



# DOKUMENTATION ISG-kernel

## Funktionsbeschreibung Zugriffe auf CNC

Kurzbezeichnung:  
FCT-C13

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

Dokumentation Version: 1.05  
07.11.2024

# Vorwort

## Rechtliche Hinweise

---

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

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

## Qualifikation des Personals

---

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

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

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

## Weiterführende Informationen

---

Unter den Links (DE)

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

bzw. (EN)

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

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

## Haftungsausschluss

---

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

## Marken und Patente

---

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

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

## Copyright

---

© ISG Industrielle Steuerungstechnik GmbH, Stuttgart, Deutschland.

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

# Allgemeine- und Sicherheitshinweise

## Verwendete Symbole und ihre Bedeutung

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

## Symbole im Erklärtext

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



### **GEFAHR**

#### **Akute Verletzungsgefahr!**

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



### **VORSICHT**

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

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



### **Achtung**

#### **Einschränkung oder Fehler**

Dieses Symbol beschreibt Einschränkungen oder warnt vor Fehlern.



### **Hinweis**

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

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



### **Beispiel**

#### **Allgemeines Beispiel**

Beispiel zu einem erklärten Sachverhalt.



### **Programmierbeispiel**

#### **NC-Programmierbeispiel**

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



### **Versionshinweis**

#### **Spezifischer Versionshinweis**

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

# Inhaltsverzeichnis

<b>Vorwort</b> .....	<b>2</b>
<b>Allgemeine- und Sicherheitshinweise</b> .....	<b>3</b>
<b>1 Einleitung</b> .....	<b>6</b>
<b>2 Beschreibung</b> .....	<b>10</b>
2.1 Task COM .....	11
2.1.1 Abfrage der Objektbeschreibung .....	11
2.1.2 Dateninhalt.....	13
2.2 Task SDA, Task GEO .....	13
2.2.1 Klassen .....	13
2.2.2 Verwaltung, IndexGroup/-Offset Adresse .....	14
<b>3 Dynamischer Zugriff auf Variablen</b> .....	<b>17</b>
3.1 P-Parameter .....	18
3.2 V.L. Variablen.....	18
3.3 V.P. Variablen .....	19
3.4 V.S. Variablen .....	19
<b>4 Schnittstelle für Variablen mit flexiblem Datentyp (V.E, V.G, V.A, V.CYC)</b> .....	<b>20</b>
4.1 Allgemeine Eigenschaften.....	20
4.2 Schnittstellen.....	21
4.2.1 Kanalspezifische Schnittstelle.....	21
4.2.2 Achsspezifische Schnittstelle (z.B. für V.A-Variablen).....	22
4.3 Übergabeparameter und Rückgabewerte .....	23
4.3.1 Funktion „Variable element number“ .....	23
4.3.2 Funktion „Variable description by index“ .....	25
4.3.3 Funktion „Variable handle by name“ .....	28
4.3.4 Funktion „Read variable value by handle“ .....	30
4.3.5 Funktion „Read variable value by name“ .....	32
4.3.6 Funktion „Write variable value by handle“.....	34
4.3.7 Funktion „Write variable value by name“ .....	36
4.3.8 Funktion „Extended description by index“ .....	38
4.3.9 Achsspezifische Funktion „Variable handle by name“ .....	40
4.3.10 Achsspezifische Funktion „Read variable value by name“ .....	42
4.3.11 Achsspezifische Funktion „Write variable value by name“.....	43
<b>5 ADS Dienste</b> .....	<b>44</b>
<b>6 Anhang</b> .....	<b>45</b>
6.1 Teildynamische Elemente .....	45
6.1.1 V.A. Variablen .....	45
6.1.2 V.G. Variablen.....	47
6.2 V.E. Variablen .....	48
6.2.1 Zugriff V.E.-Arrayelement .....	48
6.3 Anregungen, Korrekturen und neueste Dokumentation.....	49
<b>Stichwortverzeichnis</b> .....	<b>50</b>

## Abbildungsverzeichnis

Abb. 1:	Übersicht - CNC Architektur .....	7
Abb. 2:	Übersicht - Adressierung von Parameteranzahl .....	17

# 1 Einleitung

## ***Obligatorischer Hinweis zu Verweisen auf andere Dokumente***

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

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

Ein Zugriff auf die Daten der CNC erfolgt über CNC Objekte, die über Indexgroup und Indexoffset adressiert werden. Es stehen folgende Tasks zur Verfügung, die jeweils über eigene Ports angesprochen werden müssen:

- SDA,
- COM
- GEO



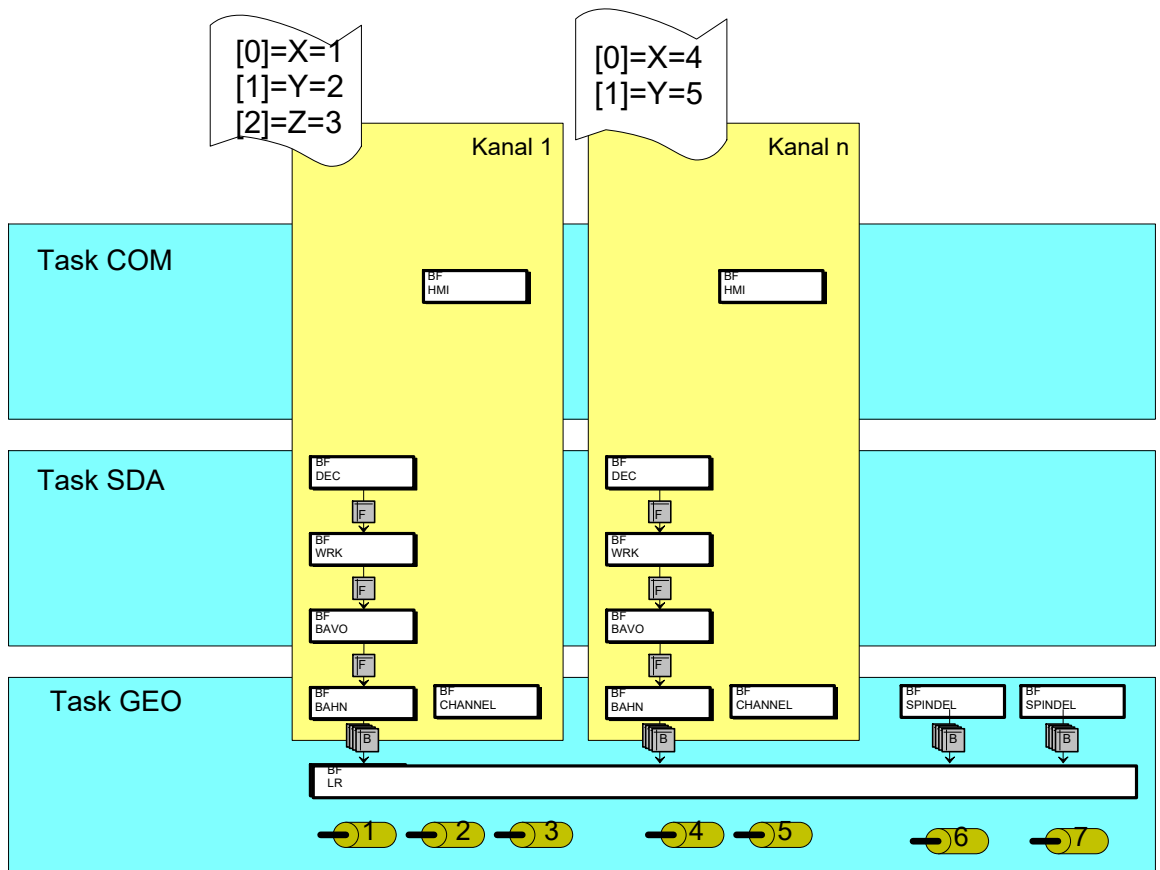
### **Hinweis**

**Es wird die Codepage CP1252 verwendet.**

## **CNC-Architektur**

Auf Objekte der CNC kann über ADS zugegriffen werden. Hierzu ist das Element über die Adresse ADS-NetId, IndexGroup und IndexOffset bestimmt.

Auf Objekte der CNC kann über IndexGroup und IndexOffset zugegriffen werden.



**Abb. 1: Übersicht - CNC Architektur**

Im obigen Bild sind beispielhaft folgende CNC-Klassen aufgeführt:

- Decodierung (BF DEC)
- Bahnplanung (BF BAVO)
- Interpolator (BF BAHN)
- Lageregelung (BF LR)
- Spindel (BF SPINDEL)

### Attribute für Zugriff

Die Instanzen der CNC-Klassen stellen Objekte zur Verfügung, die unterschiedliche Attribute besitzen. Je nach CNC-Konfiguration werden von jeder CNC-Klasse Instanzen angelegt und zu einer CNC-Architektur verknüpft. Auf diese Attribute der Objekte kann über Dienste zugegriffen werden. Entweder:

- Lesend (read)
- Schreibend (write)
- Lesend und schreibend (read/write)

### Plattform, Kanal, Achse

Je nach CNC-Klasse kann die Instanz plattformglobal sein (Kanalnummer = 0, z.B. Lageregler) oder entsprechend kanalspezifisch (Kanalnummer > 0) angelegt sein.

Zusätzlich kann eine Instanz sowohl eigene Elemente wie auch achsspezifische Elemente zur Verfügung stellen.

