültigen Wert.

Item Syntax **NC.Chan.ProgramWindow,(1),(2),(3)**

1. Kanalnummer
2. Fenstergröße (Anzahl der Sätze im Ergebnis, max. 200)
3. gewünschte Anzahl der vorlaufenden Sätze (max. 40)

Aufruf synchron / asynchron

Zugriff Read

Programmausschnitt (pro Zeile ein BSTR)

Außerdem werden geliefert:

- die gewünschte Anzahl vorlaufender Sätze
- tatsächliche Anzahl vorlaufender Sätze
- Referenz des letzten vorbereiteten Satzes

zyklisch (opt.) ja

Datenformat Die Zeilenanzahl ist abhängig von Anzahl der Sätze im Ergebnis. Nach Zeile 0 noch max. 200 Zeilen

NC.Chan.ProgramWindow, read

row, col	Typ	Beschreibung	Bsp.
0,0	long	gewünschte Anzahl vorlaufender Sätze	10
0,1	long	tatsächliche Anzahl vorlaufender Sätze	0

row, col	Typ	Beschreibung	Bsp.
0,2	long	Referenz des letzten vorbereiteten Satzes	30
n,0	string	Zeile n des Programmausschnitts	N030 X30 Y30 Z30

NC.Chan.ReentryChain

Beschreibung Mit dem Item kann die in einem Kanal gespeicherte "call-chain" gelesen, bzw. neu gesetzt werden. Liefert das Item beim Lesen einen Nullstring zurück, dann existiert in dem ausgewählten Kanal keine "call-chain".

Anmerkung

Item Syntax **NC.Chan.ReentryChain,(1)**

1. Kanalnummer - Integer Format, gültige Kanalnummer

Aufruf synchron / asynchron

Zugriff Read / Write

zyklisch (opt.) nein

Datenformat **NC.Chan.ReentryChain, read**

row, col	Typ	Beschreibung	Bsp.
0,0	string	gespeicherte Call-Chain	

NC.Chan.ReentryDelete

Beschreibung Nach dem Aufruf des Items wird die intern in der Steuerung gespeicherte "call-chain" für den angegebenen Kanal gelöscht.

Anmerkung

Item Syntax **NC.Chan.ReentryDelete,(1)**

1. Kanalnummer - Integer Format, gültige Kanalnummer

Aufruf synchron / asynchron

Zugriff Call

zyklisch (opt.) nein

NC.Chan.ReentryEvent

Beschreibung Mit dem Item wird in der NC ein Event angemeldet, das jedes Mal dann kommt, wenn innerhalb des laufenden NC-Programms die Befehlssequenz RCB(1) bzw. RCB(0) erreicht wird. Beim Erkennen des Events liefert das Item in Zeile 1 die Versionsnummer des Events. Diese Nummer ist ein laufender Wert, der mit jedem Event im NC-Kern um 1 inkrementiert wird. Die Zeile 2 meldet welche Befehlssequenz das Event ausgelöst hat. Mit diesen Befehlen markiert der Programmierer Stellen im NC-Programm, an denen ein unkomplizierter Wiedereintritt in das Programm nach einem vorherigen Abbruch möglich ist.

Anmerkung Der Befehl RCB(1) kennzeichnet den Anfang und der Befehl RCB(0) das Ende des Wiedereintrittsbereichs.

Item Syntax **NC.Chan.ReentryEvent,(1)**

1. Kanalnummer - Integer Format, gültige Kanalnummer

Aufruf synchron / asynchron

Zugriff Read

zyklisch (opt.) ja, erforderlich!

Datenformat NC.Chan.ReentryEvent, read

row, col	Typ	Beschreibung	Bsp.
0,0	long	Versionskennung	4
1,0	bool	Befehlssequenz RCB(0) = FALSE RCB(1) = TRUE	TRUE

NC.Chan.ReturnToContour

Beschreibung Liefert den aktuellen Signalzustand von "Wiederanfahren an Kontur", so wie es am NC-Interface-Eingang und am NC-Interface-Ausgang ansteht.

Anmerkung *Der Signalstatus meldet die aktive Variante des Wiederanfahrens:*

- 0 = direktes Wiederanfahren
- 1 = Wiederanfahren über Pfad

Item Syntax NC.Chan.ReturnToContour,(1)

1. Kanalnummer

Aufruf synchron / asynchron

Zugriff Read

zyklisch (opt.) ja

Datenformat NC.Chan.ReturnToContour, read

row, col	Typ	Beschreibung	Bsp.: NC.Chan.ReturnToContour,1
0,0	long	Status von "Wiederanfahren an Kontur" am NC-Eingang	0
0,1	long	Status von "Wiederanfahren an Kontur" am NC-Ausgang	0

NC.Chan.ReturnToPathStrategy

Beschreibung *Liefert Informationen zur aktuellen Wiederanfahrstrategie:*

- Wiederanfahrbetriebsart
- Wiederanfahrpunkt
- Zustand, ob Wegfahrbedingungen aufgezeichnet werden

Anmerkung Die Wiederanfahrbetriebsart wird mit Hilfe des Enumerators "LDX_McoRetPath_Mode" codiert, der folgende Werte annehmen kann:

Enumerator LDX_McoRetPath_Mode

LDX_McoRetPath_Mode	Beschreibung
LDX_McoRetPathAutomatic_C	automatisches Wiederanfahren
LDX_McoRetPathSingleBlock_C	Wiederanfahren mit Einzelsatz
LDX_McoRetPathManual_C	manuelles Wiederanfahren

Der Wiederanfahrpunkt wird mit Hilfe des Enumerators "LDX_McoRetPath_Point" codiert, der folgende Werte annehmen kann:

Enumerator LDX_McoRetPath_Point

LDX_McoRetPath_Point	Beschreibung
LDX_McoRetPathStartOfBlock_C	Wiederauffahren zum Startpunkt
LDX_McoRetPathEndOfBlock_C	Wiederauffahren zum Endpunkt
LDX_McoRetPathPointOfInterruption_C	Wiederauffahren zum Unterbrechungspunkt

Werden Wegfahrbedingungen aufgezeichnet, liefert die Funktion als dritten Wert "1" ansonsten "0".

Item Syntax NC.Chan.ReturnToPathStrategy,(1)

1. Kanalnummer

Aufruf synchron / asynchron

Zugriff Read

zyklisch (opt.) ja

Datenformat NC.Chan.ReturnToPathStrategy, read

row, col	Typ	Beschreibung	Bsp.
0,0	long	Wiederauffahrtbetriebsart	LDX_McoRetPathAutomatic_C
0,1	long	Wiederauffahrtspunkt	LDX_McoRetPathStartOfBlock_C
0,2	long	Aufzeichnung der Wegfahrbedingungen	0

NC.Chan.SkipBlockMode

Beschreibung Liefert die Zustände des NC-Eingangs-Signals "Satz überlesen" und des NC-Ausgangs-Signals "Satz überlesen aktivieren".

Item Syntax NC.Chan.SkipBlockMode,(1)

1. Kanalnummer

Aufruf synchron / asynchron

Zugriff Read

zyklisch (opt.) ja

Datenformat NC.Chan.SkipBlockMode, read

row, col	Typ	Beschreibung	Bsp.: NC.Chan.SkipBlockMode,1
0,0	long	Status vom NC-Eingangssignal	0
0,1	long	Status vom NC-Ausgangssignal	0

NC.Chan.UpCallFromNcBlock

Beschreibung Ermittelt Unterprogrammaufrufe in einem vorgegebenen NC-Satz.

Item Syntax NC.Chan.UpCallFromNcBlock,(1),(2),(3)

1. Kanalnummer

2. Pfad des Hauptprogramms (ohne abschließenden Schrägstrich)

3. NC-Satz

Aufruf synchron / asynchron

Zugriff Read

Enthält der Satz einen Unterprogrammaufruf, so werden der Unterprogramm-Name mit Pfad und die String-Länge geliefert. Ist dies nicht der Fall, wird ein Leerstring geliefert.

zyklisch (opt.)

nein

Datenformat

NC.Chan.UpCallFromNcBlock, read

row, col	Typ	Beschreibung	Bsp.: NC.Chan.UpCallFromNcBlock,1,/usr/user,"10 CALL /usrfep/subprogram.cnc"
0,0	string	Unterprogramm-Name mit Pfad	/usr/user/subprogram.cnc
0,1	long	String-Länge	24

4.8.7 Vorschub- und Drehzahlwerte

NC.Chan.ActAxisVelocity

Beschreibung

Liefert die aktuelle Achsgeschwindigkeit.

Anmerkung

In den ersten sieben Spalten werden Zusatzinformationen pro Achse geliefert. Die achte Spalte enthält die Geschwindigkeit der Achse.

Item Syntax

NC.Chan.ActAxisVelocity,(1)

1. Kanalnummer

Aufruf

synchron / asynchron

Zugriff

Read

zyklisch (opt.)

ja

Datenformat

NC.Chan.ActAxisVelocity, read

row, col	Typ	Beschreibung	Bsp.
n,0	long	phys. Achs- oder Kanalnummer	2
n,1	long	Kanalzugehörigkeit	1
n,2	string	logischer Achs- oder Kanalname	Y
n,3	long	Achse in Position 0 = FALSE 1 = TRUE	1
n,4	long	Referenzpunkt bekannt	1
n,5	long	Maßeinheit	LDX_McoUnitMetric_C
n,6	long	Durchmesserprogrammierung aktiv	0
n,7	double	Achsgeschwindigkeit	1450.000

NC.Chan.ActAxisVelocityMax

Beschreibung

Liefert die Maximalgeschwindigkeit im Achskoordinatensystem.

Anmerkung

Es werden nur Werte für asynchrone Achsen und Achsen im Tippbetrieb geliefert. Ansonsten ist dieser Wert Null.

In den ersten sieben Spalten werden Zusatzinformationen pro Achse geliefert. Die achte Spalte enthält den jeweiligen Geschwindigkeitswert der Achse.

Item Syntax

NC.Chan.ActAxisVelocityMax,(1)

1. Kanalnummer

Aufruf synchron / asynchron
Zugriff Read
zyklisch (opt.) ja
Datenformat NC.Chan.ActAxisVelocityMax, read

row, col	Typ	Beschreibung	Bsp.
n,0	long	phys. Achs- oder Kanalnummer	2
n,1	long	Kanalzugehörigkeit	1
n,2	string	logischer Achs- oder Kanalname	Z
n,3	long	Achse in Position 0 = FALSE 1 = TRUE	1
n,4	long	Referenzpunkt bekannt	1
n,5	long	Maßeinheit	LDX_McoUnitMetric_C
n,6	long	Durchmesserprogrammierung aktiv	0
n,7	double	maximale Achsgeschwindigkeit	980.000

NC.Chan.ActAxisVelocityPeak

Beschreibung Liefert den Spitzenwert, den Maximalwert und den Nominalwert der aktuellen Achsgeschwindigkeit.

Anmerkung In den ersten sieben Spalten werden Zusatzinformationen pro Achse geliefert.

Die Spalten acht bis zehn enthalten Spitzenwert, Maximalwert und Nominalwert der Achsgeschwindigkeit. Durch Division des Spitzenwerts und des Maximalwerts durch den Nominalwert, kann eine prozentuale Größe für den Spitzenwert und den Maximalwert ermittelt werden.



Die Steuerung stellt erst dann Werte für die Geschwindigkeits-Achsauslastungsdiagnose bereit, wenn diese per Systemdateneintrag aktiviert wird (/SysNcDgnCtrl/AxesLoadDgn/VelLoadDgn/DgnMode = 2).

Bitte schalten Sie per Systemdateneintrag die Geschwindigkeits-Achsauslastungsdiagnose wieder aus (.../VelLoadDgn/DgnMode = 0), wenn Sie diese nicht mehr benötigen! Damit werden unnötige Belastungen des Interpolator-Interrupts vermieden.

Item Syntax NC.Chan.ActAxisVelocityPeak,(1)

1. Kanalnummer

Aufruf synchron / asynchron
Zugriff Read
zyklisch (opt.) ja
Datenformat NC.Chan.ActAxisVelocityPeak, read

row, col	Typ	Beschreibung	Bsp.
n,0	long	phys. Achs- oder Kanalnummer	2
n,1	long	Kanalzugehörigkeit	1

row, col	Typ	Beschreibung	Bsp.
n,2	string	logischer Achs- oder Kanalname	Z
n,3	long	Achse in Position 0 = FALSE 1 = TRUE	1
n,4	long	Referenzpunkt bekannt	1
n,5	long	Maßeinheit	LDX_McoUnitMetric_C
n,6	long	Durchmesserprogrammierung aktiv	0
n,7	double	Spitzenwert der Achsgeschwindigkeit	10500.000
n,8	double	Maximalwert der Achsgeschwindigkeit	12000.000
n,9	double	Nominalwert der Achsgeschwindigkeit	10000.000

NC.Chan.ActAxisAccelerationPeak

Beschreibung Liefert den Spitzenwert, den Maximalwert und den Nominalwert der aktuellen Achsgeschwindigkeit.

Anmerkung In den ersten sieben Spalten werden Zusatzinformationen pro Achse geliefert.

Die Spalten acht bis zehn enthalten Spitzenwert, Maximalwert und Nominalwert der Achsbeschleunigung. Durch Division des Spitzenwerts und des Maximalwerts durch den Nominalwert, kann eine prozentuale Größe für den Spitzenwert und den Maximalwert ermittelt werden.



Die Steuerung stellt erst dann Werte für die Beschleunigungs-Achsauslastungsdiagnose bereit, wenn diese per Systemdateneintrag aktiviert wird (/SysNcDgnCtrl/AxesLoadDgn/AccLoadDgn/DgnMode = 2). .

Bitte schalten Sie per Systemdateneintrag die Beschleunigungs-Achsauslastungsdiagnose wieder aus (.../AccLoadDgn/DgnMode = 0), wenn Sie diese nicht mehr benötigen! Damit werden unnötige Belastungen des Interpolator-Interrupts vermieden.

Item Syntax NC.Chan.ActAxisAccelerationPeak,(1)

1. Kanalnummer

Aufruf synchron / asynchron

Zugriff Read

zyklisch (opt.) ja

Datenformat NC.Chan.ActAxisAccelerationPeak, read

row, col	Typ	Beschreibung	Bsp.
n,0	long	phys. Achs- oder Kanalnummer	2
n,1	long	Kanalzugehörigkeit	1
n,2	string	logischer Achs- oder Kanalname	Z

row, col	Typ	Beschreibung	Bsp.
n,3	long	Achse in Position 0 = FALSE 1 = TRUE	1
n,4	long	Referenzpunkt bekannt	1
n,5	long	Maßeinheit	LDX_McoUnitMetric_C
n,6	long	Durchmesserprogrammierung aktiv	0
n,7	double	Spitzenwert der Achsbeschleunigung	1100.000
n,8	double	Maximalwert der Achsbeschleunigung	1500.000
n,9	double	Nominalwert der Achsbeschleunigung	1300.000

NC.Chan.ActFeedOverride

Beschreibung	Liefert den aktuellen Wert des Vorschub-Overrides in Prozent.
Anmerkung	Neben dem aktuellen Wert wird auch der parametrisierte Maximalwert geliefert.
Item Syntax	NC.Chan.ActFeedOverride,(1) 1. Kanalnummer
Aufruf	synchron / asynchron
Zugriff	Read
zyklisch (opt.)	ja
Datenformat	NC.Chan.ActFeedOverride, read

row, col	Typ	Beschreibung	Bsp.
0,0	double	aktiver Wert in %	85.0
0,1	double	Maximalwert in %	120.0

NC.Chan.ActFeedrate

Beschreibung	Liefert den aktuellen Vorschubwert.
Anmerkung	<i>Es werden drei Bahnvorschubwerte geliefert. Bei jedem Vorschubwert wird der jeweils wirksame Override berücksichtigt:</i> externe Interpolator-Sollgeschwindigkeit: wird dem Interpolator vorgegeben (programmierter Vorschub) Interpolator-Istgeschwindigkeit: momentane Bahngeschwindigkeit interne Interpolator-Sollgeschwindigkeit: kann durch eine Anwendung, z. B. durch eine FeedAdapt-Funktion, gegenüber dem Bahnvorschub 1 geändert werden. Bei Vorschubprogrammierung in mm/U (G95) liefert sie z. B. den Bahnvorschub in mm/min.
Item Syntax	NC.Chan.ActFeedrate,(1) 1. Kanalnummer

Aufruf synchron / asynchron
Zugriff Read
 Geschwindigkeit wird in mm/min angegeben.
zyklisch (opt.) ja
Datenformat NC.Chan.ActFeedrate, read

row, col	Typ	Beschreibung	Bsp.
0,0	double	externe Interpolator-Sollgeschwindigkeit	85
0,1	double	Interpolator-Istgeschwindigkeit	85
0,2	double	interne Interpolator-Sollgeschwindigkeit	85
1,0	long	Nachkommastellen metrisch	1
1,1	long	Nachkommastellen Inch	2
2,0	long	mm/min	1
2,1	double	mm/U	1.000

NC.Chan.ActRotAxisProgrammedSpeed

Beschreibung Liefert die programmierte Drehzahl von asynchron endlosdrehenden Rundachsen.

