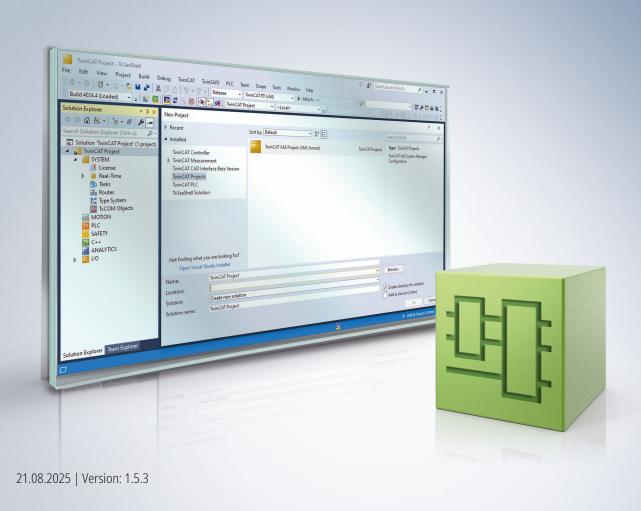
BECKHOFF New Automation Technology

Handbuch | DE

TE1000

TwinCAT 3 | PLC-Bibliothek: Tc2_IoFunctions





Inhaltsverzeichnis

1	Vorw			
	1.1	Hinweis	e zur Dokumentation	7
	1.2		Sicherheit	
	1.3	1.3 Hinweise zur Informationssicherheit		
2	Über	sicht		10
3	Funk	tionsbau	usteine	13
	3.1	Allgeme	ine IO FBs	13
		3.1.1	IOF_DeviceReset	13
		3.1.2	IOF_GetBoxAddrByName	14
		3.1.3	IOF_GetBoxAddrByNameEx	15
		3.1.4	IOF_GetBoxCount	16
		3.1.5	IOF_GetBoxNameByAddr	17
		3.1.6	IOF_GetBoxNetId	18
		3.1.7	IOF_GetDeviceCount	19
		3.1.8	IOF_GetDeviceIDByName	20
		3.1.9	IOF_GetDeviceIDs	21
		3.1.10	IOF_GetDeviceInfoByName	23
		3.1.11	IOF_GetDeviceName	24
		3.1.12	IOF_GetDeviceNetId	25
		3.1.13	IOF_GetDeviceType	26
	3.2	ASI-Mas	sterklemme	27
		3.2.1	Übersicht	27
		3.2.2	FB_ASI_Addressing	29
		3.2.3	FB_ASI_SlaveDiag	30
		3.2.4	FB_ASI_ReadParameter	31
		3.2.5	FB_ASI_WriteParameter	32
		3.2.6	FB_ASI_Processdata_digital	34
		3.2.7	FB_ASI_ParameterControl	35
		3.2.8	FB_ReadInput_analog	36
		3.2.9	FB_WriteOutput_analog	37
	3.3	AX200x	Profibus	39
		3.3.1	Übersicht	39
		3.3.2	FB_AX2000_Parameter	40
		3.3.3	FB_AX2000_AXACT	42
		3.3.4	FB_AX2000_JogMode	43
		3.3.5	FB_AX2000_Reference	44
		3.3.6	FB_AX200X_Profibus	46
	3.4	Beckhof	f Lightbus	48
		3.4.1	IOF_LB_BreakLocationTest	48
		3.4.2	IOF_LB_ParityCheck	
		3.4.3	IOF_LB_ParityCheckWithReset	
	3.5		ff USV (konfiguriert mit Windows USV service)	
		3.5.1	FB_GetUPSStatus	
	3.6	Busklem	_ nmen-Konfiguration	
		· · · · · · · · · · · · · · · · · · ·		