## Achsen der Plattform

---

Alle Achsen des Steuerungssystems werden global für die Steuerungsplattform vom Benutzer angelegt. Damit ergibt sich eine Reihenfolge der Achsen. Die Berechnung der Objekt-Adresse, mit der die einzelne Achse angesprochen werden kann, leitet sich von der Platzierung der Achse in dieser Reihenfolge der Achsen ab. Sie entspricht dem Index der Platzierung in dieser Reihenfolge. Die erste Achse in dieser Reihenfolge besitzt den Index 1. Mit dieser Adresse kann jederzeit auf die Achse zugegriffen werden.

## Achsen des Kanals

---

Die Adressierung der Achsen eines Kanals findet über den Index im Kanal statt. Durch die Kanalparametrierung und den Achstausch kann die Einordnung einer Achse in den Kanal beliebig geändert bzw. festgelegt werden.

Ist ein Index einer Kanalachse momentan nicht durch eine Achse belegt, so kann dies durch Lesen eines „leeren“ Dateninhalts eingesehen werden. Insbesondere ist in diesem Falle die logische Achsnummer = 0.

## Spindeln

---

Entsprechend eines CNC-Kanals, welcher für die koordinierte Bewegung mehrerer Achsen zuständig ist, steht die BF SPINDEL für die Bewegung einer einzigen unabhängigen Achse (Einzelachsinterpolator).

Somit ist für angelegte Spindeln ebenfalls ein Zugriff über CNC Objekte möglich.



### Hinweis

#### HLD

Ein Sonderfall stellt hier der High-Level-Treiber (HLD, BF CHANNEL) dar. Der HLD des ersten Kanals treibt zusätzlich zu den kanalspezifischen Elementen alle plattformglobale Achsen auf der Schnittstelle zur SPS. Die Adressierung der Achsen ändert sich also hier nicht durch einen Achstausch, sondern ist fest durch die Hochlaufbeschreibung gegeben.



### Hinweis

#### Basisadresse Task COM

Als Basisadresse für den Zugriff auf interne CNC-Daten der Task COM wird als IndexGroup = 0x20100 sowie 0x20200 verwendet.

Einzelne Kanäle oder Achsen können über unterschiedliche additive Werte zu dieser Basisadresse (IndexGroup) abgefragt werden.



### Hinweis

#### Basisadresse Task GEO

Als Basisadresse für den Zugriff auf interne CNC-Daten der Task SDA und der Task GEO wird als IndexGroup = 0x20300 definiert.

Einzelne Attribute eines Elements können über unterschiedliche additive Werte zu dieser Basisadresse (IndexGroup) abgefragt werden.



## Verfügbare statische CNC Objekte

Die verfügbaren statischen Objekte der CNC sind abhängig von der verwendeten CNC-Version. Über den ISG Objektbrowser kann eine Liste aller zur Verfügung stehender Objekte erstellt werden.

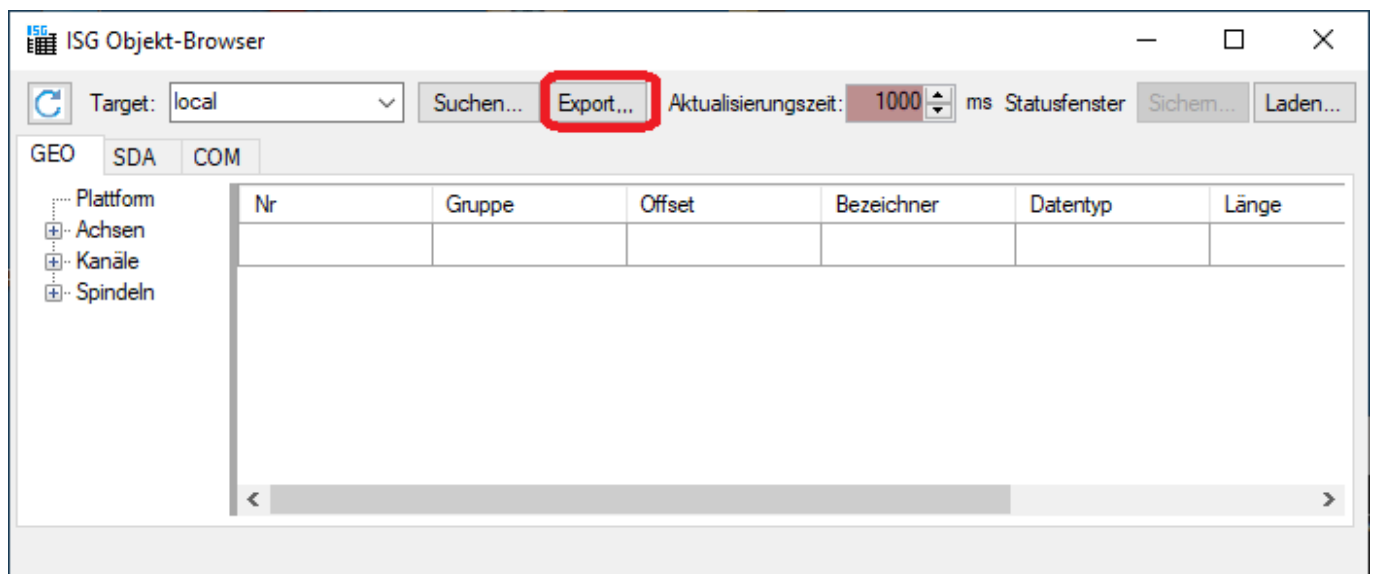
Vorraussetzungen :

ISG Objektbrowser ist installiert

Konfiguration ist in TwinCAT aktiviert und TwinCAT ist im RUNNING Modus

Folgende Schritte sind durchzuführen:

- ISG Objektbrowser starten



- Betätigen des Export –Buttons
- Festlegen des Speicherorts und des Dateinamens der erzeugten Liste.

In dieser erzeugten Liste sind alle in der CNC-Version statisch verfügbaren Objekte aufgeführt.

## 2 Beschreibung

### Hinweise zur Adressierung

<C<sub>ID</sub>> Channel- oder Kanal-ID, beginnend bei 1

<A<sub>ID</sub>> Achs-ID, beginnend bei 1

<S<sub>ID</sub>> Spindel-Index, beginnend bei 0. Der Spindel-Index ist Bestandteil der Indexgruppe.

### Adressierungsarten von "IndexGroup"

Die Angabe der Indexgruppe kann bei CNC-Objekten von Kanälen und Achsen sowohl mit „logischer“ als auch mit „indexbasierter“ Adressierung erfolgen

Die Anwahl der „logischen“ Adressierung erfolgt über das zusätzliche Bit 0x100000. Fehlt dieses Bit wird indexbasiert auf die Objekte zugegriffen. Das nachfolgende Beispiel zeigt die Bedeutung dieses Bit.

Auf Spindeln wird generell indexbasiert zugegriffen und der jeweilige Indexeintrag ist Bestandteil der Indexgruppe. Z.B. 0x25302 für die Indexgruppe der **dritten** Spindel.

Beispiel: Es gibt zwei nur Kanäle mit ihren logischen Nummern 1 und 6.

	Kanal „1“	Kanal „6“
Logischer Zugriff	IdxGrp = 0x121301	IdxGrp = 0x12130 <b>6</b>
Indexbasierter Zugriff	IdxGrp = 0x21301	IdxGrp = 0x2130 <b>2</b>

Es wird empfohlen den logischen Zugriff zu verwenden. Vorteil ist, dass bei Deaktivierung des Kanal 1 der Zugriff auf Kanal 6 unverändert bleibt. Bei indexbasiertem Zugriff ändert sich in diesem Fall die Indexgruppe des Kanals 6 von 0x21302 nach 0x21301 - er wird zum 'ersten' Kanal.



#### Hinweis

**Es wird empfohlen den logischen Zugriff zu verwenden.**

Alle in den nachfolgenden Kapiteln dargestellten Beispieladressierungen sind mit logischer Adressierungen aufgeführt.

## 2.1 Task COM

In der Task COM existiert nur eine Instanz, die sämtliche Elemente( Plattform, Kanäle, Achsen) verwaltet.

Klassen der Instanzen	Adressierung
Plattformdaten	IndexGroup = 0x120101, IndexOffset = <Element>
Achsen der Plattform	IndexGroup = 0x120200 +<Achse>, IndexOffset = <Element>
Kanaldaten	IndexGroup = 0x120100 +<Kanal>, IndexOffset = <Element>
Achsen des Kanals	0x120100 +<Kanal>, IndexOffset = 0x1000 * <Achse> + <Element>



### Hinweis

Der erste Kanal (IndexGroup = 0x120101) treibt die Plattformdaten.

### 2.1.1 Abfrage der Objektbeschreibung

Die Beschreibung eines Objekts der TASK COM kann über den Objektzugriff mit der Bitkennung 0x30000 abgefragt werden.