Anmerkung In den ersten sieben Spalten werden Zusatzinformationen pro Achse geliefert. Die achte Spalte enthält den Positionswert der Achse.

Item Syntax NC.Chan.ActRotAxisProgrammedSpeed,(1)
 1. Kanalnummer

Aufruf synchron / asynchron
Zugriff Read
zyklisch (opt.) ja
Datenformat NC.Chan.ActRotAxisProgrammedSpeed, read

row, col	Typ	Beschreibung	Bsp.
n,0	long	phys. Achs- oder Kanalnummer	2
n,1	long	Kanalzugehörigkeit	1
n,2	string	logischer Achs- oder Kanalname	Y
n,3	long	Achse in Position 0 = FALSE 1 = TRUE	1
n,4	long	Referenzpunkt bekannt	1
n,5	long	Maßeinheit	LDX_McoUnitMetric_C
n,6	long	Durchmesserprogrammierung aktiv	0
n,7	double	Drehzahlwert	

NC.Chan.ActRotAxisSpeed

Beschreibung Liefert die aktuelle Drehzahl von asynchron endlosdrehenden Rundachsen.

Anmerkung	In den ersten sieben Spalten werden Zusatzinformationen pro Achse geliefert. Die achte Spalte enthält den Positionswert der Achse.
Item Syntax	NC.Chan.ActRotAxisSpeed,(1) 1. Kanalnummer
Aufruf	synchron / asynchron
Zugriff	Read
zyklisch (opt.)	ja
Datenformat	NC.Chan.ActRotAxisSpeed, read

row, col	Typ	Beschreibung	Bsp.
n,0	long	phys. Achs- oder Kanalnummer	2
n,1	long	Kanalzugehörigkeit	1
n,2	string	logischer Achs- oder Kanalname	Y
n,3	long	Achse in Position 0 = FALSE 1 = TRUE	1
n,4	long	Referenzpunkt bekannt	1
n,5	long	Maßeinheit	LDX_McoUnitMetric_C
n,6	long	Durchmesserprogrammierung aktiv	0
n,7	double	Drehzahlwert	

NC.Chan.ProgrammedFeedrate

Beschreibung	Liefert die programmierten Vorschubwerte F1 und F2.
Anmerkung	Kennt die NC-Kernsoftware den F2-Wert nicht, dann liefert sie nur den F1-Wert zurück. Für diesen Fall ist der F2-Wert gleich "0".
Item Syntax	NC.Chan.ProgrammedFeedrate,(1) 1. Kanalnummer
Aufruf	synchron / asynchron
Zugriff	Read
zyklisch (opt.)	ja
Datenformat	NC.Chan.ProgrammedFeedrate, read

row, col	Typ	Beschreibung	Bsp.
0,0	double	programmierter Vorschubwert F1	150
0,1	double	programmierter Vorschubwert F2	200
1,0	long	Nachkommastellen metrisch	0
1,1	long	Nachkommastellen Inch	1
2,0	long	mm/min	1
2,1	double	mm/U	1.000

NC.Chan.ProgrammedFeedrateIndex

Beschreibung	Liefert die Information, welcher der programmierten Vorschübe F1 oder F2 gerade aktiv ist.
---------------------	--

Item Syntax	NC.Chan.ProgrammedFeedrateIndex,(1)
	1. Kanalnummer
Aufruf	synchron / asynchron
Zugriff	Read
	1 = Vorschub F1 ist aktiv
	2 = Vorschub F2 ist aktiv
zyklisch (opt.)	ja
Datenformat	NC.Chan.ProgrammedFeedrateIndex, read

row, col	Typ	Beschreibung	Bsp.
0,0	long	aktiver Vorschub	2

4.9 NC-Kommandos

4.9.1 Übersicht

- Item-Überblick
- **NC.Cmd.DryRunMode**
Einstellung des Testbetriebs von Kanal, Achse, Spindel oder Gesamtsystem.
 - **NC.Cmd.MessageStoreClr**
Löscht Fehler, Warnungen und Hinweise in der Fehlerdatenbank der NC.

4.9.2 Globale Kommandos

NC.Cmd.DryRunMode

Beschreibung Liest oder ändert die Einstellung des Testbetriebs von Kanal, Achse, Spindel oder Gesamtsystem.

Anmerkung *Die Daten enthalten*

- den Testbetriebszustand
- die Instanz (Gesamtsystem, Kanal, Achse oder Spindel)
- eine Nummer (Kanal-, Achs- oder Spindelnummer)

Der Enumerator "LDX_Acb_DryRunMode" gibt den Testbetriebszustand vor, er kann folgende Werte annehmen:

Enumerator LDX_Acb_DryRunMode

LDX_SDMode	Beschreibung
LDX_Acb_DryRunModeOff	Testbetrieb aus
LDX_Acb_DryRunModeOn	Testbetrieb ein
LLDX_Acb_TestFeedrateOff	Testvorschub aus
LDX_Acb_TestFeedrateOn	Testvorschub ein
LDX_Acb_TestRapidTraverseOff	Testeilgang aus
LDX_Acb_TestRapidTraverseOn	Testeilgang ein

Der Enumerator "LDX_Acb_DryRunItem" gibt die Instanz an, die eingestellt wird. Er kann folgende Werte annehmen:

Enumerator LDX_Acb_DryRunItem

LDX_SDMode	Beschreibung
LDX_Acb_AllAxis	alle Achsen
LDX_Acb_ChannelAxis	alle kanalzugeordneten Achsen
LLDX_Acb_SingleAxis	eine beliebige Achse
LDX_Acb_AsyncAxis	alle asynchronen Achsen
LDX_Acb_SingleSpindle	eine Spindel

Einschränkung: Bisher wird nur Ein- und Ausschalten der Gesamtsperre (Gesamtsystem), sowie Ein- und Ausschalten von Testvorschub und Testeilgang (Kanal) unterstützt.

Fehler, die bei der Umschaltung auftreten können, sind in Abschnitt [SAV-Fehler auf Seite 225](#) beschrieben.

Item Syntax NC.Cmd.DryRunMode

Aufruf synchron / asynchron

Zugriff Read

- Liest die Einstellung des Testbetriebs

Write

- Ändert die Einstellung des Testbetriebs

zyklisch (opt.) nein

Datenformat NC.Cmd.DryRunMode, write

row, col	Typ	Beschreibung	Einschalten des Testbetriebs in Kanal 2
0,0	long	Testbetriebszustand	LDX_Acb_DryRunModeOn
0,1	string	Instanz	LDX_Acb_ChannelAxis
0,2	long	Nummer	2

NC.Cmd.MessageStoreClr

Beschreibung Löscht Fehler, Warnungen und Hinweise in der Fehlerdatenbank der NC.

Anmerkung Über einen Lösch-Modus wird vorgegeben, welche Meldungen gelöscht werden. Außerdem kann angegeben werden, in welchem Kanal gelöscht werden soll.

Für den Lösch-Modus steht der Enumerator "LDX_WeaClrMode" zur Verfügung. Er kann folgende Werte annehmen:

Enumerator LDX_WeaClrMode

LDX_McaBackupOption	Beschreibung
LDX_WEA_RESETECHAN1_C	Löschen wegen Grundstellung Kanal 1
LDX_WEA_RESETECHAN2_C	Löschen wegen Grundstellung Kanal 2
...	
LDX_WEA_RESETECHAN12_C	Löschen wegen Grundstellung Kanal 12
LDX_WEA_CLRWRNG_C	Warnungen löschen
LDX_WEA_CLRRUNTIMEANCMT_C	Programmierte Hinweise löschen
LDX_WEA_ALLCHANRESET_C	Löschen wegen Systemgrundstellung

LDX_McaBackupOption	Beschreibung
LDX_WEA_WARMSTART_C	Löschen wegen Warmstart
LDX_WEA_COLDSTART_C	Löschen wegen Kaltstart

Der Enumerator "LDX_WeaClrChan" gibt die Kanäle an, in denen gelöscht wird:

Enumerator LDX_WeaClrChan

LDX_McaBackupOption	Beschreibung
LDX_WEA_CHAN1_C	Löschen in Kanal 1
LDX_WEA_CHAN2_C	Löschen in Kanal 2
...	
LDX_WEA_CHAN12_C	Löschen in Kanal 12
LDX_WEA_ALLCHAN_C	Löschen systemweit

Item Syntax	NC.Cmd.MessageStoreClr
Aufruf	synchron / asynchron
Zugriff	Write
zyklisch (opt.)	nein
Datenformat	NC.Cmd.MessageStoreClr, write

row, col	Typ	Beschreibung	Bsp.: alle Warnungen löschen
0,0	long	Löschmodus	LDX_WEA_CLRWARNG_C
0,1	empty		
0,2	long	Kanäle, in denen gelöscht wird	LDX_WEA_ALLCHAN_C

NC.Cmd.Shutdown

Beschreibung	Das Item führt einen Shutdown der Steuerung durch.
Anmerkung	Alle Tasks werden in Grundstellung und alle Filesysteme in einen konsistenten Zustand gebracht. <i>Die Behandlung des Filesystems vor dem Shutdown wird über den Schreibparameter definiert:</i> • 1 = Ram sichern , UsrFep sichern • 2 = Ram löschen, UsrFep sichern • 3 = Ram sichern, UsrFep löschen
Item Syntax	NC.Cmd.Shutdown
Aufruf	synchron / asynchron
Zugriff	Write
zyklisch (opt.)	nein
Datenformat	NC.Cmd.Shutdown, write

row, col	Typ	Beschreibung	Bsp.: Ram sichern , UsrFep sichern
0,0	long	Auslösen des Shutdowns bei vorherigem Sichern von RAM und UsrFep	1

NC.Cmd.Unlock

Beschreibung	Freischalten der Steuerung mit einem Unlock-Code.
Anmerkung	Der übergebene Unlock-Code wird auf Gültigkeit überprüft. Wenn er gültig ist, wird die Steuerung für die im Code enthaltene Zeit freigeschaltet. Die Version des Unlock-Codes ist optional. Der Defaultwert ist 0.
Item Syntax	NC.Cmd.Unlock
Aufruf	synchron / asynchron
Zugriff	Write
zyklisch (opt.)	nein
Datenformat	NC.Cmd.Unlock, write

row, col	Typ	Beschreibung	Bsp.: (individuelle ermittelte Codierung)
0,0	string	Unlock-Code	nT2t-NjFy-2FpB-VjRA-JESu-EA52-gs6D-42sD
0,1	long	Optional: Version des Unlock-Codes	0

System.Cmd.Reset

Beschreibung	Auslösen der Systemgrundstellung. Hierbei wird eine Grundstellung in allen Kanälen ausgelöst. Eventuell aktive Programme werden abgewählt.
Item Syntax	System.Cmd.Reset
Aufruf	synchron / asynchron
Zugriff	Write
zyklisch (opt.)	nein

System.Cmd.Restart

Beschreibung	Löst einen Steuerungshochlauf aus. Kann der Restart steuerungsseitig durchgeführt werden, läuft die Steuerung hoch.
Item Syntax	System.Cmd.Restart
Aufruf	synchron / asynchron
Zugriff	Write
zyklisch (opt.)	nein

4.9.3 Kanalabhängige Kommandos

NC.Chan.Cmd.MDI

Beschreibung	Übergibt Handeingabe-Sätze an einen vorgegebenen Kanal.
Anmerkung	Die übergebenen NC-Sätze werden von UNICODE nach ANSI konvertiert. Weitere Parameter werden über Flags vorgegeben. Hierfür steht der Enumerator "LDX_Sav_Flag" mit den folgenden Werten zur Verfügung:

Enumerator LDX_Sav_Flag

LDX_Sav_Flag	Beschreibung
LDX_SAV_FLAG_LINK_BEFORE_READY_C	Ist dieses Flag gesetzt, wird während der Anwahl ein Linklauf durchgeführt. Dabei wird eine Linktabelle für das Hauptprogramm angelegt. Diese Tabelle ist notwendig, wenn das Programm Unterprogrammaufrufe oder CPL-Elemente enthält.
LDX_SAV_FLAG_RETURN_BY_READY_C	Normalerweise wird bei der Anwahl anhand der Betriebsart und des NC-Zustands geprüft, ob die Anwahl zulässig ist. Danach wird die Anwahl quittiert. Ist dieses Flag gesetzt, erfolgt die Quittung erst nach Abschluss der Anwahl, d. h. wenn der NC-Status "bereit" ist.
LDX_SAV_FLAG_DA_C	reserviert
LDX_SAV_FLAG_REMOTE_C	reserviert
LDX_SAV_FLAG_SUPERVISOR_C	reserviert
LDX_SAV_FLAG_MDI_C	Dieses Flag muss gesetzt sein, wenn in der Betriebsart "Handeingabe" ein Satz abgearbeitet werden soll.
LDX_SAV_FLAG_PROCESSDELETE_C	Zuerst wird ein aktives Programm oder eine aktive Handeingabe abgewählt.
LDX_SAV_FLAG_AUTOSTART_C	In Verbindung mit LDX_SAV_FLAG_MDI_C: Der eingegebene Satz wird sofort gestartet. Ist der angegebene Kanal nicht aktiv, wird der Satz in diesem Kanal gestartet. Alle NC-Sätze sind dann zugelassen. Ist der angegebene Kanal aktiv, wird der Satz wie eine Schaltfunktion behandelt. Einschränkungen: -> LDX_SAV_FLAG_MACHINE_C
LDX_SAV_FLAG_JOG_C	Dieses Flag muss gesetzt sein, wenn in der Betriebsart "Tippbetrieb" Achsenverfahren werden sollen oder wenn in der Betriebsart "Einrichten" manuelle Werkstückkoordinaten-Bewegungen stattfinden sollen.
LDX_SAV_FLAG_REFPT_C	Dieses Flag muss gesetzt sein, wenn in der Betriebsart "Referenzpunkt Anfahren" Achsen gestartet werden sollen.
LDX_SAV_FLAG_MACHINE_C	Schaltfunktion wirkt in Verbindung mit LDX_SAV_FLAG_MDI_C. Der Satz wird parallel zum angegebenen Kanal ausgeführt. Mit Schaltfunktion sind allerdings nur Hilfsfunktionen und asynchrone Achsbewegungen zulässig.