	3.6.1	FB_KL1501Config	55
	3.6.2	FB_KL27x1Config	58
	3.6.3	FB_KL320xConfig	61
	3.6.4	FB_KL3208Config	63
	3.6.5	FB_KL3228Config	66
3.7	CANope	en	68
	3.7.1	IOF_CAN_Layer2Command	68
3.8	NOV/DF	P-RAM	70
	3.8.1	FB_NovRamReadWrite	70
	3.8.2	FB_NovRamReadWriteEx	72
	3.8.3	FB_GetDPRAMInfo	74
	3.8.4	FB_GetDPRAMInfoEx	77
3.9	Profibus	DPV1 (Sinamics)	78
	3.9.1	F_CreateDpv1ReadReqPkg	78
	3.9.2	F_CreateDpv1WriteReqPkg	79
	3.9.3	F_SplitDpv1ReadResPkg	80
	3.9.4	F_SplitDpv1WriteResPkg	81
	3.9.5	FB_Dpv1Read	82
	3.9.6	FB_Dpv1Write	84
3.10	Profinet	DPV1 (Sinamics)	87
	3.10.1	F_CreateDpv1ReadReqPkgPNET	87
	3.10.2	F_CreateDpv1WriteReqPkgPNET	88
	3.10.3	F_SplitDpv1ReadResPkgPNET	89
	3.10.4	F_SplitDpv1WriteResPkgPNET	89
	3.10.5	FB_Dpv1ReadPNET	90
	3.10.6	FB_Dpv1WritePNET	93
3.11	RAID Co	ontroller	95
	3.11.1	FB_RAIDFindCntlr	95
	3.11.2	FB_RAIDGetInfo	96
	3.11.3	FB_RAIDGetStatus	98
3.12	SERCO	S	99
	3.12.1	IOF_SER_GetPhase	99
	3.12.2	IOF_SER_SaveFlash	100
	3.12.3	IOF_SER_ResetErr	101
	3.12.4	IOF_SER_SetPhase	102
	3.12.5	IOF_SER_IDN_Read	104
	3.12.6	IOF_SER_IDN_Write	105
	3.12.7	IOF_SER_DRIVE_Backup	107
	3.12.8	IOF_SER_DRIVE_BackupEx	109
	3.12.9	IOF_SER_DRIVE_Reset	112
3.13	TcTouch	nLock	113
	3.13.1	FB_TcTouchLock_AcquireFocus	113
3.14	Dritthers	stellergeräte	115
	3.14.1	Phoenix IBS SC/I-T	115
	3.14.2	ads-tec	127
Funk	tionen		129

4



	4.1	[veraltete Funktionen]	129
		4.1.1 F_GetVersionTcloFunctions	129
		4.1.2 F_GetVersionRAIDController	129
5	Dater	ntypen	131
	5.1	E_PD_Dpv1Error	131
	5.2	E_BatteryStatus	132
	5.3	E_PD_Datatype	132
	5.4	E_RAIDDriveStatus	133
	5.5	E_RAIDDriveUsage	134
	5.6	E_RAIDStatus	134
	5.7	E_RAIDType	135
	5.8	E_SercosAttribLen	135
	5.9	E_SercosAttribType	136
	5.10	E_UpsCommStatus	136
	5.11	E_UpsPowerStatus	137
	5.12	IODEVICETYPES	137
	5.13	ST_AdsTecSysData	139
	5.14	ST_Dpv1ParamAddrEx	140
	5.15	ST_Dpv1ValueHeaderEx	141
	5.16	ST_NovRamAddrInfo	142
	5.17	ST_NovRamAddrInfoEx	142
	5.18	ST_Parameter_IN	142
	5.19	ST_Parameter_OUT	143
	5.20	ST_ParameterBuffer	144
	5.21	ST_PD_Dpv1Error	144
	5.22	ST_PNET_CCDSTS	144
	5.23	ST_PNIOConfigRecord	145
	5.24	ST_PNIORecord	145
	5.25	ST_PNIOState	145
	5.26	ST_PZD_IN	146
	5.27	ST_PZD_OUT	146
	5.28	ST_RAIDCntlrFound	146
	5.29	ST_RAIDConfigReq	147
	5.30	ST_RAIDDriveStatus	147
	5.31	ST_RAIDInfo	147
	5.32	ST_RAIDStatusRes	148
	5.33	ST_SercosParamAttrib	148
	5.34	ST_SercosParamErrList	149
	5.35	ST_SercosParamList	150
	5.36	ST_UPSStatus	150
	5.37	ST_KL1501InData	153
	5.38	ST_KL1501OutData	154
	5.39	ST_KL27x1InData	154
	5.40	ST_KL27x1OutData	154
	5.41	ST_KL320xInData	155
		ST_KL320xOutData	