Ein Objekt wird über folgende 84 Byte beschrieben:

-Über die Bitkennung Objektaccess (ISGADS\_IGR\_OBJECT\_ACCESS = 0x30000) in der Indexgruppe kann die Beschreibung eines Objekts der Task COM abgefragt werden-

Datentyp	Bit	Datum	Beschreibung
DWORD	0-3	ID	Interne eindeutige ID des Objekts
DWORD	4-7	Größe	Größen des Objekts in Bytes
WORD	8-9	Schreibzugriff	TRUE falls Objekt geschrieben werden kann.
WORD	10-11	-	Nicht belegt
DWORD	12-15	Indexgruppe	für direkten Objekt-Zugriff auf den Inhalt
DWORD	16-19	Indexoffset	für direkten Objekt-Zugriff auf den Inhalt
STRING(32)	20-51	Name	Name des Objekts
STRING(32)	52-83	Typ	Datentyp des Objekts: BOOL, BYTE, SINT, WORD, INT, DWORD, DINT, LWORD, LINT, REAL, LREAL, STRING

Für die Objekte des ersten Kanals (<ch> = 1) wird so die Indexgruppe = 0x13010<ch> verwendet. Sind mehrere Kanäle konfiguriert, so kann der gewünschte Kanal entsprechend über logische Kanal-ID [1; max] ausgewählt werden. Die Plattformdaten sind wahlweise unter Kanal-ID = 0 oder 1 adressierbar.

- Gesamtzahl aller Objekte wird über Indexoffset = 0 angefordert.
- Lesen einer Objektbeschreibung Indexoffset > 0 liefert Objektbeschreibung mit dem Index [1; Objektanzahl] der Rückgabespeicher muss in Summe 84 Byte pro Objekt sein.

Lesen von mehreren Objektbeschreibung mit einem Zugriff

- Wird als Rückgabespeicher exakt ein Vielfaches von 84 Byte bereitgestellt, so werden ab dem übergebenen Index in Folge Objektbeschreibungen zurückgegeben.



### Beispiel

#### Beispiel

Indexgruppe = 0x130101, Indexoffset = 0

liefert die Objektanzahl des 1. Kanals zurück

Indexgruppe = 0x130101, Indexoffset = 1  
(Kanal=1) zurück

liefert die Objektbeschreibung des 1. Objekts



### Beispiel

#### Anfordern einer einzelnen Objektbeschreibung

Indexgruppe = 0x130101, Indexoffset = 0x10 liefert einzelne Objektbeschreibung mit 84 Byte.

Exemplarische Auflistung

- ID= 82 (4Byte)
- Größe = 1 (4Byte)
- Schreibzugriff = 0, also nur lesend (2Byte)
- Frei (2Byte)
- Indexgruppe = 0x00120101 (4Byte)
- Indexoffset = 0x00000003 (4Byte)
- Name = mc\_active\_single\_block\_r (32Byte)
- Typ = BOOL (32Byte)

## 2.1.2 Dateninhalt

Der eigentliche Schreib-/Lese-Zugriff auf die Dateninhalt der Objekte finden über die Bitkennung ISGADS\_IGR\_ELEMENT\_ACCESS = 0x20000 in der IndexGruppens statt.

- Plattform 0x120100, 0x251
- Achse 0x120200, 0x10001
- Kanal 0x120101, 0x1
- Kanalachse 0x120101, 0x10001

Der später Zugriff auf die Daten des Objekts kann dann über indexGroup = 0x120101, indexOffset = 0x1 "mc\_command\_single\_block\_w" erfolgen.

## 2.2 Task SDA, Task GEO

### 2.2.1 Klassen

#### IndexGroup-Basis der CNC

IndexGroup = 0x120300

#### IndexGroup der Klassen

Folgende Klassen der CNC-Bausteine sind aktuell mit der entsprechenden IndexGroup adressierbar:

- 0x120300 : Lageregler (einmal pro CNC, kanalunabhängig mit plattformglobale Achsen)
- 0x121300 : Interpolator (kanalspezifisch mit Kanalachsen)
- 0x122300 : Decoder (kanalspezifisch mit Kanalachsen)
- 0x123300 : HLD (kanalspezifisch, HLD des ersten Kanals enthält zusätzlich Achsen plattformglobal)
- 0x124300 : Bahnvorbereitung (kanalspezifisch)
- 0x25300 : Spindel

Die einzelnen Klassen haben folgende additive Werte auf den IndexGroup der CNC:

- 0x0000 : Lageregler (einmal pro CNC, kanalunabhängig mit plattformglobale Achsen)
- 0x1000 : Interpolator (kanalspezifisch mit Kanalachsen)
- 0x2000 : Decoder (kanalspezifisch mit Kanalachsen)
- 0x3000 : HLD (kanalspezifisch, HLD des ersten Kanals enthält zusätzlich Achsen plattformglobal)
- 0x4000 : Bahnvorbereitung (kanalspezifisch)
- 0x5000 : Spindel

#### Attribute eines Elements

Folgende Attribute eines Elements können abgefragt werden (additive Werte zur IndexGroup):

- 0x000 : Dateninhalt (je nach Datentyp)
- 0x100 : Name als String
- 0x200 : Typ als String [UNS08;SGN08; ... ;SGN32; REAL32, REAL64]
- 0x300 : Länge der Daten in Bytes
- 0x400 : Einheit als String
- 0x500 : Flag BOOLEAN, ob Schreibzugriff erlaubt ist
- 0x600 : Objekt-Adresse : In (Kanal, Achse), Out (IndexGroup, IndexOffset)

## 2.2.2 Verwaltung, IndexGroup/-Offset Adresse

Um zu ermitteln, welche Instanzen einer Klasse vorhanden sind, kann über einen READ&WRITE Zugriff die Objektadresse des ersten Elements (IndexOffset = 0x0) abgefragt werden.

Zurückgegeben wird der IndexGroup und der IndexOffset. Wird als Adresse (0, 0) zurückgegeben, so existiert keine Instanz dieser Klasse.

Ist eine Instanz vorhanden, so können durch Inkrementieren des IndexOffsets alle weiteren vorhandenen Elemente der Instanz ermittelt werden.



### Beispiel

#### Abfrage von Objekten

Abfrage der Objektadresse des 1.Kanals

IndexGroup =0x123900 zusammengesetzt aus

<CNC-Basis>	+ <Class = HLD>	+ <Attribut=Adresse>
0x20300	+ 0x3000	+ 0x600

Input : Channel = **1**, Axis = **0**

Output :IndexGroup = 0x123301, IndexOffset = 0x0

Abfrage der Objektadresse des 2.Kanals

<CNC-Basis>	+ <Class = HLD>	+ <Attribut=Adresse>
0x20300	+ 0x3000	+ 0x600

Input : Channel = **2**, Axis = **0**

Output :IndexGroup = 0x123302, IndexOffset = 0x0

Abfrage der 1.Achse der Plattform

<CNC-Basis>	+ <Class = HLD>	+ <Attribut=Adresse>
0x120300	+ 0x3000	+ 0x600

Input : Channel = **0**, Axis = **1**

Output :IndexGroup = 0x123300, IndexOffset = 0x1000

Abfrage der 2. Decoderachse im 1. Kanal

<CNC-Basis>	+ <Class = HLD>	+ <Attribut=Adresse>
0x120300	+ 0x2000	+ 0x600

Input : Channel = **1**, Axis = **2**

Output :IndexGroup = 0x122301, IndexOffset = 0x20000

## Anzahl der Elemente einer Klasse

Die Anzahl der vorhandenen Elementtypen einer Instanz kann über den Wertinhalt des ersten Elements (IndexOffset = 0) abgefragt werden.

Zusätzlich beschreibt der Name dieses Elements die Instanz selbst.

## Lageregler

Der Lageregler ist nicht kanalspezifisch, d.h. es gilt immer Kanal = 0. Die Achsen gehen von 1 bis zur Anzahl der konfigurierten Achsen.

## Interpolator, Decoder, Bahnvorbereitung

Die Kanalnummer ist im Bereich [ 1; Anzahl konfiguierter Kanäle ] anzugeben.

Die Achsen sind kanalspezifisch und müssen im Bereich [0; <BAHN\_BAVO\_AXMAX>] liegen.

## Channel, HLI-Treiber

Die Kanalnummer ist im Bereich [ 1; <Anzahl konfiguierter Kanäle> ] anzugeben.

Werden Achsen referenziert, so müssen diese im Bereich [ 1; <Anzahl der konfiguierter Achsen>] liegen.

## Spindel

Für Spindeln wird als „Channel“-Eintrag der Spindelindex angegeben. Der Spindelindex geht von 0 bis zur Anzahl der konfigurierten Spindelachsen - 1.

## Abfrage der Attribute der Elemente

Am Beispiel des Elements feedhold.enable im 1. Kanal wird die Abfrage veranschaulicht. Der IndexOffset des Elements ist 0x01. Es wird die Zusammensetzung der IndexGroup gezeigt; der IndexOffset des Elements ist unverändert.