LDX_Sav_Flag	Beschreibung
LDX_SAV_FLAG_PROCESSCHANGE_C	Ist dieses Flag gesetzt, wird ein bereits aktives Programm am Programmstart nicht durch Abwahl abgebrochen, sondern durch das neu angewählte ersetzt, die modalen Zustände bleiben erhalten. Bei Handeingabe wird die alte Zeichenfolge durch die neue ersetzt.
LDX_SAV_FLAG_BUFFERED_INPUT_C	Gepufferte NC-Satzvorgabe wirkt in Verbindung mit LDX_SAV_FLAG_MDI_C. Während Sätze noch abgearbeitet werden, können weitere vorgegeben werden.
LDX_SAV_FLAG_NO_CHAN_CHANGE_C	Der vorgegebene Kanal wird beibehalten, es erfolgt kein impliziter Wechsel auf den Autostart-Kanal.
LDX_SAV_FLAG_RSP_IMMEDIATELY_C	Der Auftrag wird nach Bearbeitung quittiert und nicht erst wenn Satz aktiv ist.
LDX_SAV_FLAG_SET_SEARCHRUN_C	Mit diesem Flag können Einstellungen für den Satzvorlauf vorgenommen werden, die als Xml-String anzugeben sind. Beispiel für die Eingabe der Zielposition 100: <SearchRun><DestPos>100</DestPos></SearchRun>
LDX_SAV_FLAG_MORE_FOLLOW_C	Es folgen weitere Nachrichten mit NC-Sätzen. Die Satz vorausschau wartet so lange, bis weitere Sätze vorbereitet werden können. Ist das Flag nicht gesetzt und es kommt zu einer Kommunikationsverzögerung, erzeugt die MTX automatisch eine Down-Rampe und hält definiert an.
LDX_SAV_FLAG_OVERSTORE_C	Überspeichern in den Betriebsarten Automatik, Programmsatz, Einzelsatz und Einzelschritt. Nach dem Anhalten eines aktiven Programms mit VS-Halt kann mit der NC-Satzvorgabe und diesem Flag ein Satz an den Kanal übertragen werden. Der Kanalzustand geht dann auf bereit. Die Kanalbetriebsart bleibt auf Automatik. Mit dem nächsten NC-Start wird der vorgegebene Satz abgearbeitet.

Item Syntax NC.Chan.Cmd.MDI,(1)

1. Kanalnummer

Aufruf synchron / asynchron

Zugriff Write

Schreiben ist nur zulässig, wenn die letzte Handeingabe beendet ist (erfolgreich oder fehlerhaft). Falls Fehler auftreten, wird der Write-Vorgang negativ quittiert.



Einschränkungen für Handeingabedaten:

- Mehrere Sätze werden gesammelt übergeben. Um Einzelsatz zu realisieren, muss jeder Satz getrennt übergeben werden.
- CPL-Elemente in den Handeingabedaten sind unzulässig, es besteht jedoch die Möglichkeit eines Unterprogrammaufrufs in den Handeingabesätzen.
- Linken vor Ausführung der Handeingabesätze wird vom Kern nicht unterstützt.
- Die Länge eines NC-Satzes ist auf 511 Zeichen beschränkt.
- Bei Übergabe mehrere NC-Sätze ist die Gesamtlänge auf 8 kByte (entspricht 4 kByte in ANSI) beschränkt.
- Als Endekennung ist nur LF (\n) zulässig.

Bedingung für den Satzvorlauf:

- Alle Satzvorlaufdaten werden in einem String übergeben, der in den XML-Tags **<SearchRun>** und **</SearchRun>** eingeschlossen ist.

zyklisch (opt.) nein

Datenformat NC.Chan.Cmd.MDI, write

row, col	Typ	Beschreibung	Bsp.
0,0	long	Flag	LDX_SAV_FLAG_MDI_C
n,0	string	Handeingabe-Satz	N10 G1 F100 X35

NC.Chan.Cmd.MidProgramStartup

Beschreibung Parametrierung des Satzvorlaufs im vorgegebenen Kanal.

Anmerkung Zusätzliche Werte können über Flags vorgegeben werden. Hierfür steht der Enumerator "LDX_Sav_Flag_Set" mit den folgenden Werten zur Verfügung:

Enumerator LDX_Sav_Flag_Set

LDX_Sav_Flag_Set	Beschreibung
LDX_SAV_FLAG_SET_INTERRUPT_C	steuert asynchrone UP. In StartBlock wird die Nummer (1 bis 8) als String ("1") angegeben, in EndBlock der Auftrag. Aufträge für asynchrone UP: "SETINT" anmelden "DISABLE" ausschalten "ENABLE" wieder einschalten "CLRINT" löschen "START" starten
LDX_SAV_FLAG_SET_SEARCHRUN_C	Einstellungen für den Satzvorlauf Die Satzvorlaufdaten werden von <SearchRun> ... /<SearchRun> eingeschlossen.

Item Syntax NC.Chan.Cmd.MidProgramStartup,(1)

1. Kanalnummer

Aufruf	synchron / asynchron
Zugriff	Write
	Es wird der Name des Hauptprogramms angegeben. Außerdem werden die Satzvorlaufdaten und Flags angegeben.
zyklisch (opt.)	nein
Satzvorlaufdaten	Die Satzvorlaufdaten werden in einem String übergeben, der in <SearchRun> ... </SearchRun> eingeschlossen ist. Position des Zielsatzes im Teileprogramm Syntax: <DestPos>[Position]</DestPos> Legt die Position des Zielsatzes in einem Haupt- oder Unterprogramm fest. Position ist der Offset des ersten Zeichens des Zielsatzes zum Programmanfang. Der erste Satz hat die Position 0. <i>Die Position wird gelöscht mit:</i> • NC-Start zur weiteren Bearbeitung nach Satzvorlauf, • Betriebsartenwechsel, • Grundstellung, • erneute Vorgabe, • leerem Tag <DestPos></DestPos>



Wird der Satzvorlauf durch **<StartCmd>** wiederholt, kann die Bearbeitung in einer Schleife an beliebiger Stelle fortgesetzt werden.

Unterprogramm der Zielposition

Syntax:

<DestFName>[Teileprogramm]</DestFName>

Gibt das Teileprogramm an, in dem sich die Zielposition befindet. Ist keines angegeben, wird die Zielposition im Hauptprogramm gesucht.

Beinhaltet der Name des Teileprogramms keinen absoluten Pfad, dann wird es im Pfad des Hauptprogramms gesucht.

Der Name wird gelöscht mit:

- NC-Start zur weiteren Bearbeitung nach Satzvorlauf,
- Betriebsartenwechsel,
- Grundstellung,
- erneute Vorgabe,
- leerem Tag **<DestFName></DestFName>**

Satzvorlauf starten

Syntax:

<StartCmd>[Kommandos]</StartCmd>

Startet den Satzvorlauf und steuert verschiedene Wiederanfahrstrategien.

*Die Kommandos werden addiert oder bitweise verknüpft (z. B. **Flags = LDX_PRsStartSearchRun_C | LDX_PRsAsupAfterRepos_C**):*

- "**LDX_PRsStartSearchRun_C**" startet den Satzvorlauf am Programmanfang oder zum wiederholten Start nach Erreichen eines Zielsatzes. In Schleifen kann der Zielsatz mehrfach erreicht werden.

Der Start wirkt nur,

- wenn die Zielposition gesetzt ist und
- wenn das Unterprogramm der Zielposition gesetzt ist oder sich die Bearbeitungsposition im Hauptprogramm befindet und
- wenn die aktive Betriebsart "Satzvorlauf" ist.
- "LDX_PRsActionBlkWhileRepos1_C" aktiviert den ersten Aktionssatz. Er wird nach dem NC-Start zur weiteren Bearbeitung oder nach dem ersten Asup ausgeführt.
- "LDX_PRsActionBlkWhileRepos2_C" aktiviert den zweiten Aktionssatz. Er wird nach dem ersten Aktionssatz ausgeführt.
- "LDX_PRsActionBlkAfterRepos_C" aktiviert den dritten Aktionssatz. Er wird nach Erreichen der Endposition des Zielsatzes ausgeführt.
- "LDX_PRsAsupBeforeRepos_C" aktiviert das erste Asup. Es wird nach dem NC-Start zur weiteren Bearbeitung gestartet.
- "LDX_PRsAsupWhileRepos_C" aktiviert das zweite Asup. Es wird nach Beendigung des zweiten Aktionssatz gestartet (falls keiner aktiviert nach dem ersten Asup) .
- "LDX_PRsAsupAfterRepos_C" aktiviert das dritte Asup. Es wird nach dem dritten Aktionssatz gestartet (wenn nicht vorhanden nach Erreichen der Endposition des Zielsatzes).
- "LDX_PRsRcbeCallChainEnable_C" schaltet die CallChain-Generierung des RCBE-Befehls frei.
- "LDX_PRsRunToEndOfProgram_C" Der Satzvorlauf soll das Programm komplett abarbeiten, um z. B. mit RCBE CallChain Marken zu generieren oder das Programm zu prüfen.
- "LDX_PRsUseNcCallChain_C" benutzt die gespeicherte Zustandsstruktur, um den Zielsatz zu finden.

Die Kommandos werden gelöscht mit:

- Beendigung aller Aktionssätze und Asups,
- Betriebsartenwechsel,
- Grundstellung,
- Tag <ResetCmd>...</ResetCmd>

Satzvorlauf Kommandos löschen

Syntax:

<ResetCmd>[Kommandos]</ResetCmd>

Löscht Satzvorlauf-Kommandos. Die Kommandos werden wie bei <StartCmd> gesetzt.

Unterprogrammschachtelung speichern

Syntax:

<StoreCallChain>1</StoreCallChain>

Speichert in einem Kanal eine Zustandsstruktur, bestehend aus einer Call-Chain. Sie kann nach Anhalten einer Bearbeitung für den nächsten Satzvorlauf gesetzt werden.

Sie wird gelöscht mit

- erneuter Vorgabe,
- leerem Tag <StoreCallChain></StoreCallChain>

Verschiedene Aktionssätze anmelden

Syntax:

<ActionBlkWhile1>[Nc-Satz]**</ActionBlkWhile1>**

Syntax:

<ActionBlkWhile2>[Nc-Satz]**</ActionBlkWhile2>**

Syntax:

<ActionBlkAfter>[Nc-Satz]**</ActionBlkAfter>**

Meldet Aktionssätze an.

Parameter:

NC-Satz mit maximaler Länge "LDX_McoMaxNcBlockSize_C"

Sie werden gelöscht mit

- erneuter Vorgabe,
- leerem Tag **<ActionBlk...></ActionBlk...>**

Verschiedene Asups anmelden**Syntax:**

<AsupBefore>[Pfadname] [Anwahlflags]**</AsupBefore>**

Syntax:

<AsupWhile>[Pfadname] [Anwahlflags]**</AsupWhile>**

Syntax:

<AsupAfter>[Pfadname] [Anwahlflags]**</AsupAfter>**

Meldet Asups an. Sie können vor, während oder nach dem Wiederanfahren aufgerufen werden.

Parameter:

- Pfadname des Asups
Fehlt der Pfad, dann wird im Pfad des Hauptprogramms gesucht.
- Anwahlflags der Programmanwahl "LDX_SavProcessCreate".
Wichtig sind "LDX_SAV_FLAG_LINK_BEFORE_READY_C" für Linken und "LDX_SAV_FLAG_AUTOSTART_C" für automatischen Start des Asups.

Sie werden gelöscht mit

- erneuter Vorgabe,
- leerem Tag **<Asup...></Asup...>**

Beispiel

Es soll "/usrfep/sekt.cnc" angewählt werden. Zielsatz ist der Satz "N20". Der Satzvorlauf soll in der Betriebsart "Satzvorlauf" gestartet werden. Nach NC-Start zur weiteren Bearbeitung soll ein Aktionssatz und nach Erreichen der Zielposition soll das Asup "sektAsup.cpl" aus "/usrfep" gestartet werden.

Erläuterungen zum XML-Stream (Satzvorlaufdaten):

```
<SearchRun>
```

```
<DestPos>25</DestPos>
```

```
<DestFName>/usrfep/sekt.cnc</DestFName>
```

```
// in diesem Fall überflüssig
```

```
// Starten (1) mit Aktivieren von Aktionssatz 1 (2) und  
Asup 3 (64) (=67)
```

```
<StartCmd>67</StartCmd>
```

```
// Asup soll mit Linken (1) und Autostart (128) angewählt werden.
```

```
<AsupAfter>sektAsup.cpl>129</AsupAfter>
```

```
<ActionBlkWhile1>M99 S1000</ActionBlkWhile1>
```

```
</SearchRun>
```

Datenformat NC.Chan.Cmd.MidProgramStartup, write

row, col	Typ	Beschreibung	Bsp.
0,0	string	Programmname mit Pfad	/mnt/sekt.npg
0,1	string (XML-Stream)	Satzvorlaufdaten	<SearchRun> <DestPos>106</DestPos> <StartCmd>1<\StartCmd> <\SearchRun>
0,2	empty	reserviert???	
0,3	long	Flags	LDX_SAV_FLAG_SET_SEAR- CHRUN_C

NC.Chan.Cmd.ModeOfOperation

Beschreibung Dient zur Umschaltung der Betriebsart in einem Kanal.

Anmerkung Zur Codierung der gewünschten Betriebsarten wird der Enumerator "LDX_SavOpModes" verwendet. Dieser Aufzählungstyp kann folgende Werte annehmen:

Enumerator LDX_SavOpModes

LDX_SavOpModes	Beschreibung
LDX_SAV_OP_MODE_EMPTY_C	keine Betriebsart aktiv
LDX_SAV_OP_MODE_JOGGING_C	Tippbetrieb
LDX_SAV_OP_MODE_REFERENCE_C	Referenzpunkt Anfahren
LDX_SAV_OP_MODE_BLOCKS-EARCH_C	Satzvorlauf
LDX_SAV_OP_MODE_MDI_C	Handeingabe
LDX_SAV_OP_MODE_AUTOMATIC_C	Automatik: Teileprogramme können in Folgesatz abgearbeitet werden
LDX_SAV_OP_MODE_SIN- GLE_BLOCK_C	Programmsatz: Mit "NC-Start" werden einzelne Sätze, wie sie im Teileprogramm stehen, abgearbeitet. Die Satzvorbereitung startet erst mit "NC-Start".
LDX_SAV_OP_MODE_SINGLE_STEP_C	Einzelschritt: Mit "NC-Start" wird ein einzelner Block an den Interpolator zur Bearbeitung weitergereicht.
LDX_SAV_OP_MODE_TEACH_C	reserviert
LDX_SAV_OP_MODE_SUSPEND_C	reserviert

LDX_SavOpModes	Beschreibung
LDX_SAV_OP_MODE_SINGLE_BLOCK_C	Einzelsatz: Mit "NC-Start" werden alle Blöcke, die aufgrund eines einzelnen NC-Satzes im Teileprogramm generiert und vorbereitet wurden, an den Interpolator zur Bearbeitung weitergereicht.
LDX_SAV_OP_MODE_RETURN_TO_PATH_C	Wiederauffahren: Achsen können manuell von der Kontur weggefahren und automatisch oder manuell wieder herangefahren werden.
LDX_SAV_OP_MODE_DEBUG_STEP_C	CPL-Debugger: Es werden einzelne Sätze, wie sie im Teileprogramm stehen, abgearbeitet.
LDX_SAV_OP_MODE_DEBUG_CONTINUE_C	CPL-Debugger: Es werden alle Sätze bis zum nächsten Breakpoint abgearbeitet.
LDX_SAV_OP_MODE_JOGGING_COORDS_C	Einrichten Manuell: Bewegung in Werkstückkoordinaten.

Fehler, die bei Betriebsartenumschaltung auftreten können, sind in Abschnitt 0 beschrieben.