Attribut Dateninhalt:

<CNC-Basis>	+ <Class = HLD>	+ <Attribut=Adresse>	<Kanal>
0x120300	+ 0x3000	+ 0x0	+ 1

Input : IndexGroup = 0x123301, IndexOffset = 0x01

Output : 0/1 (abhängig vom Datentyp des Elements)

Attribut Name:

<CNC-Basis>	+ <Class = HLD>	+ <Attribut=Adresse>	<Kanal>
0x120300	+ 0x3000	+ 0x100	+ 1

Input : IndexGroup = 0x123401, IndexOffset = 0x01

Output : feedhold.enable

## Attribut Typ:

<CNC-Basis>	+ <Class = HLD>	+ <Attribut=Adresse>	<Kanal>
0x20300	+ 0x3000	+ 0x200	+ 1

Input : IndexGroup = 0x123501, IndexOffset = 0x01

Output : BOOLEAN

## Attribut Länge in Byte:

<CNC-Basis>	+ <Class = HLD>	+ <Attribut=Adresse>	<Kanal>
0x120300	+ 0x3000	+ 0x300	+ 1

Input : IndexGroup = 0x123601, IndexOffset = 0x01

Output : 1

## Attribut Einheit:

<CNC-Basis>	+ <Class = HLD>	+ <Attribut=Adresse>	<Kanal>
0x120300	+ 0x3000	+ 0x400	+ 1

Input : IndexGroup = 0x123701, IndexOffset = 0x01

Output : -

## Attribut Schreibschutz:

<CNC-Basis>	+ <Class = HLD>	+ <Attribut=Adresse>	<Kanal>
0x120300	+ 0x3000	+ 0x500	+ 1

Input : IndexGroup = 0x123801, IndexOffset = 0x01

Output : 0



### 3 Dynamischer Zugriff auf Variablen

Objekte der CNC sind prinzipiell statisch und eindeutig identifizierbar über IndexGroup und IndexOffset.

Ausnahmen bilden die folgende Gruppierungen, diese Variablen oder Parameter werden dynamisch gelesen.

#### Task GEO

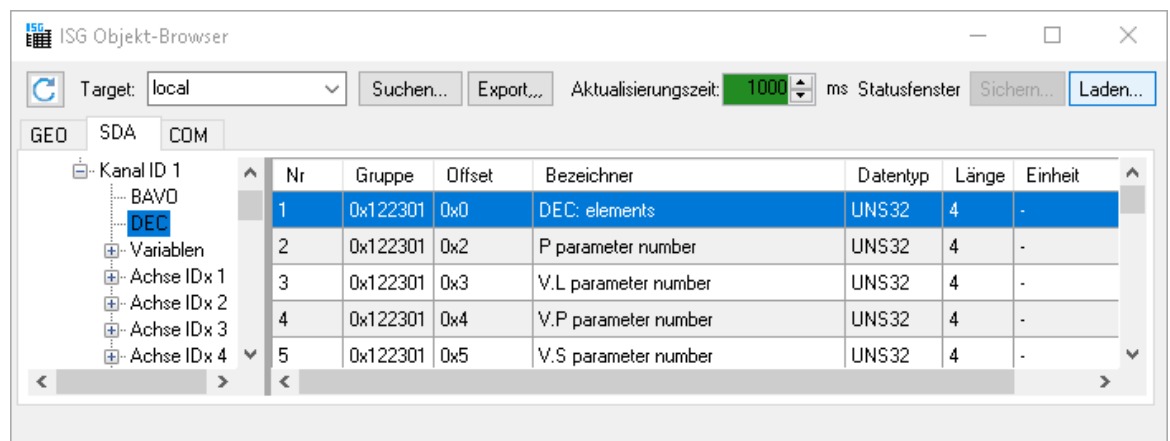
- AEP-Parameter

#### Task SDA

- V.A.-Variablen
- V.E.-Variablen
- V.CYC.-Variablen
- V.G.-Variablen
- P-Parameter [▶ 18]
- V.L.-Variablen [▶ 18]
- V.P.-Variablen [▶ 19]
- V.S.-Variablen [▶ 19]

### Dynamische Elemente

Während der Decodierung eines NC-Programms können Variablen dynamisch angelegt und auch wieder gelöscht werden. Auch auf diese Variablen kann zugegriffen werden.



Nr	Gruppe	Offset	Bezeichner	Datentyp	Länge	Einheit
1	0x122301	0x0	DEC: elements	UNS32	4	-
2	0x122301	0x2	P parameter number	UNS32	4	-
3	0x122301	0x3	V.L parameter number	UNS32	4	-
4	0x122301	0x4	V.P parameter number	UNS32	4	-
5	0x122301	0x5	V.S parameter number	UNS32	4	-

**Abb. 2: Übersicht - Adressierung von Parameteranzahl**

Betroffen sind:

- P-Parameter [▶ 18]
- V.L.-Variablen [▶ 18]
- V.P.-Variablen [▶ 19]
- V.S.-Variablen [▶ 19]

Für die Anbindung dieser Variablen gilt der folgende Ablauf:

1. Lesen der aktuellen Parameteranzahl  $n_{max}$
2. Für jeden Parameter [ 1; $n_{max}$ ] den Namen lesen
3. Lesen des aktuellen Inhaltes eines Parameters



### Hinweis

Der Wert eines Parameters sollte immer über seinen Namen gelesen werden.

Das Lesen eines Parameters über den Index ist nicht zu empfehlen, da sich der Index eines Parameters durch dynamisches Anlegen und Löschen von Elementen ändern kann.



### Hinweis

Die maximale Länge eines Variablennamens beträgt 96 Byte. Das betrifft alle V.-Variablen und P-Parameter.

## 3.1 P-Parameter

Gruppe	Offset	Bezeichner	Typ	Länge
0x122301	0x2	P-parameter number	UNS32	4
0x122301	0x1B	P-name by index	UNS32/STRING	96
0x122301	0x1C	P-value by name	STRING/REAL64	8
0x122301	0x1D	P-value by index	UNS32/REAL64	8



### Beispiel

Lesen eines P-Parameters mit vorheriger Abfrage der Parameteranzahl

- Lesen der Anzahl der P-Parameter im 1.Kanal  
 Input : IndexGroup = 0x122301, IndexOffset = 0x02  
 Output : 1
- Lesen des Namen des P-Parameters über den Index des Parameters  
 Input : IndexGroup = 0x122301, IndexOffset = 0x1b , Parameter = 1  
 Output : P1
- Lesen des Wertes eines P-Parameters über dessen Namen  
 Input : IndexGroup = 0x122301, IndexOffset = 0x1c , Parameter : P1  
 Output : 4711

## 3.2 V.L. Variablen

Gruppe	Offset	Bezeichner	Typ	Länge
0x122301	0x3	V.L parameter number	UNS32	4
0x122301	0x1E	V.L-name by index	UNS32/STRING	96
0x122301	0x1F	V.L-value by name	STRING/REAL64	8
0x122301	0x20	V.L-value by index	UNS32/REAL64	8

### 3.3 V.P. Variablen

Gruppe	Offset	Bezeichner	Typ	Länge
0x122301	0x4	V.P parameter number	UNS32	4
0x122301	0x21	V.P-name by index	UNS32/STRING	96
0x122301	0x22	V.P-value by name	STRING/REAL64	8
0x122301	0x23	V.P-value by index	UNS32/REAL64	8

### 3.4 V.S. Variablen

Gruppe	Offset	Bezeichner	Typ	Länge
0x122301	0x5	V.S parameter number	UNS32	4
0x122301	0x24	V.S-name by index	UNS32/STRING	96
0x122301	0x25	V.S-value by name	STRING/REAL64	8
0x122301	0x26	V.S-value by index	UNS32/REAL64	8

## 4 Schnittstelle für Variablen mit flexiblem Datentyp (V.E, V.G, V.A, V.CYC)

### 4.1 Allgemeine Eigenschaften

- Für das Lesen und Schreiben von V.E, V.G, V.A und V.CYC Variablen gibt es eine gemeinsame, kanalspezifische Schnittstelle
- Für achsspezifische Variablen (V.A) gibt es zusätzlich eine achsspezifische Schnittstelle. In der kanalspezifischen Schnittstelle muss bei achsspezifischen Variablen zur Adressierung der gewünschten Achse an den Variablennamen der Achsname angehängt werden. In der achsspezifischen Schnittstelle hingegen wird implizit der Variablenwert der aktuell auf diesem Platz einsortierten Achse gelesen. Beispiel:  
Achsspezifische Schnittstelle: V.A.ACT\_POS  
Kanalspezifische Schnittstelle: V.A.ACT\_POS.X
- Der Zugriff auf den Variablenwert kann über Name oder Handle stattfinden, wobei letzterer schneller ist. Ausgenommen sind hier die V.CYC-Variablen. Diese sind nur per Name zugreifbar!  
Das Handle der Variablen ist variabel (abhängig von der Konfiguration)! Daher kann es über den Variablennamen abgefragt werden.
- Um auch ohne „Vorwissen“ auf die Variablen zugreifen zu können, kann die Anzahl der vorhandenen Variableneinträge ermittelt und die Variablenbeschreibungen pollend (per Index) abgefragt werden.
- Variablenarrays mit elementarem Datentyp (auch mehrdimensionale Felder und „Endknoten“ von Variablenstrukturen) werden als Einheit behandelt. Beispiel:  
V.E.array\_sgn32, V.E.struct.array\_uns32, V.G.NP[0].V, V.G.NP[1].V  
Bei Lesen oder Schreiben einer Array-Variablen kann entweder auf einzelnes Element oder komplettes Feld zugegriffen werden. Beispiel:  
V.G.NP[0].V[1] -> einzelnes Element  
V.G.NP[0].V -> Komplettes Array V.G.NP[0].V[0] ... V.G.NP[0].V[n]
- Variablenstrukturen werden in ihre Einzelelemente aufgelöst, d.h. der Variablenname setzt sich aus Strukturnamen, Elementname und Array Indizes zusammen. Beispiele:  
V.G.WZ\_AKT.SPDL\_AX\_NR  
V.E.Trajektorie.Vektor.x, V.E.Trajektorie.Vektor.y, V.E.Trajektorie.Vektor.z  
V.E.struct1[0].struct2[0].array\_sgn32, ..., V.E.struct1.struct1[5].struct2[8].array\_sgn32  
(Endknoten „array\_sgn32“ wird als Einheit behandelt!)

## 4.2 Schnittstellen



### Hinweis

Nur V.CYC.-Variablen können n-dimensional verwendet werden.

### 4.2.1 Kanalspezifische Schnittstelle

Index-Gruppe: IdxGrp = 0x122300 + Kanal\_ID , beispielsweise für Kanal 1: 0x122301:

IdxGrp	IdxOffs	Funktion	Beschreibung	Input	Output
0x122301	0x44	Variable element number [▶ 23]	Anzahl vorhandener Variableneinträge der gewählten Variablenart. Ist keine Variablenart angegeben → Gesamtzahl aller verfügbaren Variablen V.E + V.G + V.A (Größter Index = Anzahl der Elemente – 1)	STRING max. 255 Zeichen	UNS32
0x122301	0x45	Variable description by index [▶ 25]	Variablenbeschreibung (Name, Handle, Variablentyp, Zugriff, Größe, Arraydimensionen, größte Indizes)	BYTE[2048] variable Länge	BYTE[2048] variable Länge
0x122301	0x46	Variable handle by name [▶ 28]	Handle für Zugriff auf die Variable	STRING max. 255 Zeichen	Handle (UNS32)
0x122301	0x47	Read variable value by handle [▶ 30]	Variablenwert über Handle der Variablen lesen	BYTE[2048] variable Länge	BYTE[2048] variable Länge
0x122301	0x48	Read variable value by name [▶ 32]	Variablenwert über Variablennamen lesen	STRING max. 255 Zeichen	BYTE[2048] variable Länge
0x122301	0x49	Write variable value by handle [▶ 34]	Variablenwert über Handle der Variablen schreiben	BYTE[2048] variable Länge	-
0x122301	0x4A	Write variable value by name [▶ 36]	Variablenwert über Variablennamen schreiben	BYTE[2048] variable Länge	-
0x122301	0x4B	Extended variable description by index [▶ 38]	Erweiterte Variablenbeschreibung z.B. für V.E Variablen Synchronisierung und Gültigkeitsbereich	BYTE[2048] variable Länge	BYTE[2048] variable Länge
0x122301	0x4C-0x4E	reserved	Reserviert	-	-



## Achtung

### Folgende Funktionen sind nicht für V.CYC.-Variablen verfügbar:

"Variable handle by name"

"Read variable value by handle"

"Write variable value by handle"

## 4.2.2

### Achsspezifische Schnittstelle (z.B. für V.A-Variablen)

Index-Gruppe: IdxGrp = 0x122300 + Kanal\_ID

Index-Offset weiterer Achsen: IdxOffs += 0x10000 \* Achsindex

Beispiel 1. Kanal, 1. Achse: IdxGrp = 0x122301

IdxGrp	IdxOffs	Funktion	Beschreibung	Input	Output
0x122301	0x10004	Variable handle by name [▶ 40]	Handle für Zugriff auf die Variable	STRING max. 255 Zeichen	Handle (UNS32)
0x122301	0x10005	Read variable value by name [▶ 42]	Variablenwert über Variablennamen lesen	STRING max. 255 Zeichen	BYTE[2048] variable Länge
0x122301	0x10006	Write variable value by name [▶ 43]	Variablenwert über Variablennamen schreiben	BYTE[2048] variable Länge	-
0x122301	0x10007-0x10009	reserved	Reserviert	-	-

#### Instanzkennung

In der achsspezifischen Schnittstelle beinhalten die Variablenamen keine Instanzkennung.

Beispiel: V.A.ACT\_POS

Die achsspezifische Schnittstelle liefert immer die Variablenwerte der aktuell auf diesem Platz einsortierten Achse. Bei Achstausch ändert sich daher der angezeigte Variablenwert.

## 4.3 Übergabeparameter und Rückgabewerte



### Hinweis

Die Übergabeparameter und die Rückgabewerte werden als Stream behandelt.

### 4.3.1 Funktion „Variable element number“

Index-Gruppe: IdxGrp = 0x122300 + Kanal\_ID

Index-Offset: IdxOffs = 0x44

<b>Anzahl von Variableneinträgen</b>	Diese Funktion liefert für die gewählte Variablenart die Anzahl der vorhandenen Variableneinträge. Ist keine Variablenart angegeben, wird die Gesamtzahl aller verfügbaren Variablen V.E + V.G + V.A zurückgeliefert. Mit der Funktion „Variable description by index“ die jeweilige Variablenbeschreibung abgefragt werden.
--------------------------------------	--

Übergabeparameter (hier mit Angabe des Stream-Offsets):

0	Bezeichner für Variablenart
---	-----------------------------

wobei

Inhalt	Typ	Größe
Bezeichner für Variablenart mit Nullterminierung	STRING “V.G“ - Globale Variablen “V.E“ - Externe Variablen “V.A“ - Achsspezifische Variablen	variabel, max. 256 Byte

=> Streamlänge max. 256 Byte

Rückgabewert (hier mit Angabe der Stream-Offsets):

0	Anzahl der vorhanden Variableneinträge
---	--

wobei

Inhalt	Typ	Größe
Anzahl der vorhanden Variableneinträge	UNS32	fest, max. 4 Byte

=> Streamlänge max. 4 Byte



### Hinweis

Der Bezeichner für die Variablenart z.B. V.E darf abschließend keinen Punkt haben.

**Anwendungsbeispiele**

Übergabeparameter:

V.E	0
-----	---

V.E Bezeichner der Variablenart : V.E

0 Nullterminierung

Rückgabewert:

123
-----

Im 1. Kanal sind 123 V.E.Variablen parametrieret.



### 4.3.2 Funktion „Variable description by index“

Index-Gruppe: IdxGrp = 0x122300 + Kanal\_ID

Index-Offset: IdxOffs = 0x45

<b>Variablenbeschreibung</b>	Diese Funktion liefert für gegebenen Index und Variablenart die zugehörige Variablenbeschreibung. Die Gesamtzahl der vorhandenen Variableneinträge kann mit der Funktion „Variable element number“ abgefragt werden. Gültiger Wertebereich des Index ist 0 bis (Anzahl der Elemente – 1).
------------------------------	--

Übergabeparameter (hier mit Angabe der Stream-Offsets):

0	4
Index der Variablen	Bezeichner für Variablenart mit Nullterminierung

wobei

Inhalt	Typ	Größe
Index der Variablen	UNS32	fest, 4 Byte
Bezeichner für Variablenart	STRING “V.G“ - Globale Variablen “V.E“ - Externe Variablen “V.A“ - Achsspezifische Variablen	variabel, max. 256 Byte

=> Streamlänge max. 260 Byte

Rückgabewert (hier mit Angabe der Stream-Offsets):

0	4	8	12	16	(20)	(24)	...	20 + 4 *m
Handle auf die Variable	Variablentyp	Zugriff	Größe	Anzahl der Arraydimensionen m	Anzahl Elemente in Dimension 1	Anzahl Elemente in Dimension 2	...	Variablenname

wobei

Inhalt	Variablentyp	Größe
Handle auf die Variable	UNS32	fest, 4 Byte
Variablentyp	Enumeration (UNS32) (*)	fest, 4 Byte
Zugriff	Enumeration (UNS32) 1 = READ_ONLY 2 = READ_WRITE	fest, 4 Byte
Größe	UNS32	fest, 4 Byte
Anzahl der Arraydimensionen m	UNS32 ( = 0, elementare Variable)	fest, 4 Byte
Anzahl der Elemente Index: 0 – (n-1) je Arraydimension	UNS32	variabel, 4 Byte * m
Variablenname	String	variabel, max. 256 Byte

=> Streamlänge 276 Byte + Anzahl Arraydimensionen n \* 4 Byte, max. 2048 Byte

(\*) Enumeration für Variablentyp:

1 = BOOLEAN	4 = SGN16	7 = UNS32
2 = SGN08	5 = UNS16	8 = REAL64
3 = UNS08	6 = SGN32	9 = STRING



### Hinweis

Der Bezeichner für die Variablenart z.B. V.E darf abschließend keinen Punkt haben.



## Beispiel

### Anwendungsbeispiele

Im nachfolgenden Beispiel soll die Beschreibung der V.E. Variable mit dem Index 2 gelesen werden.