Item Syntax	NC.Chan.Cmd.ModeOfOperation,(1)
	1. Kanalnummer
Aufruf	synchron / asynchron
Zugriff	Write
zyklisch (opt.)	nein
Datenformat	NC.Chan.Cmd.ModeOfOperation, write

row, col	Typ	Beschreibung	Bsp.
n,0	long	Betriebsart	LDX_SAV_OP_MODE_AUTOMATIC_C

NC.Chan.Cmd.OptionalStop

Beschreibung	Schaltet "Wahlweise Halt" für den vorgegebenen Kanal um.
Item Syntax	NC.Chan.Cmd.OptionalStop,(1)
	1. Kanalnummer
Aufruf	synchron / asynchron
Zugriff	Write
zyklisch (opt.)	nein
Datenformat	NC.Chan.Cmd.OptionalStop, write

row, col	Typ	Beschreibung	Bsp.
0,0	bool	Wahlweise Halt	TRUE

NC.Chan.Cmd.Reset

Beschreibung	Grundstellung eines Kanals.
--------------	-----------------------------

Anmerkung Der Enumerator "LDX_Sav_RFlag" steht mit folgendem Wert zur Verfügung:

Enumerator LDX_Sav_RFlag

LDX_Sav_RFlag	Beschreibung
LDX_SAV_RFLAG_NORESELECT_C	unterdrückt bei Grundstellung die implizierte automatische Programmwiederanwahl

Item Syntax NC.Chan.Cmd.Reset,(1)

1. Kanalnummer

Aufruf synchron / asynchron

Zugriff Write

zyklisch (opt.) nein

Datenformat NC.Chan.Cmd.Reset, write

row, col	Typ	Beschreibung	Bsp.: NC.Chan.Cmd.Reset, 3 Grundstellung in Kanal 3
0,0	long	Grundstellung auslösen	1

NC.Chan.Cmd.ReturnToPath

Beschreibung Beendet das Wegfahren von der Kontur, d. h. die Aufzeichnung der Wegfahrbewegungen wird abgeschlossen. Alle weiteren Tippbewegungen werden nicht mehr aufgezeichnet.

Anmerkung Anschließendes "NC-Start" führt zum Wiederanfahren mit der aktiven Wiederanfahrstrategie.

Item Syntax NC.Chan.Cmd.ReturnToPath,(1)

1. Kanalnummer

Aufruf synchron / asynchron

Zugriff Write

zyklisch (opt.) nein

NC.Chan.Cmd.ReturnToPathStrategy

Beschreibung Definiert die Wiederanfahrstrategie durch Vorgabe der Wiederanfahrbetriebsart und des Wiederanfahrpunktes.

Anmerkung Die Wiederanfahrbetriebsart wird mit Hilfe des Enumerators "LDX_McoRetPath_Mode" bestimmt, der folgende Werte annehmen kann:

Enumerator LDX_McoRetPath_Mode

LDX_McoRetPath_Mode	Beschreibung
LDX_McoRetPathAutomatic_C	automatisches Wiederanfahren
LDX_McoRetPathSingleBlock_C	Wiederanfahren mit Einzelsatz
LDX_McoRetPathManual_C	manuelles Wiederanfahren

Der Wiederanfahrpunkt wird mit Hilfe des Enumerators "LDX_McoRetPath_Point" bestimmt, der folgende Werte annehmen kann:

Enumerator LDX_McoRetPath_Point

LDX_McoRetPath_Point	Beschreibung
LDX_McoRetPathStartOfBlock_C	Wiederauffahren zum Startpunkt
LDX_McoRetPathEndOfBlock_C	Wiederauffahren zum Endpunkt
LDX_McoRetPathPointOfInterruption_C	Wiederauffahren zum Unterbrechungspunkt

Item Syntax NC.Chan.Cmd.ReturnToPathStrategy,(1)

1. Kanalnummer

Aufruf synchron / asynchron

Zugriff Write

zyklisch (opt.) nein

Datenformat NC.Chan.Cmd.ReturnToPathStrategy, write

row, col	Typ	Beschreibung	Bsp.
0,0	long	Wiederauffahrtbetriebsart	LDX_McoRetPathAutomatic_C
0,1	long	Wiederauffahrtspunkt	LDX_McoRetPathStartOfBlock_C

NC.Chan.Cmd.SearchBlock

Beschreibung Positioniert im angewählten Hauptprogramm auf den vorgegebenen NC-Satz.

Anmerkung "Satz-Suchen" ist nur im Hauptprogramm möglich.

Die Parameter (Kennungen) "Startsatz" und optional "Endesatz" werden als String übergeben.

Entsprechend folgender Konvention wird gesucht:

- Leerzeichen am Anfang eines NC-Satzes werden ignoriert,
- Wird eine Kennung, die mit einer Ziffer beginnt, nicht gefunden, dann wird nach N<Kennung> gesucht.
z. B. Kennung "50" findet NC-Satz "N50X100"
- Endet die Kennung auf eine Ziffer, dann dürfen im gefundenen NC-Satz keine weiteren Ziffern folgen.
z. B. Kennung "G1X10" findet nicht NC-Satz "G1X100"
- Endet die Kennung auf einen Buchstaben, dann muss im gefundenen NC-Satz ein Leerzeichen folgen.
z. B. Kennung "50A" findet NC-Satz "50A = 1", aber nicht "50A=1".
- Ansonsten wird ein NC-Satz gefunden, wenn Kennung und Anfang des NC-Satzes übereinstimmen.
z. B. Kennung "G1" findet "G1F1000"

Werden die Kennungen gefunden, beginnt die Bearbeitung mit dem Satz der Start-Kennung und endet mit dem Satz der Ende-Kennung. Fehlt eine Angabe oder wird eine Kennung nicht gefunden, beginnt die Bearbeitung am Programmstart bzw. endet am Programmende. Ist der Suchlauf beendet, dann geht der NC-Status auf "bereit" (LDX_SAV_OP_STATE_READY_C).

Item Syntax NC.Chan.Cmd.SearchBlock,(1)

1. Kanalnummer

Aufruf synchron / asynchron

Zugriff Write

Es wird der Programmname mit Pfad angegeben.

Außerdem können Start- und Endesatz und Flags angegeben werden.

zyklisch (opt.) nein

Datenformat NC.Chan.Cmd.SearchBlock, write

row, col	Typ	Beschreibung	Bsp.
0,0	string	Start-Satz	N545
0,1	string	Ende-Satz	

NC.Chan.Cmd.SelectProgram

Beschreibung Wählt ein Programm im vorgegebenen Kanal an.

Anmerkung Zusätzliche Werte können über Flags eingestellt werden. Hierfür steht der Enumerator "LDX_SavFlags" mit den folgenden Werten zur Verfügung:

Enumerator LDX_SavFlags

LDX_SavFlags	Beschreibung
LDX_SAV_FLAG_LINK_BEFORE_READY_C	Ist dieses Flag gesetzt, wird während der Anwahl ein Linklauf durchgeführt. Dabei wird eine Linktabelle für das Hauptprogramm angelegt. Diese Tabelle ist notwendig, wenn das Programm Unterprogrammaufrufe oder CPL-Elemente enthält.
LDX_SAV_FLAG_RETURN_BY_READY_C	Normalerweise wird bei der Anwahl anhand der Betriebsart und des NC-Zustands geprüft, ob die Anwahl zulässig ist. Danach wird die Anwahl quittiert. Ist dieses Flag gesetzt, erfolgt die Quittung erst nach Abschluss der Anwahl, d. h. wenn der NC-Status "bereit" ist.
LDX_SAV_FLAG_DA_C	reserviert
LDX_SAV_FLAG_REMOTE_C	reserviert
LDX_SAV_FLAG_SUPERVISOR_C	reserviert
LDX_SAV_FLAG_MDI_C	Dieses Flag muss gesetzt sein, wenn in der Betriebsart "Handeingabe" ein Satz abgearbeitet werden soll.
LDX_SAV_FLAG_PROCESSDELETE_C	Zuerst wird ein aktives Programm oder eine aktive Handeingabe abgewählt.
LDX_SAV_FLAG_AUTOSTART_C	In Verbindung mit LDX_SAV_FLAG_MDI_C: Der eingegebene Satz wird sofort gestartet. Ist der angegebene Kanal nicht aktiv, wird der Satz in diesem Kanal gestartet. Alle NC-Sätze sind dann zugelassen. Ist der angegebene Kanal aktiv, wird der Satz wie eine Schaltfunktion behandelt. Einschränkungen: -> LDX_SAV_FLAG_MACHINE_C

LDX_SavFlags	Beschreibung
LDX_SAV_FLAG_JOG_C	Dieses Flag muss gesetzt sein, wenn in der Betriebsart "Tippbetrieb" Achsen verfahren werden sollen oder wenn in der Betriebsart "Einrichten" manuelle Werkstückkoordinaten-Bewegungen stattfinden sollen.
LDX_SAV_FLAG_REFPT_C	Dieses Flag muss gesetzt sein, wenn in der Betriebsart "Referenzpunkt Anfahren" Achsen gestartet werden sollen.
LDX_SAV_FLAG_MACHINE_C	Schaltfunktion wirkt in Verbindung mit LDX_SAV_FLAG_MDI_C. Der Satz wird parallel zum angegebenen Kanal ausgeführt. Mit Schaltfunktion sind allerdings nur Hilfsfunktionen und asynchrone Achsbewegungen zulässig.
LDX_SAV_FLAG_PROCESSCHANGE_C	Ist dieses Flag gesetzt, wird ein bereits aktives Programm am Programmstart nicht durch Abwahl abgebrochen, sondern durch das neu angewählte ersetzt, die modalen Zustände bleiben erhalten. Bei Handeingabe wird die alte Zeichenfolge durch die neue ersetzt.
LDX_SAV_FLAG_BUFFERED_INPUT_C	Gepufferte NC-Satzvorgabe wirkt in Verbindung mit LDX_SAV_FLAG_MDI_C. Während Sätze noch abgearbeitet werden, können weitere vorgegeben werden.
LDX_SAV_FLAG_NO_CHAN_CHANGE_C	Der vorgegebene Kanal wird beibehalten, es erfolgt kein impliziter Wechsel auf den Autostart-Kanal.
LDX_SAV_FLAG_RSP_IMMEDIATELY_C	Der Auftrag wird nach Bearbeitung quittiert und nicht erst, wenn Satz aktiv ist.
LDX_SAV_FLAG_CALLCHAIN_SELECT_C	Es erfolgt eine Programmanwahl anhand der intern gespeicherten Callchain. Dies wirkt nur, wenn für "Programmname mit Pfad" ein Leerstring übergeben wird.

Item Syntax NC.Chan.Cmd.SelectProgram,(1)

1. Kanalnummer

Aufruf synchron / asynchron

Zugriff Write

Es wird der Programmname mit Pfad angegeben.

Außerdem können Start- und Endesatz und Flags angegeben werden.

zyklisch (opt.) nein

Datenformat NC.Chan.Cmd.SelectProgram, write

row, col	Typ	Beschreibung	Bsp.
0,0	string	Programmname mit Pfad	/mnt/myprogram.npg
0,1	string	Start-Satz	""

row, col	Typ	Beschreibung	Bsp.
0,2	string	Ende-Satz	""
0,3	long	Flags	LDX_SAV_FLAG_LINK_BEFORE_READY_C LDX_SAV_FLAG_PROCESSDELETE_C

NC.Chan.Cmd.SkipBlockMode

Beschreibung	Schaltet "Satz Überlesen" für den vorgegebenen Kanal um.
Item Syntax	NC.Chan.Cmd.SkipBlockMode,(1) 1. Kanalnummer
Aufruf	synchron / asynchron
Zugriff	Write
zyklisch (opt.)	nein
Datenformat	NC.Chan.Cmd.SkipBlockMode, write

row, col	Typ	Beschreibung	Bsp.
0,0	bool	Satz überlesen	TRUE

4.10 Fehlermeldungen

4.10.1 Fehlerbehandlung

Um dem Anwender genaue Informationen über Fehler zur Verfügung zu stellen, wird das Standard-SCP-Fehlerhandling unterstützt. Fehlerinformationen werden über die Schnittstelle "ISCPError" bereitgestellt. Falls bei der Ausführung eines angeforderten Dienstes ein Fehler auftritt, wird ein SCP-Fehlercode zurückgegeben. Bei fehlerhafter Benutzung der SCP-Schnittstelle (z.B. falsche Parametrierung beim Anlegen eines Items) werden Standard-SCP-Fehlercodes geliefert.

Treten bei der Ausführung einer angeforderten Funktion (z.B. Programmwahl, Lesen einer Tabelle) Fehler auf, werden gerätespezifische Fehlercodes geliefert. Fehler, die eindeutig einem gerätespezifischen Server zugeordnet sind, liefern einen individuellen Fehlercode. Alle anderen erscheinen als allgemeine SCPErrNACK(x)-Meldungen.

Über die ISCPError-Schnittstelle liefert das Logical Device MTX weitergehende Informationen zu dem aufgetretenen Fehler:

- die Zuordnung zum jeweiligen NCS-Server
- eine Kurzbeschreibung des Fehlers in Textform
- einen eindeutigen Fehlercode, der als Fehlerkonstante abrufbar ist und eine genaue Analyse des Fehlers ermöglicht
- eine ausführliche Beschreibung des Fehlers in Textform

Die folgende Tabelle gibt einen Überblick über die Fehlerinformationen:

Fehlercode	Beschreibung
SCPErrTimeOut - 1	siehe SCP-Doku
SCPErrFailed - 2	
...	
SCPErrDriver - 17	
reserviert (bis 100)	
SCPErrNACK1 - 101	Sercos Fehler
SCPErrNACK2 - 102	ISP Fehler
SCPErrNACK3 - 103	NCS-Fehler
SCPErrNACK4 - 104	Fehler vom NCS-Server (SAV, MCA)
SCPErrNACK5 - 105	allgemeine MTX-Fehler
SCPErrNACK6 - 106	Firmware Monitor-Fehler
SCPErrNACK7 - 107	Profibus-Fehler
SCPErrNACK8 - 108	Fehler MTX FIS-Server
SCPErrNACK9 - 109	MTX-Fehler bei Dateioperationen
SCPErrNACK10 - 110	Win32-Fehler
SCPErrNACK11 - 111	Sercos Fehler (SIP)
...	reserviert
SCPErrNACK17 - 117	CoDeSys Gateway-Fehler
SCPErrNACK18 - 118	Verwaltungsfehler
SCPErrNACK19 - 119	reserviert
SCPErrMtxKrnPxr - 120	Fehler Basic-Kernel
SCPErrMtxKrnSCS - 121	Fehler Sercos Server
SCPErrMtxKrnACB - 122	Fehler Achsbewegungs-Server
SCPErrMtxKrnDIV - 123	Fehler diverser Kernel-Gruppen (Konfig, DB, System, XML und Tabellen)
SCPErrMtxKrnWEA - 124	Fehler Wera-Server
SCPErrMtxKrnFII - 125	Fehler Filesystem-Info
SCPErrMtxKrnCPL - 126	Fehler CPL-Server
SCPErrMtxKrnTAR - 127	Fehler Archivierungs-Server
SCPErrMtxKrnTSV - 128	Fehler Text-Server
...	reserviert für MTX
SCPErrNACK40 - 140	reserviert für MTX

Tab. 4-33: Fehlerinformationen

Fehlercode SCPErrNACK3

Der Fehlercode SCPErrNACK3 gibt an, dass ein NCS-Fehler aufgetreten ist. Die Kurzbeschreibung enthält den Namen der NCS-Funktion, bei der der Fehler aufgetreten ist. Der detaillierte Fehlercode und die detaillierte Fehler-

beschreibung bezeichnen den aufgetretenen NCS-Fehler genauer. Diese Fehler sollten im Normalbetrieb nicht auftreten.

Fehlercode SCPErrNACK4

Der Fehlercode SCPErrNACK4 gibt an, dass bei der Ausführung einer NCS-Funktion ein Fehler des SAV- bzw. MCA-Servers gemeldet wurde oder der Zugriff auf modale NCS-Funktionen fehlerhaft war.

Zum Fehler werden eine Kurzbeschreibung, der genaue Fehlercode und die detaillierte Fehlerbeschreibung geliefert. Die Fehlercodes sind als definierte Fehlerkonstanten beschrieben. Die Dokumentation der Items verweist auf Fehler, die für dieses Item relevant sind und in den folgenden Kapiteln erläutert werden.

Fehlercodes 120 bis 128

Diese Fehlercodes sind den beschriebenen NCS-Servern der MTX zugeordnet.

Gemeldet wird die Zuordnung zum NCS-Server, eine Kurzbeschreibung, der genaue Fehlercode vom Server sowie eine detaillierte Fehlerbeschreibung. Die den einzelnen Fehlercodes zugewiesenen Fehlerkonstanten sind in den folgenden Kapiteln beschrieben.