IndexGrp: 0x122301 für den ersten Kanal

IndexOffs: 0x45

Übergabeparameter:

2	V.E	0
---	-----	---

2 für die Variable mit dem Indexeintrag 2

V.E Bezeichner der Variablenart : V.E

0 Nullterminierung

Rückgabewert:

0x2000002	1	1	2	1	2	V.E.Variable_3
-----------	---	---	---	---	---	----------------

0x2000002 Handle

1 Typ der Variablen : Boolean

1 Zugriffsrechte : Read-Only

2 Gesamtgröße in Bytes

1 Arraydimension

2 Anzahl der Arrayelemente

V.E.Variable\_3 Name der Variable

### 4.3.3 Funktion „Variable handle by name“

Index-Gruppe: IdxGrp = 0x122300 + Kanal\_ID

Index-Offset: IdxOffs = 0x46

<b>Handle über Namen</b>	<p>Übergibt das zum Variablennamen gehörende Handle für den Zugriff auf die Variable. Bei achsspezifischen Variablen ist das Handle abhängig vom Platz (Achsisindex) der Achse, an der sie aktuell im Kanal einsortiert ist. Wird die Achse daher durch Achstausch auf einen anderen Achsisindex gelegt, muss das Handle erneut ermittelt werden.</p> <p>Bei Strukturen wie eigendefinierte V.E.-Variablen muss der Name des Strukturelements angegeben werden. Die Struktur wird in Einzelelemente aufgelöst.</p>
--------------------------	--



#### Hinweis

**Das Handle ist nur zur Laufzeit der Steuerung gültig. Es muss bei einem Neustart der Steuerung erneut gelesen werden.**

Übergabeparameter (hier mit Angabe des Stream-Offsets):

0
Variablenname mit Nullterminierung

wobei

Inhalt	Typ	Größe
Variablenname (mit Instanzkennung bei achsspezifischen Variablen)	STRING	variabel, max. 256 Byte

=> Streamlänge max. 256 Byte

Rückgabewert (hier mit Angabe des Stream-Offsets):

0
Handle auf die Variable

wobei

Inhalt	Typ	Größe
Handle auf die Variable	UNS32	fest, max. 4 Byte

=> Streamlänge max. 4 Byte

**Anwendungsbeispiele**

Übergabeparameter:

V.E.vartest	0
-------------	---

V.E.vartest Name der Variable

0 Nullterminierung

Rückgabewert:

0x2000004
-----------

0x2000004 Handle der angeforderten V.E.Variablen

### 4.3.4 Funktion „Read variable value by handle“

Index-Gruppe: IdxGrp = 0x122300 + Kanal\_ID

Index-Offset: IdxOffs = 0x47

<b>Lesen über Handle</b>	Mit dieser Funktion kann der Wert einer Variablen über ihren Handle gelesen werden.
--------------------------	---

Übergabeparameter (hier mit Angabe des Stream-Offsets):

0	4	8	12	8 + 4 Byte * n
Handle auf die Variable	Anzahl n Arraydimensionen	1. Index	2. Index	...

wobei

Inhalt	Typ	Größe
Handle auf die Variable	UNS32	fest, 4 Byte
Anzahl n Arraydimensionen	UNS32	fest, 4 Byte
1. – n. Arrayindex	UNS32	variabel, n * 4 Byte

=> Streamlänge 8 + Anzahl Arraydimensionen n \* 4 Byte, max. 2048 Byte

Rückgabewert (hier mit Angabe des Stream-Offsets):

0
Variablenwert

wobei

Inhalt	Typ	Größe
Variablenwert	String, REAL64, SGN32, REAL64[ ], SGN32[ ][ ], ...	variabel, max. 2048 Byte

=> Streamlänge max. 2048 Byte

Bei einer Array-Variablen kann neben einem Einzelwert auch das komplette Feld gelesen werden, in dem keine Array-Indizes übergeben werden.

### Anwendungsbeispiele

1. Beispiel für das Lesen eines Arrayelements eines eindimensionalen Arrays:

Übergabeparameter:

Handle auf die Variable	Anzahl Arraydimensionen	1. Index	Rückgabewert
Handle auf das Array 0x20012341	1	4	Wert des 5. Elements V.G.array[4]

0x20012341 Handle auf die Variablen

1 Anzahl der Arraydimensionen

4 für Index 4 also den Wert des 5. Elements der Variablen V.E.test[] → V.E.test[4]

Rückgabewert:

333
-----

333 Variablenwert

2. Beispiel für das Lesen eines gesamten Arrays mit 5 Elementen:

Handle auf die Variable	Anzahl Arraydimensionen	1. Index	Rückgabewert
Handle auf das Array 0x2001234	0		Wert aller Array-Elemente 0 bis (n-1)

0x2001234 Handle auf die Variablen

0 Anzahl der Arraydimensionen

Rückgabewerte:

123	234	345	456	567
-----	-----	-----	-----	-----

123 Wert des 1.Element

234 Wert des 2.Element

345 Wert des 3.Element

456 Wert des 4.Element

567 Wert des 5.Element

### 4.3.5 Funktion „Read variable value by name“

Index-Gruppe: IdxGrp = 0x122300 + Kanal\_ID

Index-Offset: IdxOffs = 0x48

<b>Lesen über Namen</b>	Mit dieser Funktion kann der Wert einer Variablen über ihren Namen gelesen werden.
-------------------------	--

Übergabeparameter: (hier mit Angabe der Stream-Offsets)::

0
Variablenname mit Nullterminierung

wobei

Inhalt	Typ	Größe
Variablenname (mit Instanzkennung bei achsspezifischen Variablen)	STRING	variabel, max. 256 Byte

=> Streamlänge 256 Byte

Rückgabewert: (hier mit Angabe der Stream-Offsets)::

0
Variablenwert

wobei

Inhalt	Typ	Größe
Variablenwert	String, REAL64, SGN32, REAL64[ ], SGN32[ ][ ], ...	variabel, max. 2048 Byte

=> Streamlänge max. 2048 Byte

Bei einer Array-Variablen kann neben einem Einzelwert auch das komplette Feld gelesen werden, in dem im Variablennamen keine Indizes angegeben werden.

Beispiel Variable V.G.array[0 .. 9]:

Variablenname	Rückgabewert
V.G.array[4]	Wert des 5. Elements V.G.array[4]
V.G.array	Wert der Array-Elemente 0 - 9



### Anwendungsbeispiele

Beispiel für das Lesen eines Arrayelements eines eindimensionalen Arrays:

Übergabeparameter:

V.E.Testvar[3]	0
----------------	---

V.E.Testvar[3] Name der Variable

0 Nullterminierung

Rückgabewert:

333
-----

333 Variablenwert des Elements

Beispiel für das Lesen eines gesamten Arrays mit 5 Elementen:

Übergabeparameter:

V.E.Testvar	0
-------------	---

V.E.Testvar Name der Variable

0 Nullterminierung

Rückgabewerte:

123	234	345	456	567
-----	-----	-----	-----	-----

123 Wert des 1.Element

234 Wert des 2.Element

345 Wert des 3.Element

456 Wert des 4.Element

567 Wert des 5.Element

### 4.3.6 Funktion „Write variable value by handle“

Index-Gruppe: IdxGrp = 0x122300 + Kanal\_ID

Index-Offset: IdxOffs = 0x49

<b>Schreiben über Handle</b>	Mit dieser Funktion kann der Wert einer Variablen über ihr Handle geändert werden.
------------------------------	--

Übergabeparameter: (hier mit Angabe der Stream-Offsets)::

0	4	8	12	...	$8 + 4 * n$
Handle auf die Variable	Anzahl n Arrayindizes	1. Index	2. Index	...	Variablenwert

wobei

Inhalt	Typ	Größe
Handle auf die Variable	UNS32	fest, 4 Byte
Anzahl n übertragener Arrayindizes	UNS32	fest, 4 Byte
1. – n. Arrayindex	UNS32	variabel, $n * 4$ Byte
Variablenwert	String, REAL64, SGN32, REAL64[ ], SGN32[ ][ ], ...	variabel, max. 2040 Byte

=> Streamlänge max. 2048 Byte

Diese Funktion besitzt keinen Rückgabewert.

Bei einer Array-Variablen kann neben einem Einzelwert auch das komplette Feld geschrieben werden, in dem keine Array-Indizes übergeben werden.

Beispiel Variable V.G.array[0 .. 9]:

Handle auf die Variable	Anzahl Arrayindizes	1. Index	Variablenwert
Handle auf das Array	1	4	Wert des 5. Elements V.G.array[4]
Handle auf das Array	0		Wert der Array-Elemente 0 - 9

### Anwendungsbeispiele

1. Beispiel für das Schreiben eines Arrayelements eines eindimensionalen Arrays:

Übergabeparameter:

0x2001234	1	4	123
-----------	---	---	-----

0x2001234 Handle der Variablen

1 Anzahl Arraydimensionen

4 für den Index 4 also das 5. Element der Variablen V.E.test[] → V.E.test4[]

123 Neuer Wert des 5. Elements der Variablen

Rückgabewert: -

2. Beispiel für das Schreiben eines gesamten Arrays mit 5 Elementen:

0x2001234	0	123	234	345	456	567
-----------	---	-----	-----	-----	-----	-----

0x2001234 Handle der Variablen

0 Anzahl Arraydimensionen

123 Neuer Wert des 1. Elements der Variablen

234 Neuer Wert des 2. Elements der Variablen

345 Neuer Wert des 3. Elements der Variablen

456 Neuer Wert des 4. Elements der Variablen

567 Neuer Wert des 5. Elements der Variablen

Rückgabewert: -

### 4.3.7 Funktion „Write variable value by name“

Index-Gruppe: IdxGrp = 0x122300 + Kanal\_ID

Index-Offset: IdxOffs = 0x4A

<b>Schreiben über Namen</b>	Mit dieser Funktion kann der Wert einer Variablen über ihren Namen geändert werden. Bei achsspezifischen Variablen muss der Achsname die Instanzkennung (Bsp. Achsnamen V.A.ACT_POS.X) beinhalten.
-----------------------------	--

Übergabeparameter (hier mit Angabe der Stream-Offsets):

0	1...256
Variablenname mit Nullterminierung	Variablenwert

wobei

Inhalt	Typ	Größe
Variablenname (mit Instanzkennung bei achsspezifischen Variablen)	String	variabel, max. 256 Byte
Variablenwert	String, REAL64, SGN32, REAL64[ ], SGN32[ ][ ], ...	variabel, max. 2047 Byte

=> Streamlänge max. 2048 Byte

Diese Funktion besitzt keinen Rückgabewert.