4.10.2 SAV-Fehler

Die folgende Tabelle gibt die vom SAV-Server verwendeten Fehlercodes wieder, die u.a. bei Betriebsartenumschaltung, Programmanwahl oder Grundstellung geliefert werden können:

Fehlercode	Beschreibung
LDX_SAV_SERVICEOK_C	kein Fehler aufgetreten
LDX_SAV_ILLCHANNUMBER_C	angesprochener Kanal nicht vorhanden
LDX_SAV_CHANUSED_C	Dienst kann nicht ausgeführt werden, da Kanal belegt ist.
LDX_SAV_CHANRESETTING_C	Dienst kann nicht ausgeführt werden, da die Grundstellung des Kanals nicht beendet ist.
LDX_SAV_FILENAMEETOOLONG_C	Programmname zu lang (z.Z. nicht genutzt)
LDX_SAV_NOREFPKT_C	Dienst erfordert angefahrne Referenzpunkte
LDX_SAV_NOPROGRAM_C	Angewähltes Programm existiert nicht oder ist nicht ausführbar.
LDX_SAV_SERVICEINTERRUPTED_C	Schreiben in den Puffer unterbrochen bei gepufferter NC-Satzvorgabe. Eine zweite Instanz hatte gleichzeitig versucht in den Puffer zu schreiben.
LDX_SAV_ILLMODE_C	Dienst kann nicht ausgeführt werden, da die momentane Betriebsart dies nicht zulässt.
LDX_SAV_STATENOTREADY_C	Kanal kann nicht gestartet werden, da der Zustand des Kanals nicht "bereit" ist.
LDX_SAV_NOTSELECTED_C	Dienst kann nicht ausgeführt werden, da kein Programm angewählt ist.

Items

Fehlercode	Beschreibung
LDX_SAV_EXECUTEFEEDHOLD_C	Programm kann nicht abgewählt werden, da der Zustand des Kanals dies nicht zulässt (z. B. Sav und Ipo "läuft").
LDX_SAV_ILLREMLOC_C	z. Z. nicht benutzt
LDX_SAV_ILLSTATE_C	Betriebsart kann nicht gewechselt werden, da der Zustand des Kanals dies nicht zulässt.
LDX_SAV_UNKNOWNLABEL_C	Ziel von "Satz suchen" nicht gefunden
LDX_SAV_STATENOTREADYFOR- PROGSTART_C	"Satz suchen" nicht möglich, da der Zustand des Kanals zwar "bereit" ist, die Bearbeitung des Hauptprogramms aber bereits begonnen hat (Programm steht z.B. auf M0)
LDX_SAV_NMBOFAXISEXCCEEDED_C	zu hohe Anzahl der Achsen bei der Vorgabe externer NPVen
LDX_SAV_NMBOFEXTZOFFSGRP- SEXCEEDED_C	zu hohe Anzahl der NPV-Gruppen bei der Vorgabe externer NPVen
LDX_SAV_UNKNOWNSYNTAX_C	vorgegebene Syntax nicht bekannt
LDX_SAV_ILLTOOLCOMPINDEX_C	unzulässiger Index bei der Vorgabe einer ext. Werkzeugkorrektur
LDX_SAV_NMBOFEXTTOOLCOMP- SEXCEEDED_C	zu hohe Anzahl der Korrekturen bei der Vorgabe einer ext. Werkzeugkorrektur (evtl. in Verbindung mit Korrekturindex)
LDX_SAV_ILLTOOLCOMPTYPE_C	unzulässige Formatangabe bei der Vorgabe einer ext. Werkzeugkorrektur
LDX_SAV_ILLTOOLCOMPORIENTATI- ON_C	unzulässige Schneidenlage bei der Vorgabe einer ext. Werkzeugkorrektur
LDX_SAV_ILLTOOLCOMPGROUP_C	unzulässige Korrekturgruppe
LDX_SAV_ILLAXISNUMBER_C	angesprochene Achse nicht vorhanden
LDX_SAV_RUNTIMEERROR_C	Laufzeitfehler bei NC-Satzvorgabe mit automatischem Start, z.B. Syntaxfehler
LDX_SAV_BUFFER_OVERFLOW_C	Überlauf des Puffers bei einer gepuffer-ten NC-Satzvorgabe
LDX_SAV_ILLCOORDFILTER_C	Vorgabe für Koordinatenfilter nicht korrekt
LDX_SAV_VNCILLSTATE_C	Vorgabe für VNC nicht erlaubt, da VNC-Zustand nicht eingefroren
LDX_SAV_NOMEM_C	Speichermangel im Server. Angeforderte Daten konnten nicht abgelegt werden.
LDX_SAV_ILLPARAM_C	Der NCS-Funktion wurde ein unzulässiger Wert vorgegeben, Server kann Daten nicht liefern.
LDX_SAV_ILLREQUEST_C	interner Fehler
LDX_SAV_ILLRESPONSE_C	interner Fehler

Fehlercode	Beschreibung
LDX_SAV_FUNCTIONNOTIMPLEMEN- TED_C	angesprochener Dienst in dieser Softwa- reversion nicht verfügbar
LDX_SAV_EVDATAFAIL_C	interner Fehler

Tab. 4-34: SAV-Fehlercodes

4.10.3 MACODA-Fehler

Die folgende Tabelle gibt die vom MACODA-Server verwendeten Fehlercodes wieder. Sie können beim Lesen von MACODA-Parametern auftreten:

Fehlercode	Beschreibung
LDX_MCA_OK_C	kein Fehler
LDX_MCA_SQL_NOTFOUND_C	Parameter in aktueller Software-Version nicht vorhanden
LDX_MCA_INSUFFICIENT_MEMORY_C	verfügbarer Speicherbereich reicht nicht zur Aufnahme der angeforderten Daten
LDX_MCA_NO_GROUP_OR_AREA_VE RSION_C	zur angegebenen Blockkennung gibt es keine Gruppen- oder Funktionsbereichs- Version
LDX_MCA_CHANNEL_SPECIFIC_PA- RAMETER_EXIST_C	
LDX_MCA_NO_CHANNEL_SPECI- FIC_PARAMETER_C	
LDX_MCA_NO_CONCURRENT_EDIT_C	

Tab. 4-35: MACODA-Fehlercodes

4.10.4 TAD-Fehler

Die folgende Tabelle gibt die vom Tabellen-Server verwendeten Fehlercodes wieder. Sie können bei Lese- oder Schreib-Zugriffen auf z.B. D-Korrekturta-
bellen auftreten.

Die im Abschnitt [XML-Fehler, Seite 232](#) aufgeführten XML-Fehler können ebenfalls vom Tabellen-Server gemeldet werden.

Fehlercode	Beschreibung
LDX_TAD_ERR_OK_C	kein Fehler
LDX_TAD_ERR_INTERNAL_ERROR_C	interner Fehler (s. Wera-Fehler)
LDX_TAD_ERR_SERVER_NOT_REA- DY_C	Server nicht bereit (s. Wera-Fehler)
LDX_TAD_ERR_NAME2LONG_C	unzulässige Namenlänge
LDX_TAD_ERR_UNKNOWN_TABLE- TYPE_C	unbekannter Tabellentyp
LDX_TAD_ERR_TABLE_NOT_FOUND_ C	Tabelle nicht vorhanden
LDX_TAD_ERR_TABLE_EXISTS_C	Tabelle existiert bereits
LDX_TAD_ERR_BAD_DATASIZE_C	Datenpuffer zu klein

Fehlercode	Beschreibung
LDX_TAD_ERR_BAD_PATHSIZE_C	Path-Puffer zu klein
LDX_TAD_ERR_BAD_SEGMENTPATH-SIZE_C	Segment-Path zu klein (intern)
LDX_TAD_ERR_SCHEMA_INIT_FAILED_C	Fehler beim Initialisieren des Schemas
LDX_TAD_ERR_BAD_NCS_MSG_C	ungültige NCS-Nachricht (nur von ..X-Funktionen benutzt) (intern)
LDX_TAD_ERR_NO_MORE_MOD_EVENTS_C	maximale Anzahl der ModEvents erreicht, keine weiteren ModEvents können angelegt werden
LDX_TAD_ERR_UNKNOWN_MOD_EVENT_C	Unbekannte Event-Id
LDX_TAD_ERR_EVENT_ALREADY_REQ_C	ModEvent wurde bereits angefordert
LDX_TAD_ERR_EVENT_DELETED_C	Fehlercode einer Event-Anforderung nach Löschen des zugehörigen ModEvents (kein Fehler)
LDX_TAD_ERR_CHAN_NUMB_TOO_BIG_C	Kanalanzahl zu groß
LDX_TAD_ERR_EXTENSION_C	Fehler in Datei-Extension
LDX_TAD_ERR_INSERT_AXIS_NOT_ALLOWED_C	Achse einfügen nur bei NPV-Tabellen erlaubt
LDX_TAD_ERR_DELETE_AXIS_NOT_ALLOWED_C	Achse löschen nur bei NPV-Tabellen erlaubt
LDX_TAD_ERR_PATH_BLOCK_POS_INVALID_C	ungültiger XPath für diese Funktion
LDX_TAD_ERR_MAX_NUMBER_AXES_REACHED_C	maximale Achsanzahl erreicht
LDX_TAD_ERR_AXIS_POSITION_NOT_EXIST_C	Achse nicht vorhanden
LDX_TAD_ERR_INPUT_PARAMETER_C	ungültige Parameter
LDX_TAD_ERR_INVALID_SCHEMA_C	Schema enthält unerwartete Elemente
LDX_TAD_ERR_NO_READ_ACCESS_FOR_FILE_C	keine Leserechte für Datei
LDX_TAD_ERR_NO_WRITE_ACCESS_FOR_FILE_C	keine Schreibrechte für Datei
LDX_TAD_ERR_TABLE_REC_TOO_BIG_C	Tabellen-Record zu groß, muss mit einer NCS-Response übertragbar sein

Tab. 4-36: TAD-Fehlercodes

4.10.5 DBD-Fehler

Die folgende Tabelle gibt die Fehlercodes wieder, die beim Zugriff auf Datenbank-Tabellen auftreten können.

Die im Abschnitt [XML-Fehler, Seite 232](#) aufgeführten XML-Fehler können ebenfalls gemeldet werden.

Fehlercode	Beschreibung
LDX_DBD_ERR_OK_C	kein Fehler
LDX_DBD_ERR_INTERNAL_ERROR_C	interner Fehler (s. Wera-Fehler)
LDX_DBD_ERR_SERVER_NOT_READY_C	Server nicht bereit (s. Wera-Fehler)
LDX_DBD_ERR_NAME2LONG_C	unzulässige Namenlänge
LDX_DBD_ERR_UNKNOWN_TABLETYPE_C	unbekannter Tabellentyp
LDX_DBD_ERR_TABLE_NOT_FOUND_C	Tabelle nicht vorhanden
LDX_DBD_ERR_TABLE_EXISTS_C	Tabelle existiert bereits
LDX_DBD_ERR_BAD_DATASIZE_C	Datenpuffer zu klein
LDX_DBD_ERR_BAD_PATHSIZE_C	Path-Puffer zu klein
LDX_DBD_ERR_BAD_SEGMENTPATHSIZE_C	Segment-Path zu klein (intern)
LDX_DBD_ERR_SCHEMA_INIT_FAILED_C	Fehler beim Initialisieren des Schemas
LDX_DBD_ERR_BAD_NCS_MSG_C	ungültige NCS-Nachricht (nur von ..X-Funktionen benutzt) (intern)
LDX_DBD_ERR_NO_MORE_MOD_EVENTS_C	maximale Anzahl der ModEvents erreicht, keine weiteren ModEvents können angelegt werden
LDX_DBD_ERR_UNKNOWN_MOD_EVENT_C	unbekannte Event-Id
LDX_DBD_ERR_MOD_EVENT_ALREADY_REQ_C	ModEvent wurde bereits angefordert
LDX_DBD_ERR_MOD_EVENT_DELETED_C	Fehlercode einer Event-Anforderung nach Löschen des zugehörigen ModEvents (kein Fehler)
LDX_DBD_ERR_REC_ALREADY_LOCKED_C	Datensatz ist z.Zt. von einem Client gelockt
LDX_DBD_ERR_TABLE_REC_TOO_BIG_C	Tabellen-Record zu groß, muss mit einer NCS-Response übertragbar sein

Tab. 4-37: DBD-Fehlercodes

4.10.6 CFG-Fehler

Die folgende Tabelle gibt die Fehlercodes wieder, die beim Zugriff auf Konfigurationsdaten auftreten können.

Die im Abschnitt [XML-Fehler, Seite 232](#) aufgeführten XML-Fehler können ebenfalls gemeldet werden.

Items

Fehlercode	Beschreibung
LDX_CFG_ERR_OK_C	kein Fehler
LDX_CFG_ERR_INTERNAL_ERROR_C	interner Fehler (s. Wera-Fehler)
LDX_CFG_ERR_SERVER_NOT_READY_C	Server nicht bereit (s. Wera-Fehler)
LDX_CFG_ERR_NAME2LONG_C	unzulässige Namenlänge
LDX_CFG_ERR_BAD_DATASIZE_C	Datenpuffer zu klein
LDX_CFG_ERR_BAD_PATHSIZE_C	Path-Puffer zu klein
LDX_CFG_ERR_BAD_SEGMENTPATHSIZE_C	Segment-Path zu klein (intern)
LDX_CFG_ERR_SCHEMA_INIT_FAILED_C	Fehler beim Initialisieren des Schemas
LDX_CFG_ERR_BAD_NCS_MSG_C	ungültige NCS-Nachricht (nur von ..X-Funktionen benutzt) (intern)
LDX_CFG_ERR_NO_MORE_MOD_EVENTS_C	maximale Anzahl der ModEvents erreicht, keine weiteren ModEvents können angelegt werden
LDX_CFG_ERR_UNKNOWN_MOD_EVENT_C	unbekannte Event-Id
LDX_CFG_ERR_MOD_EVENT_ALREADY_REQ_C	ModEvent wurde bereits angefordert
LDX_CFG_ERR_MOD_EVENT_DELETED_C	Fehlercode einer Event-Anforderung nach Löschen des zugehörigen ModEvents (kein Fehler)
LDX_CFG_ERR_INVALID_MODE_C	ungültiger Mode
LDX_CFG_ERR_INVALID_ROOT_ELEM_C	ungültiges Root-Element
LDX_CFG_ERR_CORRUPT_DATA_GROUP_C	Datengruppe nicht korrekt initialisiert
LDX_CFG_ERR_FILE_ACCESS_ERROR_C	Fehler beim Zugriff auf Datei
LDX_CFG_ERR_INVALID_USER_ID_C	ungültige Benutzerkennung
LDX_CFG_ERR_NO_MCA_INFO_C	Element enthält keine MACODA-Informationen im Schema
LDX_CFG_ERR_INVALID_MCA_INFO_C	Element enthält ungültige MACODA-Informationen im Schema
LDX_CFG_ERR_NO_CONVERSION_CLASS_DEFINED_C	keine MACODA-CFG- Abbildungsklasse für Konfigurationsdatenelement definiert
LDX_CFG_ERR_CONVERSION_CLASS_NOT_VALID_C	keine gültige MACODA-CFG- Abbildungsklasse für Konfigurationsdatenelement definiert
LDX_CFG_ERR_NO_CONVERSION_FUNC_DEFINED_C	keine MACODA-CFG- Abbildungsfunktion für Konfigurationsdatenelement definiert

Fehlercode	Beschreibung
LDX_CFG_ERR_CONVERSION_CLASS_1_NOT_ALLOWED_C	MACODA-CFG-Abbildungsklasse 1 (keine Abbildung) nicht erlaubt für Konfigurationsdatenelement
LDX_CFG_ERR_MACODA_ID_NOT_IN_CONVERSION_TABLE_C	kein Eintrag in MACODA-CFG-Umsetzungstabelle zur vorgegebenen MACODA-Blocknummer
LDX_CFG_ERR_OPEN_MACODA_BLOCK_FAILED_C	Öffnen eines MACODA-Blocks gescheitert
LDX_CFG_ERR_XMLSTRING_EMPTY_C	übergebener XML-String ist leer
LDX_CFG_ERR_NO_START_TAG_IN_XMLSTRING_C	übergebener XML-String hat kein Start-TAG
LDX_CFG_ERR_NO_END_TAG_IN_XMLSTRING_C	übergebener XML-String hat kein End-TAG
LDX_CFG_ERR_READ_MACODA_PARAMETER_FAILED_C	Lesen eines MACODA-Blocks gescheitert
LDX_CFG_ERR_WRITE_MACODA_PARAMETER_FAILED_C	Schreiben eines MACODA-Blocks gescheitert
LDX_CFG_ERR_CLOSE_MACODA_BLOCK_FAILED_C	Schließen eines MACODA-Blocks gescheitert
LDX_CFG_ERR_XMLSTRING_BUFFER_TOO_SMALL_C	Puffer für XML-Daten reicht nicht aus
LDX_CFG_ERR_NO_CONVERSION_PATH_DEFINED_C	kein Abbildungspfad in Umsetzungstabelle definiert für Abbildung eines MACODA-Parameters in die CFGDaten
LDX_CFG_ERR_INVALID_CONVERSION_PATH_C	fehlerhafter Abbildungspfad in Umsetzungstabelle definiert für Abbildung eines MACODA-Parameters in die CFGDaten
LDX_CFG_ERR_UNEXPECTED_XPATH_ELEMENT_C	übergebener XPath oder ein XPath in der MACODACFG- Umsetzungstabelle enthält unerwarteten Eintrag

Tab. 4-38: CFG-Fehlercodes

4.10.7 SD-Fehler

Die folgende Tabelle gibt die Fehlercodes wieder, die beim Zugriff auf Systemdaten auftreten können.

Die im Abschnitt [XML-Fehler, Seite 232](#) aufgeführten XML-Fehler können ebenfalls vom Tabellen-Server gemeldet werden.

Fehlercode	Beschreibung
LDX_SD_ERR_OK_C	kein Fehler
LDX_SD_ERR_INTERNAL_ERROR_C	interner Fehler (s. Wera-Fehler)
LDX_SD_ERR_SERVER_NOT_READY_C	Server nicht bereit (s. Wera-Fehler)
LDX_SD_ERR_NAME2LONG_C	unzulässige Namenlänge

Fehlercode	Beschreibung
LDX_SD_ERR_BAD_DATASIZE_C	Datenpuffer zu klein
LDX_SD_ERR_BAD_PATHSIZE_C	Path-Puffer zu klein
LDX_SD_ERR_BAD_SEGMENTPATH-SIZE_C	Segment-Path zu klein (intern)
LDX_SD_ERR_SCHEMA_INIT_FAILED_C	Fehler beim Initialisieren des Schemas
LDX_SD_ERR_BAD_NCS_MSG_C	ungültige NCS-Nachricht (nur von .._X-Funktionen benutzt) (intern)
LDX_SD_ERR_NO_MORE_MOD_EVENTS_C	maximale Anzahl der ModEvents erreicht, keine weiteren ModEvents können angelegt werden
LDX_SD_ERR_UNKNOWN_MOD_EVENT_C	unbekannte Event-Id
LDX_SD_ERR_MOD_EVENT_ALREADY_REQ_C	ModEvent wurde bereits angefordert
LDX_SD_ERR_MOD_EVENT_DELETED_C	Fehlercode einer Event-Anforderung nach Löschen des zugehörigen ModEvents (kein Fehler)
LDX_SD_ERR_INVALID_MODE_C	ungültiger Mode
LDX_SD_ERR_INVALID_ROOT_ELEM_C	ungültiges Root-Element
LDX_SD_ERR_CORRUPT_DATA_GROUP_C	Datengruppe nicht korrekt initialisiert
LDX_SD_ERR_FILE_ACCESS_ERROR_C	Fehler beim Zugriff auf eine Datei
LDX_SD_ERR_INVALID_USER_ID_C	ungültige Benutzerkennung

Tab. 4-39: SD-Fehlercodes

4.10.8 XML-Fehler

Die folgende Tabelle gibt XML-Fehlercodes wieder, die bei Zugriff auf dateibasierte und datenbankbasierte Tabellen verwendet werden.

Für XML-Fehler wird ein eigener Enumerator zur Verfügung gestellt. Außerdem sind die XML-Fehler in den Enumeratoren für dateibasierte Tabellen und für datenbankbasierte Tabellen enthalten.

Fehlercode	Beschreibung
LDX_XML_SERVICEOK_C	kein Fehler
LDX_XML_ERR_NOT_IMPLEMENTED_C	Funktion noch nicht implementiert
LDX_XML_ERR_INTERNAL_ERROR_C	interner Fehler (s. Wera-Fehler)
LDX_XML_ERR_SCHEMA_PARSE_FAILED_C	Fehler beim Parsen der Schema-Datei (s. Wera-Fehler)
LDX_XML_ERR_UNKNOWN_LIBXML_ERROR_C	unbekannter Fehler aus libxml

Fehlercode	Beschreibung
LDX_XML_ERR_FOPEN_FAILED_C	Fehler beim Öffnen der Datei
LDX_XML_ERR_MAX_OCCURS_EXCEEDED_C	Parserfehler: mehr Elemente vorhanden als laut maxOccurs zulässig
LDX_XML_ERR_BAD_ELEM_CONTENT_C	Parserfehler: Elementinhalt entspricht nicht dem Datentyp laut Schema
LDX_XML_ERR_PATH_SYNTAX_ERROR_C	Syntax-Fehler in Path-String
LDX_XML_ERR_PATH_CONFLICT_C	SegmentPath liegt nicht im RootPath (intern)
LDX_XML_ERR_PATH_ELEM_POS_INVALID_C	Element-Position laut Schema unzulässig (Path)
LDX_XML_ERR_PATH_ELEM_CREATE_FAILED_C	über Path spezifiziertes Element kann nicht angelegt werden
LDX_XML_ERR_PATH_ELEM_DELETE_FAILED_C	über Path spezifiziertes Element kann nicht gelöscht werden
LDX_XML_ERR_PATH_ELEM_NOT_EXISTING_C	über Path spezifiziertes Element ist nicht in den XML-Daten enthalten
LDX_XML_ERR_PATH_ELEM_NOT_DECLARED_C	über Path spezifiziertes Element ist nicht im Schema deklariert
LDX_XML_ERR_PATH_ELEM_MISMATCH_C	letztes Element im Path stimmt nicht mit erstem Element in Data überein
LDX_XML_ERR_ELEM_NOT_DECLARED_C	Element aus XML-Datei oder XML-String nicht im Schema deklariert
LDX_XML_ERR_INVALID_DATA_SIZE_C	Datengröße passt nicht zu spezifiziertem Element
LDX_XML_ERR_INVALID_TYPE_ID_C	Datentyp passt nicht zu spezifiziertem Element
LDX_XML_ERR_TOO_MUCH_DATA_REQUESTED_C	zu viele binäre Daten angefordert
LDX_XML_LIBERR_INTERNAL_ERROR_C	
LDX_XML_LIBERR_NO_MEMORY_C	
LDX_XML_LIBERR_DOCUMENT_START_C	
LDX_XML_LIBERR_DOCUMENT_EMPTY_C	
LDX_XML_LIBERR_DOCUMENT_END_C	
LDX_XML_LIBERR_INVALID_HEX_CHARREF_C	
LDX_XML_LIBERR_INVALID_DEC_CHARREF_C	

Items

Fehlercode	Beschreibung
LDX_XML_LIBERR_INVALID_CHAR-REF_C	
LDX_XML_LIBERR_INVALID_CHAR_C	
LDX_XML_LIBERR_CHAR-REF_AT_EOF_C	
LDX_XML_LIBERR_CHAR-REF_IN_PROLOG_C	
LDX_XML_LIBERR_CHARREF_IN_EPI-LOG_C	
LDX_XML_LIBERR_CHAR-REF_IN_DTD_C	
LDX_XML_LIBERR_ENTITY-REF_IN_PROLOG_C	
LDX_XML_LIBERR_ENTITY-REF_IN_EPIOLOG_C	
LDX_XML_LIBERR_ENTITY-REF_IN_DTD_C	
LDX_XML_LIBERR_PREFER_AT_EOF_C	
LDX_XML_LIBERR_PREFER_IN_PRO-LOG_C	
LDX_XML_LIBERR_PREFER_IN_EPIO-LOG_C	
LDX_XML_LIBERR_PREFER_IN_INT_SUBSET_C	
LDX_XML_LIBERR_ENTITY-REF_NO_NAME_C	
LDX_XML_LIBERR_PREFER_NO_NAME_C	
LDX_XML_LIBERR_PREFER_SEMI-COL_MISSING_C	
LDX_XML_LIBERR_UNDECLA-RED_ENTITY_C	
LDX_XML_LIBWAR_UNDECLA-RED_ENTITY_C	
LDX_XML_LIBERR_UNPARSED_ENT-ITY_C	
LDX_XML_LIBERR_ENTITY_IS_EXTER-NAL_C	
LDX_XML_LIBERR_ENTITY_IS PARA-METER_C	
LDX_XML_LIBERR_UNKNOWN_ENCO-DING_C	

Fehlercode	Beschreibung
LDX_XML_LIBERR_UNSUPPOR- TED_ENCODING_C	
LDX_XML_LI- BERR_STRING_NOT_STARTED_C	
LDX_XML_LI- BERR_STRING_NOT_CLOSED_C	
LDX_XML_LI- BERR_NS_DECL_ERROR_C	
LDX_XML_LIBERR_ENT- ITY_NOT_STARTED_C	
LDX_XML_LIBERR_ENTITY_NOT_FI- NISHED_C	
LDX_XML_LIBERR_LT_IN_ATTRIBU- TE_C	
LDX_XML_LIBERR_ATTRIBU- TE_NOT_STARTED_C	
LDX_XML_LIBERR_ATTRIBU- TE_NOT_FINISHED_C	
LDX_XML_LIBERR_ATTRIBUTE_WI- THOUT_VALUE_C	
LDX_XML_LIBERR_ATTRIBUTE_REDE- FINED_C	
LDX_XML_LIBERR_LITE- RAL_NOT_STARTED_C	
LDX_XML_LIBERR_LITERAL_NOT_FI- NISHED_C	
LDX_XML_LIBERR_COM- MENT_NOT_FINISHED_C	
LDX_XML_LIBERR_PI_NOT_START- ED_C	
LDX_XML_LIBERR_PI_NOT_FINIS- HED_C	
LDX_XML_LIBERR_NOTATI- ON_NOT_STARTED_C	
LDX_XML_LIBERR_NOTATI- ON_NOT_FINISHED_C	
LDX_XML_LIBERR_ATT- LIST_NOT_STARTED_C	
LDX_XML_LIBERR_ATTLIST_NOT_FI- NISHED_C	
LDX_XML_LI- BERR_MIXED_NOT_STARTED_C	
LDX_XML_LIBERR_MIXED_NOT_FINIS- HED_C	

Items

Fehlercode	Beschreibung
LDX_XML_LIBERR_ELEMCONTENT_NOT_STARTED_C	
LDX_XML_LIBERR_ELEMCONTENT_NOT_FINISHED_C	
LDX_XML_LIBERR_XMLDECL_NOT_STARTED_C	
LDX_XML_LIBERR_XMLDECL_NOT_FINISHED_C	
LDX_XML_LIBERR_CONDITION_NOT_STARTED_C	
LDX_XML_LIBERR_CONDITION_NOT_FINISHED_C	
LDX_XML_LIBERR_EXT_SUBSET_NOT_FINISHED_C	
LDX_XML_LIBERR_DOCTYPE_NOT_FINISHED_C	
LDX_XML_LIBERR_MISPLACED_CDATA_END_C	
LDX_XML_LIBERR_CDATA_NOT_FINISHED_C	
LDX_XML_LIBERR_RESERVED_XML_NAME_C	
LDX_XML_LIBERR_SPACE_REQUIRED_C	
LDX_XML_LIBERR_SEPARATOR_REQUIRED_C	
LDX_XML_LIBERR_NMTOKEN_REQUIRED_C	
LDX_XML_LIBERR_NAME_REQUIRED_C	
LDX_XML_LIBERR_PCDATA_REQUIRED_C	
LDX_XML_LIBERR_URI_REQUIRED_C	
LDX_XML_LIBERR_PUBID_REQUIRED_C	
LDX_XML_LIBERR_LT_REQUIRED_C	
LDX_XML_LIBERR_GT_REQUIRED_C	
LDX_XML_LIBERR_LTSLASH_REQUIRED_C	
LDX_XML_LIBERR_EQUAL_REQUIRED_C	
LDX_XML_LIBERR_TAG_NAME_MISMATCH_C	

Fehlercode	Beschreibung
LDX_XML_LIBERR_TAG_NOT_FINISHED_C	
LDX_XML_LIBERR_STANDALONE_VALUE_C	
LDX_XML_LIBERR_ENCODING_NAME_C	
LDX_XML_LIBERR_HYPHEN_IN_COMMENT_C	
LDX_XML_LIBERR_INVALID_ENCODING_C	
LDX_XML_LIBERR_EXT_ENTITY_STANDALONE_C	
LDX_XML_LIBERR_CONDSEC_INVALID_C	
LDX_XML_LIBERR_VALUE_REQUIRED_C	
LDX_XML_LIBERR_NOT_WELL_BALANCED_C	
LDX_XML_LIBERR_EXTRA_CONTENT_C	
LDX_XML_LIBERR_ENTITY_CHAR_ERROR_C	
LDX_XML_LIBERR_ENTITY_PE_INTERNAL_C	
LDX_XML_LIBERR_ENTITY_LOOP_C	
LDX_XML_LIBERR_ENTITY_BOUNDARY_C	

Tab. 4-40: XML-Fehlercodes

4.10.9 TAR-Fehler

Die folgende Tabelle gibt die Fehlercodes wieder, die bei der Archivierung auftreten können.

Falls einer der Fehler auftritt, wird der Archivierungsvorgang abgeschlossen. Die Informationen im Log-File sind in diesem Fall ungültig.

Fehlercode	Beschreibung
LDX_TAR_ERR_OK_C	Archivierung bzw. Einlesen des Archivs erfolgreich abgeschlossen, weitere Informationen wurden protokolliert
LDX_TAR_ERR_NCS_C	NCS-Fehler
LDX_TAR_ERR_BAD_PARAMETERS_C	Parameter fehlerhaft
LDX_TAR_ERR_OPENING_ARCHIVE_C	Fehler beim Öffnen des Archivs
LDX_TAR_ERR_WRITING_ARCHIVE_C	Fehler beim Schreiben des Archivs
LDX_TAR_ERR_READING_ARCHIVE_C	Fehler beim Lesen des Archivs

Fehlercode	Beschreibung
LDX_TAR_ERR_CHDIR_C	Fehler beim Wechseln auf Zielverzeichnis
LDX_TAR_ERR_LOGFILE_C	Fehler beim Anlegen oder Schreiben der Log-Datei
LDX_TAR_ERR_MACODA_C	Fehler bei Bearbeitung der MACODA-Parameter
LDX_TAR_ERR_TABLES_C	Fehler bei Bearbeitung der Tabellen
LDX_TAR_ERR_CPLVARS_C	Fehler bei Bearbeitung permanenter CPL-Variablen
LDX_TAR_ERR_DATAMODULES_C	Fehler bei Bearbeitung permanenter Daten
LDX_TAR_ERR_MARKERS_C	Fehler bei Bearbeitung permanenter Daten
LDX_TAR_ERR_FORMAT_C	Format des Archivs fehlerhaft
LDX_TAR_ERR_USRFEP_C	Fehler bei Programmierung des Benutzer-FEPROM
LDX_TAR_ERR_ENOMEM_C	zu wenig Speicher
LDX_TAR_ERR_ALREADY_ACTIVE_C	Archivierung bereits aktiv
LDX_TAR_ERR_NOLOGIN_C	Login beim Fileserver fehlerhaft
LDX_TAR_ERR_PLSTATE_RUN_C	SPS nicht im Zustand STOP

Tab. 4-41: TAR-Fehlercodes bei der Archivierung

Während der Archivierung wird ein Log-File geschrieben, das alle gesicherten bzw. wiederhergestellten Dateien und eventuelle Fehler enthält.