Bei einer Array-Variablen kann neben einem Einzelwert auch das komplette Feld geschrieben werden, in dem im Variablenamen keine Indizes angegeben werden.

Beispiel Variable V.G.array[0 .. 9]:

Variablenname	Variablenwert
V.G.array[4]	Wert des 5. Elements V.G.array[4]
V.G.array	Wert der Array-Elemente 0 - 9

#### Anwendungsbeispiele

1. Beispiel für das Schreiben eines Arrayelements eines eindimensionalen Arrays:

Übergabeparameter:

V.E.test[4]	0	123
-------------	---	-----

V.E.Testvar[4] Name der Variable

0 Nullterminierung

123 Neuer Wert des 5. Elements der Variablen

Rückgabewert: -

2. Beispiel für das Schreiben eines gesamten Arrays mit 5 Elementen:

V.E.test	0	123	234	345	456	567
----------	---	-----	-----	-----	-----	-----

V.E.Testvar[4] Name der Variable

0 Nullterminierung

123 Neuer Wert des 1. Elements der Variablen

234 Neuer Wert des 2. Elements der Variablen

345 Neuer Wert des 3. Elements der Variablen

456 Neuer Wert des 4. Elements der Variablen

567 Neuer Wert des 5. Elements der Variablen

Rückgabewerte: -

### 4.3.8 Funktion „Extended description by index“

Index-Gruppe: IdxGrp = 0x122300 + Kanal\_ID

Index-Offset: IdxOffs = 0x4B

<b>Erweiterte Informationen</b>	Diese Funktion liefert erweiterte Informationen über die Variable. Die übermittelten Daten sind von der jeweiligen Variablenart abhängig. Gültiger Wertebereich des Index ist 0 bis.(Anzahl der Elemente – 1)
---------------------------------	--

Übergabeparameter (hier mit Angabe der Stream-Offsets):

0	4
Index der Variablen	Bezeichner für Variablenart

wobei

Inhalt	Typ	Größe
Index der Variablen	UNS32	fest, 4 Byte
Bezeichner für Variablenart	STRING "V.G" - Globale Variablen "V.E" - Externe Variablen "V.A" - Achsspezifische Variablen	variabel, max. 256 Byte

=> Streamlänge max. 260 Byte

Rückgabewert: (hier mit Angabe der Stream-Offsets):

Der Rückgabewert ist abhängig von der Variablenart:

- a.) V.G, V.A Variablen -> kein Rückgabewert, da keine erweiterte Beschreibung vorhanden.
- b.) V.E. Variablen:

0	4
Synchronisation	Gültigkeitsbereich

wobei

Inhalt	Typ	Größe
Synchronisation	Enumeration (UNS32) 1 = NO_SYNC 2 = SYNCHRONIZED	fest, 4 Byte
Gültigkeitsbereich	Enumeration (UNS32) 1 = CHANNEL 2 = GLOBAL	fest, 4 Byte

=> Streamlänge V.E. Variable 8 Byte, max. 2048 Byte



## Hinweis

Der Bezeichner für die Variablenart z.B. V.E darf abschließend keinen Punkt haben.

### Anwendungsbeispiele

Beispiel für das Lesen der erweiterten Eigenschaften einer V.E. Variable:

Übergabeparameter:

3	V.E	0
---	-----	---

3 für die Variable mit dem Indexeintrag 3

V.E Bezeichner der Variablenart : V.E

0 Nullterminierung

Rückgabewert:

2	2
---	---

2 Synchronisation: SYNCHRONIZED

2 Gültigkeitsbereich: GLOBAL

### 4.3.9 Achsspezifische Funktion „Variable handle by name“

Index-Gruppe: IdxGrp = 0x122300 + Kanal\_ID

Index-Offset: IdxOffs = 0x10004 für Achse 1

= 0x20004 für Achse 2

<b>Handle über Namen</b>	Diese Funktion liefert das Handle für den angegebenen Variablenname zurück, der zu einer auf diesem Platz (Achsisindex) einsortierten Achse gehört. Durch Achstausch kann sich daher der über das Handle gelesene Variablenwert ändern.
--------------------------	---

Übergabeparameter (hier mit Angabe der Stream-Offsets)::

0
Variablenname

wobei

Inhalt	Typ	Größe
Variablenname (ohne Instanzkennung) mit Nullterminierung	STRING	variabel, max. 256 Byte

=> Streamlänge 256 Byte

Rückgabewert (hier mit Angabe der Stream-Offsets):

0 4
Handle auf die Variable

wobei

Inhalt	Typ	Größe
Handle auf die Variable	UNS32	fest, max. 4 Byte

=> Streamlänge max. 4 Byte



### Anwendungsbeispiele

Beispiel für das Lesen eines Handles einer achsspezifischen Variablen.  
Wichtig ist hierbei der Index-Offset.

Übergabeparameter:

Index-Offset: 0x20004

V.A.WCS	0
---------	---

V.A.WCS Name der Variablen (ohne Instanzkennung V.A.WCS.Y)

0 Nullterminierung

Rückgabewert:

0x0102001b
------------

0x0102001b Handle der Variablen hier der 2. Achse

### 4.3.10 Achsspezifische Funktion „Read variable value by name“

Index-Gruppe: IdxGrp = 0x122300 + Kanal\_ID

Index-Offset: IdxOffs = 0x10005 für Achse 1

= 0x20005 für Achse 2

<b>Lesen über Namen</b>	Mit dieser Funktion kann der Wert einer Variablen über ihren Variablennamen gelesen werden. Dabei wird immer auf die Variable zugegriffen, die zu der auf diesem Platz (Achsindex) gerade einsortierten Achse gehört.
-------------------------	---

Übergabeparameter (hier mit Angabe der Stream-Offsets):

0
Variablenname mit Nullterminierung

wobei

Inhalt	Typ	Größe
Variablenname (ohne Instanzkennung)	STRING	variabel, max. 256 Byte

=> Streamlänge 256 Byte

Rückgabewert (hier mit Angabe der Stream-Offsets):

0
Variablenwert

wobei

Inhalt	Typ	Größe
Variablenwert	String, REAL64, SGN32, REAL64[ ], SGN32[ ][ ], ...	variabel, max. 2048 Byte

=> Streamlänge max. 2048 Byte



#### Beispiel

#### Anwendungsbeispiele

Beispiel für das Lesen einer achsspezifischen Variablen. Es wird die Variable V.A.WCS.Y, also der 2. Achse gelesen mit dem Index-Offset 0x20005

Übergabeparameter:

V.A.WCS	0
---------	---

V.A.WCS Name der Variablen (ohne Instanzkennung V.A.WCS.Y)

0 Nullterminierung

Rückgabewert:

1234.2
--------

1234.2 Wert der Variablen der 2. Achse

### 4.3.11 Achsspezifische Funktion „Write variable value by name“

Index-Gruppe: IdxGrp = 0x122300 + Kanal\_ID

Index-Offset: IdxOffs = 0x10006 für Achse 1

= 0x20006 für Achse 2

<b>Schreiben über Namen</b>	Mit dieser Funktion kann der Wert einer Variablen über ihren Variablennamen geändert werden. Dabei wird immer auf die Variable zugegriffen, die zu der auf diesem Platz (Achsindex) gerade einsortierten Achse gehört.
-----------------------------	--

Übergabeparameter (hier mit Angabe der Stream-Offsets):

0	1-256
Variablenname mit Nullterminierung	Variablenwert

wobei

Inhalt	Typ	Größe
Variablenname z.B. V.A.WCS (ohne Instanzkennung V.A.WCS.Y)	String	variabel, max. 256 Byte
Variablenwert	String, REAL64, SGN32, REAL64[ ], SGN32[ ][ ], ...	variabel, max. 2047 Byte

=> Streamlänge max. 2048 Byte

Diese Funktion besitzt keinen Rückgabewert.

#### Anwendungsbeispiele

Beispiel für das Schreiben einer achsspezifischen Variablen. Es wird die Variable V.A.WCS.Y, also der 2. Achse geschrieben:

Index-Offset: 0x20006

Übergabeparameter:

V.A.WCS	0	123.456
---------	---	---------

V.A.WCS Name der Variablen (ohne Instanzkennung V.A.WCS.Y )

0 Nullterminierung

123.456 Neuer Wert der Variablen

Rückgabewert: -

## 5 ADS Dienste

### Interface für ADS-Dienste

Die CNC-Instanzen bieten ein Interface für den Zugriff auf die internen Elemente über folgende ADS-Dienste an:

Adressiert werden die Daten durch die Index Group und dem Index Offset.

### ADS Read

Mit *ADS Read* werden Daten aus einem ADS Gerät ausgelesen.