Die folgende Tabelle enthält die im Log-File möglichen Fehlercodes:

Fehlercode	Beschreibung
LDX_TAR_LOG_OK_C	Datei ohne Fehler bearbeitet
LDX_TAR_LOG_NOT_YET_PROCESSED_C	Datei (noch) nicht bearbeitet
LDX_TAR_LOG_STAT_FAILED_C	Fehler bei Aufruf der Funktion "stat"
LDX_TAR_LOG_SKIPPED_C	Datei bzw. Verzeichnis nicht bearbeitet (kein Fehler!)
LDX_TAR_LOG_PATH_NOT_OPENED_C	Verzeichnis konnte nicht geöffnet werden
LDX_TAR_LOG_READING_C	Fehler beim Lesen der Datei
LDX_TAR_LOG_EACCESS_C	Zugriff auf Datei nicht möglich
LDX_TAR_LOG_EEXIST_C	Datei existiert bereits
LDX_TAR_LOG_ENOENT_C	Datei nicht vorhanden
LDX_TAR_LOG_ENOMEM_C	kein Speicher
LDX_TAR_LOG_EROFS_C	Read-Only File-System
LDX_TAR_LOG_MODE_DIFF_C	Schutz unterschiedlich

Fehlercode	Beschreibung
LDX_TAR_LOG_SIZE_DIFF_C	Größe unterschiedlich
LDX_TAR_LOG_MTIME_DIFF_C	Datum unterschiedlich
LDX_TAR_LOG_OTHER_DIFF_C	Dateidaten unterschiedlich
LDX_TAR_LOG_FILE_DATA_DIFF_C	andere Unterschiede
LDX_TAR_LOG_TIME_MODE_C	Datum/Schutz nicht wiederhergestellt
LDX_TAR_LOG_MKDIR_FAILED_C	Fehler bei Anlegen eines Verzeichnisses
LDX_TAR_LOG_ERR_UNKNOWN_C	unbekannter Fehler
LDX_TAR_LOG_LOADING_MACODA_C	Fehler bei Übernahme MACODA-Daten
LDX_TAR_LOG_LOADING_TABLE_C	Fehler bei Übernahme Werkzeugtabelle
LDX_TAR_LOG_CORRUPTED_C	Datei zerstört
LDX_TAR_LOG_WRITE_ENABLED_C	Schreibschutz aufgehoben
LDX_TAR_LOG_DATAMODULES_C	Fehler bei Übernahme remanenter SPS-Daten
LDX_TAR_LOG_MARKERS_C	Fehler bei Übernahme remanenter SPS-Daten
LDX_TAR_LOG_CPLVARS_C	Fehler bei Übernahme der CPL-Variablen
LDX_TAR_LOG_RAMFS_NOT-FOUND_C	RAM-File-System nicht im Archiv
LDX_TAR_LOG_USRFEPFS_NOT-FOUND_C	usrfep-File-System nicht im Archiv
LDX_TAR_LOG_MACODA_NOT-FOUND_C	MACODA-Daten nicht im Archiv
LDX_TAR_LOG_TABLE_NOTFOUND_C	Werkzeugtabelle nicht im Archiv
LDX_TAR_LOG_DATAMODULES_NOT-FOUND_C	remanente SPS-Daten nicht im Archiv
LDX_TAR_LOG_MARKERS_NOT-FOUND_C	remanente SPS-Daten nicht im Archiv
LDX_TAR_LOG_CPLVARS_NOT-FOUND_C	CPL-Variablen nicht im Archiv
LDX_TAR_LOG_SYSTEMDATA_NOT-FOUND_C	Systemdaten nicht im Archiv
LDX_TAR_LOG_PLSTATE_RUN_C	SPS nicht im Zustand STOP
LDX_TAR_LOG_SYSTEMDATA_C	Fehler bei Übernahme der Systemdaten

Tab. 4-42: TAR-Fehlercodes im LOG-File

4.10.10 Unlock-Fehler

Die folgende Tabelle gibt die Fehlercodes wieder, die beim Freischalten der Steuerung auftreten können.

Fehlercode	Beschreibung
LDX_Unlock_Ok_C	Kein Fehler - Freischaltcode ist gültig
LDX_Unlock_TargetNotSupported_C	Die Funktionalität "CtrlLocking" wird von diesem Target nicht unterstützt
LDX_Unlock_NotEnabled_C	Die Funktionalität "CtrlLocking" ist an dieser Steuerung nicht aktiviert
LDX_Unlock_WrongFormat_C	Der UnlockCode hat das falsche Format. Ggf. fehlen Zeichen oder es welche zu viel
LDX_Unlock_Corrupted_C	Der UnlockCode ist beschädigt. Ggf. wurden einzelne Zeichen falsch übermittelt. Groß- und Kleinschreibung sind relevant
LDX_Unlock_Invalid1_C	Der UnlockCode ist nicht gültig. Ggf. wurden die Daten von der Steuerung zum OEM falsch übermittelt
LDX_Unlock_Invalid2_C	Der UnlockCode ist nicht gültig. Wahrscheinlich wurde dieser UnlockCode bereits benutzt. Ggf. wurden die Daten von der Steuerung zum OEM falsch übermittelt
LDX_Unlock_Outdated_C	Der UnlockCode ist zu alt. Nachdem die UnlockDatenvon der Steuerung an den OEM übertragen wurden, muss ein generierten UnlockCode innerhalb von 48 benutzt werden. Sonst verfällt er automatisch
LDX_Unlock_UnknownVersion_C	Die Version des UnlockCodes ist der Steuerung nicht bekannt. Ggf. ist die Firmware zu alt
LDX_Unlock_WrongMachine_C	Der Freischaltcode passt nicht zu dieser Maschine. Er ist nur an der Steuerung gültig, vor der die UnlockData stammt
LDX_Unlock_NoOEMKey_C	Der Schlüssel des OEM ist nicht in der Steuerung verfügbar. Die entsprechende Datei wurde nicht gefunden oder war defekt

Tab. 4-43: Unlock-Fehlercodes

5 Service und Support

Für Ihre schnelle und optimale Unterstützung verfügen wir über ein dichtes weltweites Servicenetz. Unsere Experten stehen Ihnen mit Rat und Tat zur Seite. Sie erreichen uns täglich **rund um die Uhr – auch an Wochenenden und Feiertagen**.

Service Deutschland

Unser technologieorientiertes Competence Center in Lohr deckt alle Belange rund um den Service für elektrische Antriebe und Steuerungen ab.

Sie erreichen unsere **Service-Hotline** und unseren **Service-Helpdesk** unter:

Telefon: **+49 9352 40 5060**
Fax: **+49 9352 18 4941**
E-Mail: service.svc@boschrexroth.de
Internet: <http://www.boschrexroth.com>

Auf unseren Internetseiten finden Sie ergänzende Hinweise zu Service, Reparatur (z. B. Anlieferadressen) und Training.

Service weltweit

Außerhalb Deutschlands nehmen Sie bitte zuerst Kontakt mit Ihrem Ansprechpartner auf. Die Hotline-Rufnummern entnehmen Sie bitte den Vertriebsadressen im Internet.

Vorbereitung der Informationen

Wir können Ihnen schnell und effizient helfen, wenn Sie folgende Informationen bereithalten:

- Eine detaillierte Beschreibung der Störung und der Umstände
- Angaben auf dem Typenschild der betreffenden Produkte, insbesondere Typenschlüssel und Seriennummern
- Ihre Kontaktdaten (Telefon-, Faxnummer und E-Mail-Adresse)

Index

A

Abgleichfunktionen.....	178
Achskorrekturen.....	167
Achspositionen.....	157
Allgemeine Items	
AutoItemAccess.....	13
CNF.AddComponent.....	14
CNF.RemoveComponent.....	15
DTY.....	17
ListOfItems.....	18
LOI.....	18
SCL.....	18
SDL.....	19
Anregungen.....	6
AutoItemAccess.....	13

B

Beschwerde.....	6
Bestimmungsgemäßer Gebrauch	
Anwendungsbereiche.....	12
Einleitung.....	11
Einsatzfälle.....	12

C

CFG-Fehler.....	229
CNF.AddComponent.....	14
CNF.CMD.....	15
CNF.RemoveComponent.....	15
ComPortList.....	17

D

DBD-Fehler.....	228
DeviceAddress.....	44
DeviceAddressList.....	44
DeviceTypeTest2.....	13
Diagnose.....	46, 48
Diagnose Items	
System.MessageStatus.....	53
System.MessageStore.....	48
System.MessageStoreEx.....	51
Dialog Items	
System.ErrorStatus.....	54
System.InfoStatus.....	55
System.MessageReset.....	57
System.SetupStatus.....	57
System.StartupStatus.....	56
System.WarningStatus.....	55
DTY.....	17

E

Eingriff.....	68
---------------	----

F

Feedback.....	6
---------------	---

Fehler-Codes.....	30
Fehlermeldungen.....	223
Feldbus.....	69
Firmware.....	19, 65
Firmware - Download	
Firmware.CheckDownloadRequirements.....	20
Firmware.ComponentFirmware.....	21
Firmware.DeviceFirmware.....	22
Firmware.Download.....	22
Firmware.ModuleFirmware.....	25
Firmware.Reboot.....	25
Firmware.ResumeConnection.....	26
Firmware.Status.....	26
Firmware.StatusReset.....	28
Firmware.SubmoduleFirmware.....	28
NC.LanguagesDownload.....	29
Firmware Items.....	20
Firmware.CheckDownloadRequirements.....	20
Firmware.ComponentFirmware.....	21
Firmware.DeviceFirmware.....	22
Firmware.Download.....	22
Firmware.ModuleFirmware.....	25
Firmware.Reboot.....	25
Firmware.ResumeConnection.....	26
Firmware.Status.....	26
Firmware.StatusReset.....	28
Firmware.SubmoduleFirmware.....	28

G

GDL.....	18
Geräteabhängige Items	
DeviceTypeTest2.....	13
Globale Kommandos.....	207

H

Helpdesk.....	241
Hotline.....	241

I

IndraLogic.....	64
IndraLogic-Item	
Plc.FeldBus.....	69
Plc.FirmwareConnection.....	65
Plc.FirmwareReboot.....	65
Plc.FirmwareStatus.....	66
Plc.FirmwareVersion.....	66
Plc.ModDiagnose.....	66
Plc.ModStatus.....	67
Plc.PVA.....	69
Plc.PVArray.....	69
Plc.PVBS.....	70
Plc.PVF.....	71
Plc.PVL.....	71
Plc.PVLK.....	72
Plc.PVT.....	73

Index

Plc.StartStop.....	68	NC.XmlTableEvent.....	107
Plc.State2.....	67	NC - global (Datenbankbasierte Tabellen)	
Info.....	66	NC.DbTableDataRec_U.....	107
Items.....	13	NC.DbTableDataRecLocked_U.....	108
K		NC.DbTableEvent.....	109
Kanalabhängige Kommandos.....	210	NC.DbTableLoad.....	110
Konfiguration		NC.DbTableQuery_U.....	111
CNF.CMD.....	15	NC.DbTableSave.....	113
GDL.....	18	NC - global (Maschinenparameter)	
Kritik.....	6	NC.CfgData_U.....	114
Kundenfeedback.....	6	NC.CfgDataBackup.....	114
		NC.CfgDataEvent.....	115
		NC.CfgDataLoad.....	116
		NC.CfgDataSave.....	116
		NC.MacodaParameter.....	113
L		NC - global (Systemdaten)	
ListOfItems.....	18	NC.SystemData_U.....	118
Logbuch.....	46, 58	NC.SystemDataEvent.....	120
Logbuch Items		NC.SystemDataLoad.....	121
System.LogbookClear.....	63	NC.SystemDataSave.....	121
System.LogbookDevices.....	64	NC.SystemDataWatch_U.....	122
System.LogbookStart.....	64	NC - global (Systemfunktionen)	
System.LogbookStore.....	58	System.Archive.....	123
System.LogbookStoreEx.....	61	System.ArchiveLog.....	124
		System.ArchiveWin.....	126
		System.ArchiveWinLog.....	127
		System.HardwareDescription.....	129
		System.HardwareDescription2.....	129
		System.Language.....	130
		System.MainReleases.....	130
		System.PcReleases.....	131
		System.ProVi.Active.....	136
		System.ProVi.KaFreeze.....	131
		System.ProVi.KaOnline.....	134
		System.Sfc.Active.....	143
		System.Sfc.Execute.....	143
		System.Sfc.KaExecute.....	144
		System.Sfc.KaFreeze.....	136
		System.Sfc.KaOnline.....	139
		System.Sfc.Mode.....	142
		System.Sfc.StepInfo.....	142
		System.SubsystemReleases.....	146
		System.SupportedLanguages.....	147
		System.SystemTime.....	147
		NC - kanalabhängig.....	149
		NC - kanalabhängig (Abgleichdaten)	
		NC.Chan.AuxActual.....	178
		NC.Chan.AuxAdjustment.....	178
		NC.Chan.AuxReference.....	178
		NC.Chan.FctActual.....	179
		NC.Chan.FctActualSorted.....	179
		NC.Chan.FctAdjustment.....	180
		NC.Chan.FctAdjustmentSorted.....	180
		NC.Chan.FctReference.....	180
		NC.Chan.FctReferenceSorted.....	181
		NC.Chan.FunctionBaseConfiguration.....	154
		NC - kanalabhängig (Achsen)	
		NC.Chan.AxisPosAcs.....	157
M			
MACODA-Fehler.....	227		
N			
NC - global.....	74		
NC - global (Allg. Informationen)			
NC.AxisNamesPhys.....	74		
NC.BitInterface.....	74		
NC.CheckFile.....	78		
NC.ConnectionQuality.....	75		
NC.CpuUsage2.....	76		
NC.DryRunMode.....	77		
NC.Filesystem.....	77		
NC.FIS.....	79		
NC.MtxCommand.....	79		
NC.NofChannels.....	81		
NC.ProgramChanInfo.....	91		
NC.ReloadExportUs.....	92		
NC.WaitingStates.....	93		
NC - global (Antriebe)			
NC.OscilloscopeConfiguration.....	94		
NC.SercosPhase2.....	95		
NC.SPA1.....	96		
NC.SPH1.....	99		
NC - global (CPL-Daten)			
NC.CplPermVariable.....	100		
NC.CplPermVariableList.....	101		
NC.CplPermVariableWatch.....	102		
NC - global (Dateibasierte Tabellen)			
NC.XmlRootElementInfo.....	103		
NC.XmlRootElementList.....	103		
NC.XmlSchemaList.....	104		
NC.XmlTable_U.....	104		
NC.XmlTableCreate.....	106		

NC.Chan.AxisPosAcsCoupl.....	158	NC.Chan.ModalNcFunctions.....	194
NC.Chan.AxisPosAcsDrv.....	159	NC.Chan.ModalTransformationFunctions....	194
NC.Chan.AxisPosAcsDrvAdj.....	159	NC.Chan.ModeOfOperation.....	194
NC.Chan.AxisPosBcs.....	160	NC.Chan.OptionalStop.....	196
NC.Chan.AxisPosBcsDrv.....	160	NC.Chan.ProgramPosition.....	196
NC.Chan.AxisPosDistAcs.....	161	NC.Chan.ProgramWindow.....	197
NC.Chan.AxisPosDistMcs.....	162	NC.Chan.ReentryChain.....	198
NC.Chan.AxisPosDistWcs.....	162	NC.Chan.ReentryDelete.....	198
NC.Chan.AxisPosEndWcs.....	163	NC.Chan.ReentryEvent.....	198
NC.Chan.AxisPosLag.....	163	NC.Chan.ReturnToContour.....	199
NC.Chan.AxisPosMcs.....	164	NC.Chan.ReturnToPathStrategy.....	199
NC.Chan.AxisPosProg.....	165	NC.Chan.SkipBlockMode.....	200
NC.Chan.AxisPosWcs.....	165	NC.Chan.UpCallFromNcBlock.....	200
NC.Chan.AxisPosWcsDrv.....	166	NC - kanalabhängig (Spindeldaten)	
NC - kanalabhängig (Allg. Informationen)		NC.Chan.ActSpindleAutoGears.....	181
NC.Chan.BalancedConditionState.....	149	NC.Chan.ActSpindleGears.....	182
NC.Chan.ChanHidden.....	149	NC.Chan.ActSpindleMovements.....	183
NC.Chan.ChanState.....	150	NC.Chan.ActSpindleOverride.....	183
NC.Chan.DryRunFeed.....	152	NC.Chan.ActSpindlePosition.....	184
NC.Chan.DryRunMode.....	153	NC.Chan.ActSpindlePower.....	184
NC.Chan.DryRunRapid.....	153	NC.Chan.ActSpindleSpeed.....	185
NC.Chan.ProgramLinkInfo.....	155	NC.Chan.ActSpindleSpeedDrv.....	185
NC.Chan.WaitingStates.....	156	NC.Chan.ActSpindleSwitchGears.....	186
NC - kanalabhängig (Korrekturdaten)		NC.Chan.ActSpindleTorque.....	186
NC.Chan.ActAxisOffsetTable.....	167	NC.Chan.ProgrammedSpindleSpeed.....	187
NC.Chan.ActDCorrectionNmb.....	167	NC - kanalabhängig (Vorschub- u. Drehzahlwerte)	
NC.Chan.ActDCorrectionTable.....	167	NC.Chan.ActAxisAccelerationPeak.....	203
NC.Chan.ActDCorrectionValues.....	168	NC.Chan.ActAxisVelocity.....	201
NC.Chan.ActOnlineCorrValue.....	168	NC.Chan.ActAxisVelocityMax.....	201, 202
NC.Chan.ActPlacementBcrCorr.....	169	NC.Chan.ActFeedOverride.....	204
NC.Chan.ActPlacementData.....	169	NC.Chan.ActFeedrate.....	204
NC.Chan.ActPlacementSumData.....	170	NC.Chan.ActRotAxisProgrammedSpeed....	205
NC.Chan.ActPlacementTable.....	170	NC.Chan.ActRotAxisSpeed.....	205
NC.Chan.ActProgCoordATrans.....	171	NC.Chan.ProgrammedFeedrate.....	206
NC.Chan.ActProgCoordTrans.....	171	NC.Chan.ProgrammedFeedrateIndex.....	206
NC.Chan.ActProgCoordTransSum.....	172	NC-Kommandos.....	207
NC.Chan.ActToolCorrectionValues.....	172	NC-Kommandos (global)	
NC.Chan.ActZeroOffsetSummary.....	172	NC.Cmd.DryRunMode.....	207
NC.Chan.ActZeroOffsetValues.....	173	NC.Cmd.MessageStoreClr.....	208
NC.Chan.AxisShift.....	174	NC.Cmd.Shutdown.....	209
NC.Chan.CompCrossData.....	174	NC.Cmd.Unlock.....	210
NC.Chan.CompLeadScrewErrData.....	175	NC.OsciUtil.....	82
NC.Chan.CompOffsetData.....	176	NC.PlcData.....	89
NC.Chan.CompTempData.....	177	System.Cmd.Reset.....	210
NC - kanalabhängig (Prozessdaten)		System.Cmd.Restart.....	210
NC.Chan.ActCallChain.....	187	NC-Kommandos (kanalabhängig)	
NC.Chan.ActMainProgram.....	187	NC.Chan.Cmd.MDI.....	210
NC.Chan.ActNcBlock.....	188	NC.Chan.Cmd.MidProgramStartup.....	213
NC.Chan.ActSubProgram.....	188	NC.Chan.Cmd.ModeOfOperation.....	217
NC.Chan.AxisMirror.....	188	NC.Chan.Cmd.OptionalStop.....	218
NC.Chan.AxisPoleSet.....	189	NC.Chan.Cmd.Reset.....	218
NC.Chan.AxisRotate.....	190	NC.Chan.Cmd.ReturnToPath.....	219
NC.Chan.AxisScale.....	190	NC.Chan.Cmd.ReturnToPathStrategy.....	219
NC.Chan.EqualizeState.....	191	NC.Chan.Cmd.SearchBlock.....	220
NC.Chan.MessageWindow.....	192	NC.Chan.Cmd.SelectProgram.....	221
NC.Chan.ModalAuxFunctions.....	192	NC.Chan.Cmd.SkipBlockMode.....	223
NC.Chan.ModalAxisOffsetFunctions.....	193	NC.AxisNamesPhys.....	74
NC.Chan.ModalFunction.....	193	NC.BitInterface.....	74

Index

NC.CfgData_U.....	114	NC.Chan.AxisPosMcs.....	164
NC.CfgDataBackup.....	114	NC.Chan.AxisPosProg.....	165
NC.CfgDataEvent.....	115	NC.Chan.AxisPosWcs.....	165
NC.CfgDataLoad.....	116	NC.Chan.AxisPosWcsDrv.....	166
NC.CfgDataSave.....	116	NC.Chan.AxisRotate.....	190
NC.Chan.ActAxisAccelerationPeak.....	203	NC.Chan.AxisScale.....	190
NC.Chan.ActAxisOffsetTable.....	167	NC.Chan.AxisShift.....	174
NC.Chan.ActAxisVelocity.....	201	NC.Chan.BalancedConditionState.....	149
NC.Chan.ActAxisVelocityMax.....	201	NC.Chan.ChanHidden.....	149
NC.Chan.ActAxisVelocityPeak.....	202	NC.Chan.ChanState.....	150
NC.Chan.ActCallChain.....	187	NC.Chan.Cmd.MDI.....	210
NC.Chan.ActDCorrectionNmb.....	167	NC.Chan.Cmd.MidProgramStartup.....	213
NC.Chan.ActDCorrectionTable.....	167	NC.Chan.Cmd.ModeOfOperation.....	217
NC.Chan.ActDCorrectionValues.....	168	NC.Chan.Cmd.OptionalStop.....	218
NC.Chan.ActFeedOverride.....	204	NC.Chan.Cmd.Reset.....	218
NC.Chan.ActFeedrate.....	204	NC.Chan.Cmd.ReturnToPath.....	219
NC.Chan.ActMainProgram.....	187	NC.Chan.Cmd.ReturnToPathStrategy.....	219
NC.Chan.ActNcBlock.....	188	NC.Chan.Cmd.SearchBlock.....	220
NC.Chan.ActOnlineCorrValue.....	168	NC.Chan.Cmd.SelectProgram.....	221
NC.Chan.ActPlacementBcrCorr.....	169	NC.Chan.Cmd.SkipBlockMode.....	223
NC.Chan.ActPlacementData.....	169	NC.Chan.CompCrossData.....	174
NC.Chan.ActPlacementSumData.....	170	NC.Chan.CompLeadScrewErrData.....	175
NC.Chan.ActPlacementTable.....	170	NC.Chan.CompOffsetData.....	176
NC.Chan.ActProgCoordATrans.....	171	NC.Chan.CompTempData.....	177
NC.Chan.ActProgCoordTrans.....	171	NC.Chan.DryRunFeed.....	152
NC.Chan.ActProgCoordTransSum.....	172	NC.Chan.DryRunMode.....	153
NC.Chan.ActRotAxisProgrammedSpeed.....	205	NC.Chan.DryRunRapid.....	153
NC.Chan.ActRotAxisSpeed.....	205	NC.Chan.EqualizeState.....	191
NC.Chan.ActSpindleAutoGears.....	181	NC.Chan.FctActual.....	179
NC.Chan.ActSpindleGears.....	182	NC.Chan.FctActualSorted.....	179
NC.Chan.ActSpindleMovements.....	183	NC.Chan.FctAdjustment.....	180
NC.Chan.ActSpindleOverride.....	183	NC.Chan.FctAdjustmentSorted.....	180
NC.Chan.ActSpindlePosition.....	184	NC.Chan.FctReference.....	180
NC.Chan.ActSpindlePower.....	184	NC.Chan.FctReferenceSorted.....	181
NC.Chan.ActSpindleSpeed.....	185	NC.Chan.FunctionBaseConfiguration.....	154
NC.Chan.ActSpindleSpeedDrv.....	185	NC.Chan.MessageWindow.....	192
NC.Chan.ActSpindleSwitchGears.....	186	NC.Chan.ModalAuxFunctions.....	192
NC.Chan.ActSpindleTorque.....	186	NC.Chan.ModalAxisOffsetFunctions.....	193
NC.Chan.ActSubProgram.....	188	NC.Chan.ModalFunction.....	193
NC.Chan.ActToolCorrectionValues.....	172	NC.Chan.ModalNcFunctions.....	194
NC.Chan.ActZeroOffsetSummary.....	172	NC.Chan.ModalTransformationFunctions.....	194
NC.Chan.ActZeroOffsetValues.....	173	NC.Chan.ModeOfOperation.....	194
NC.Chan.AuxActual.....	178	NC.Chan.OptionalStop.....	196
NC.Chan.AuxAdjustment.....	178	NC.Chan.ProgramLinkInfo.....	155
NC.Chan.AuxReference.....	178	NC.Chan.ProgrammedFeedrate.....	206
NC.Chan.AxisMirror.....	188	NC.Chan.ProgrammedFeedrateIndex.....	206
NC.Chan.AxisPoleSet.....	189	NC.Chan.ProgrammedSpindleSpeed.....	187
NC.Chan.AxisPosAcs.....	157	NC.Chan.ProgramPosition.....	196
NC.Chan.AxisPosAcsCoupl.....	158	NC.Chan.ProgramWindow.....	197
NC.Chan.AxisPosAcsDrv.....	159	NC.Chan.ReentryChain.....	198
NC.Chan.AxisPosAcsDrvAdj.....	159	NC.Chan.ReentryDelete.....	198
NC.Chan.AxisPosBcs.....	160	NC.Chan.ReentryEvent.....	198
NC.Chan.AxisPosBcsDrv.....	160	NC.Chan.ReturnToContour.....	199
NC.Chan.AxisPosDistAcs.....	161	NC.Chan.ReturnToPathStrategy.....	199
NC.Chan.AxisPosDistMcs.....	162	NC.Chan.SkipBlockMode.....	200
NC.Chan.AxisPosDistWcs.....	162	NC.Chan.UpCallFromNcBlock.....	200
NC.Chan.AxisPosEndWcs.....	163	NC.Chan.WaitingStates.....	156
NC.Chan.AxisPosLag.....	163	NC.CheckFile.....	78

NC.Cmd.DryRunMode.....	207
NC.Cmd.MessageStoreClr.....	208
NC.Cmd.Shutdown.....	209
NC.Cmd.Unlock.....	210
NC.ConnectionQuality.....	75
NC.CplPermVariable.....	100
NC.CplPermVariableList.....	101
NC.CplPermVariableWatch.....	102
NC.CpuUsage2.....	76
NC.DbTableDataRec_U.....	107
NC.DbTableDataRecLocked_U.....	108
NC.DbTableEvent.....	109
NC.DbTableLoad.....	110
NC.DbTableQuery_U.....	111
NC.DbTableSave.....	113
NC.DryRunMode.....	77
NC.Filesystem.....	77
NC.FIS.....	79
NC.LanguagesDownload.....	29
NC.MacodaParameter.....	113
NC.MtxCommand.....	79
NC.NofChannels.....	81
NC.OscilloscopeConfiguration.....	94
NC.OsciUtil.....	82
NC.PlcData.....	89
NC.ProgramChanInfo.....	91
NC.ReloadExportUs.....	92
NC.SercosPhase2.....	95
NC.SPA1.....	96
NC.SPH1.....	99
NC.SystemData_U.....	118
NC.SystemDataEvent.....	120
NC.SystemDataLoad.....	121
NC.SystemDataSave.....	121
NC.SystemDataWatch_U.....	122
NC.WaitingStates.....	93
NC.XmlRootElementInfo.....	103
NC.XmlRootElementList.....	103
NC.XmlSchemaList.....	104
NC.XmlTable_U.....	104
NC.XmlTableCreate.....	106
NC.XmlTableEvent.....	107
Nicht-bestimmungsgemäßer Gebrauch.....	12
Folgen, Haftungsausschluss.....	11

P

Plc.FeldBus.....	69
Plc.FirmwareConnection.....	65
Plc.FirmwareReboot.....	65
Plc.FirmwareStatus.....	66
Plc.FirmwareVersion.....	66
Plc.ModDiagnose.....	66
Plc.ModStatus.....	67
Plc.PVA.....	69
Plc.PVArray.....	69
Plc.PVBS.....	70
Plc.PVF.....	71

Plc.PVL.....	71
Plc.PVLK.....	72
Plc.PVT.....	73
Plc.StartStop.....	68
Plc.State2.....	67
Prozessdaten.....	187

R

Root.....	13
-----------	----

S

SAC.Config.....	44
SAV-Fehler.....	225
Scanning	
ComPortList.....	17
SCL.....	18
SD-Fehler.....	231
SDL.....	19
Sercos	
Attribute.....	32
Attributecache.....	35
Datentypen.....	32
DeviceAddress.....	44
DeviceAddressList.....	44
Elemente.....	31
Fehlercodes.....	36
Idsents.....	31
Listen.....	33
SAC.Config.....	44
Sercos Item.....	31
Sercos Items.....	38
SPA1.....	38
SPA2.....	41
Service-Hotline.....	241
SPA1.....	38
SPA2.....	41
Spindeldaten.....	181
Support.....	241
System.Archive.....	123
System.ArchiveLog.....	124
System.ArchiveWin.....	126
System.ArchiveWinLog.....	127
System.Cmd.Reset.....	210
System.Cmd.Restart.....	210
System.ErrorStatus.....	54
System.HardwareDescription.....	129
System.HardwareDescription2.....	129
System.InfoStatus.....	55
System.Language.....	130
System.LogbookClear.....	63
System.LogbookDevices.....	64
System.LogbookStart.....	64
System.LogbookStore.....	58
System.LogbookStoreEx.....	61
System.MainReleases.....	130
System.MessageReset.....	57
System.MessageStatus.....	53

Index

System.MessageStore.....	48
System.MessageStoreEx.....	51
System.PcReleases.....	131
System.ProVi.Active.....	136
System.ProVi.KaFreeze.....	131
System.ProVi.KaOnline.....	134
System.SetupStatus.....	57
System.Sfc.Active.....	143
System.Sfc.Execute.....	143
System.Sfc.KaExecute.....	144
System.Sfc.KaFreeze.....	136
System.Sfc.KaOnline.....	139
System.Sfc.Mode.....	142
System.Sfc.StepInfo.....	142
System.StartupStatus.....	56
System.SubsystemReleases.....	146
System.SupportedLanguages.....	147
System.SystemTime.....	147
System.Text	
System.TimeSync.....	147
System.TimeZone.....	148
System.TimeSync.....	147
System.TimeZone.....	148
System.WarningStatus.....	55
T	
TAD-Fehler.....	227
TAR-Fehler.....	237
U	
Unlock-Fehler.....	239
V	
Variablen.....	69
Vorschub- und Drehzahlwerte.....	201
X	
XML-Fehler.....	232
Z	
Zielgruppen.....	1
Zu dieser Dokumentation.....	1
Gültigkeit der Dokumentation.....	1

Notizen

Bosch Rexroth AG

Electric Drives and Controls

Postfach 13 57

97803 Lohr, Deutschland

Bgm.-Dr.-Nebel-Str. 2

97816 Lohr, Deutschland

Tel. +49 9352 18 0

Fax +49 9352 18 8400

www.boschrexroth.com/electrics



R911399271