<b>In</b>	IndexGroup, 4 Byte, Index Group der Daten, die gelesen werden sollen. IndexOffset, 4 Byte, Index Offset der Daten, die gelesen werden sollen. Length, 4 Byte, Länge der Daten in Bytes, die gelesen werden sollen.
<b>Out</b>	Result, 4 Byte, ADS Fehlernummer. Length, 4 Byte, Länge der Daten in Bytes, die zurückgeliefert werden. Daten, n Byte, Daten die zurückgeliefert werden.

### ADS Write

Mit *ADS Write* werden Daten in ein ADS Gerät geschrieben.

<b>In</b>	IndexGroup, 4 Byte, Index Group der Daten, die gelesen werden sollen. IndexOffset, 4 Byte, Index Offset der Daten, die gelesen werden sollen. ReadLength, 4 Byte, Länge der Daten in Bytes, die gelesen werden sollen. WriteLength, 4 Byte, Länge der Daten in Bytes, die geschrieben werden sollen. Daten, n Byte, Daten die zu schreiben sind.
<b>Out</b>	Result, 4 Byte, ADS Fehlernummer.

### ADS Read & Write

Mit *ADS Read Write* werden Daten in ein ADS Gerät geschrieben. Zusätzlich können Daten aus dem ADS Gerät gelesen werden.

<b>In</b>	IndexGroup, 4 Byte, Index Group der Daten, die gelesen werden sollen. IndexOffset, 4 Byte, Index Offset der Daten, die gelesen werden sollen. ReadLength, 4 Byte, Länge der Daten in Bytes, die gelesen werden sollen. WriteLength, 4 Byte, Länge der Daten in Bytes, die geschrieben werden sollen. Daten, n Byte, Daten die zu schreiben sind.
<b>Out</b>	Result, 4 Byte, ADS Fehlernummer Length, 4 Byte, Länge der Daten in Bytes, die zurückgeliefert werden Daten, n Byte, Daten die zurückgeliefert werden

ADS – Automation Device Specification ( Quelle : Beckhoff Hilfe)

## 6 Anhang

Die nachfolgenden Kapitel sind aus Gründen der Abwärtskompatibilität aufgeführt. Sie beinhalten bisher unterstützte Schnittstellen. Diese werden auch weiterhin unterstützt, werden aber nicht empfohlen.

### 6.1 Teildynamische Elemente

V.A.-Variablen und V.G.-Variablen bilden eine weitere Gruppe von Elementen.

Eine Besonderheit bei diesen Variablentypen ist, dass der Name aus zwei Teilen bestehen kann. Einem statischen Teil und einem dynamischen Teil.

Der statische Teil der Variable wird übermittelt wenn der Name der Variablen über den Index gelesen wird. Der dynamische Teil besteht entweder aus einem Achsnamen des jeweiligen Kanals oder aus einem Arrayindex. In seltenen Fällen ist auch eine Kombination aus beidem möglich.

Der Achsname wird, mit einem Punkt getrennt, hinter dem Variablennamen (statischer Anteil) angehängt. Im Fall einer Arrayvariablen wird der maximal gültige Index in dem Namen übermittelt. Beim Zugriff ist dieser maximale Index durch den gewünschten zu ersetzen.



#### Beispiel

#### Teildynamische Elemente

##### Variable ergänzen um Achsname

V.A.Variable: V.A.+SWE\_MDS.X\_ACHSE

Statischer Anteil: V.A.+SWE\_MDS

Dynamischer Anteil: X\_ACHSE

##### Seltener Fall: Array Zugriff + Variable ergänzen um Achsname

V.G.Variable: V.G.NP[5].V.X\_ACHSE

Statischer Anteil: V.G.NP[5].V

Dynamischer Anteil: X\_ACHSE

##### Ergänzung um Achsname

Eine Ergänzung um den Achsnamen muss bei V.A.-Variablen immer stattfinden. Bei V.G.-Variablen gibt die Eigenschaft „V.G-node by index“ an, ob die Variable um einen Achsnamen zu ergänzen ist oder nicht. Ein Achsname ist anzuhängen wenn Node den Wert „V\_AXIS\_NAME“ hat. Ob ein Array vorhanden ist, lässt sich nur durch das Durchsuchen des Namens auf die Zeichen „[“ oder „]“ prüfen.

#### 6.1.1 V.A. Variablen

**Empfehlung:** Verwenden der Achsspezifische Schnittstelle (z.B. für V.A-Variablen) [► 22]

Gruppe	Offset	Bezeichner	Typ	Länge
0x122301	0x31	V.A-max. number	UNS32	4
0x122301	0x32	V.A-name by index	UNS32/STRING	96
0x122301	0x33	V.A-type by index	UNS32/STRING	96
0x122301	0x34	V.A-mode by index	UNS32/STRING	96
0x122301	0x37	V.A-read value by name	STRING/EXT_VAR_WERT	24
0x122301	0x38	V.A-write value by name	S_WRITE_VAR_DATA	120

**V.A-type by index Werte:**

Die Funktion liefert die folgenden Werte als String zurück

V\_BOOLEAN, V\_UNSO8, V\_SGN08, V\_UNSO16, V\_SGN16, V\_UNSO32, V\_SGN32, V\_REAL64,  
V\_STRING, V\_UNKNOWN\_TYPE

**V.A-mode by index Werte:**

Die Funktion liefert die folgenden Werte als String zurück:

NO\_ACCESS, UNKNOWN\_ACCESS, READ\_ONLY, WRITE\_ONLY, PROT\_ONLY,  
READ\_WRITE, READ\_WRITE\_PROT, READ\_PROT, WRITE\_PROT

## 6.1.2 V.G. Variablen

**Empfehlung: Verwenden der Kanalspezifische Schnittstelle [▶ 21]**

Gruppe	Offset	Bezeichner	Typ	Länge
0x122301	0x39	V.G-max. number	UNS32	4
0x122301	0x3A	V.G-name by index	UNS32/STRING	96
0x122301	0x3B	V.G-type by index	UNS32/STRING	96
0x122301	0x3C	V.G-mode by index	UNS32/STRING	96
0x122301	0x3D	V.G-node by index	UNS32/STRING	96
0x122301	0x3E	V.G-read value by name	STRING/EXT_VAR_WERT	24
0x122301	0x3F	V.G-write value by name	S_WRITE_VAR_DATA	120

### V.G-type by index Werte: siehe V.A.-Variablen

Die Funktion liefert die folgenden Werte als String zurück:

V\_BOOLEAN, V\_UNSO8, V\_SGN08, V\_UNSO16, V\_SGN16, V\_UNSO32, V\_SGN32, V\_REAL64, V\_STRING, V\_UNKNOWN\_TYPE

### V.G-mode by index Werte: siehe V.A.-Variablen

Die Funktion liefert die folgenden Werte als String zurück:

NO\_ACCESS, UNKNOWN\_ACCESS, READ\_ONLY, WRITE\_ONLY, PROT\_ONLY, READ\_WRITE, READ\_WRITE\_PROT, READ\_PROT, WRITE\_PROT

### V.G-node by index Werte:

Die Funktion liefert die folgenden Werte als String zurück:

V\_CONTINUE, V\_AXIS\_INDEX, V\_AXIS\_NAME, V\_CP\_NAME, V\_USER\_DEF, V\_USER\_DEF\_ARRAY, V\_ARRAY\_INDEX, V\_VALUE, UNKNOWN\_NODE

## 6.2 V.E. Variablen

**Empfehlung:** Verwenden der Kanalspezifische Schnittstelle [▶ 21]

Gruppe	Offset	Bezeichner	Typ	Länge
0x122301	0x27	V.E parameter number	UNS32	4
0x122301	0x28	V.E-dim by index	UNS32/UNS32	4
0x122301	0x29	V.E-name by index	UNS32/STRING	20
0x122301	0x2A	V.E-type by index	UNS32/STRING	20
0x122301	0x2B	V.E-address by index	UNS32/UNS32	4
0x122301	0x2C	V.E-mem-place... by index	UNS32/STRING	20
0x122301	0x2D	V.E-Synchron mode	UNS32/UNS32	4
0x122301	0x2E	V.E-value by index	UNS32/EXT_VAR_WERT	24
0x122301	0x2F	V.E-array-value by index	STRING/EXT_VAR_WERT	24
0x122301	0x30	V.E-Size by index	UNS32/UNS32	4

### 6.2.1 Zugriff V.E.-Arrayelement

Der Zugriff auf ein V.E.-Arrayelement wird mittels eines Strings durchgeführt. Dieser muss folgende Form haben.



#### Hinweis

Codierung des Strings  
 <Index der V.E.-Variable> @ <Index Arrayelement> @

Basis für das nachfolgende Beispiel :

Es soll die Variable A[2] ausgelesen werden, diese ist im Kanal 1 das 2. V.E. Element und hat den Index 2. Es wird das 3. Element des Arrays ausgelesen.



#### Beispiel

**Abfrage eine V.E.-Arrayelement:**

IndexGroup = 0x22301

Input : "2@2@" ASCII String codiert

Output : datentypabhängig (24 Byte) Datentyp + Rest = 24 Byte



#### Hinweis

Der Index der V.E.-Variablen startet mit 1, der Index der Arrayelemente mit 0.



## 6.3

### Anregungen, Korrekturen und neueste Dokumentation

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



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

Der o.g. Link ist eine Weiterleitung zu:

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



#### Hinweis

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

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

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

##### PDFs zum Download:

DE:

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

EN:

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

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

# Stichwortverzeichnis



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