	5.43	ST_KL3208InData	155
	5.44	ST_KL3208OutData	155
	5.45	ST_KL3228InData	156
	5.46	ST_KL3228OutData	156
6	Globa	ale Konstanten	157
	6.1	Bibliotheksversion	157
7	Anha	ng	158
	7.1	SERCOS Dateiformat der Backup-Datei	158
	7.2	AX200x Profibus Parameternummer	159
	7.3	Fehlercodes	162
	7.4	ADS Return Codes	163
		Support und Service	

Version: 1.5.3



1 Vorwort

1.1 Hinweise zur Dokumentation

Diese Beschreibung wendet sich ausschließlich an ausgebildetes Fachpersonal der Steuerungs- und Automatisierungstechnik, das mit den geltenden nationalen Normen vertraut ist.

Zur Installation und Inbetriebnahme der Komponenten ist die Beachtung der Dokumentation und der nachfolgenden Hinweise und Erklärungen unbedingt notwendig.

Das Fachpersonal ist verpflichtet, stets die aktuell gültige Dokumentation zu verwenden.

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

Disclaimer

Diese Dokumentation wurde sorgfältig erstellt. Die beschriebenen Produkte 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.

Marken

Beckhoff®, ATRO®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, MX-System®, Safety over EtherCAT®, TC/BSD®, TwinCAT®, TwinCAT/BSD®, TwinSAFE®, XFC®, XPlanar® und XTS® sind eingetragene und lizenzierte Marken der Beckhoff Automation GmbH.

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



EtherCAT® ist eine eingetragene Marke und patentierte Technologie, lizenziert durch die Beckhoff Automation GmbH, Deutschland.

Copyright

© Beckhoff Automation GmbH & Co. KG, 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-, Gebrauchsmusteroder Geschmacksmustereintragung vorbehalten.

Fremdmarken

In dieser Dokumentation können Marken Dritter verwendet werden. Die zugehörigen Markenvermerke finden Sie unter: https://www.beckhoff.com/trademarks.



1.2 Zu Ihrer Sicherheit

Sicherheitsbestimmungen

Lesen Sie die folgenden Erklärungen zu Ihrer Sicherheit.

Beachten und befolgen Sie stets produktspezifische Sicherheitshinweise, die Sie gegebenenfalls an den entsprechenden Stellen in diesem Dokument vorfinden.

Haftungsausschluss

Die gesamten Komponenten werden je nach Anwendungsbestimmungen in bestimmten Hard- und Software-Konfigurationen ausgeliefert. Änderungen der Hard- oder Software-Konfiguration, die über die dokumentierten Möglichkeiten hinausgehen, sind unzulässig und bewirken den Haftungsausschluss der Beckhoff Automation GmbH & Co. KG.

Qualifikation des Personals

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

Signalwörter

Im Folgenden werden die Signalwörter eingeordnet, die in der Dokumentation verwendet werden. Um Personen- und Sachschäden zu vermeiden, lesen und befolgen Sie die Sicherheits- und Warnhinweise.

Warnungen vor Personenschäden

▲ GEFAHR

Es besteht eine Gefährdung mit hohem Risikograd, die den Tod oder eine schwere Verletzung zur Folge hat.

MARNUNG

Es besteht eine Gefährdung mit mittlerem Risikograd, die den Tod oder eine schwere Verletzung zur Folge haben kann.

⚠ VORSICHT

Es besteht eine Gefährdung mit geringem Risikograd, die eine mittelschwere oder leichte Verletzung zur Folge haben kann.

Warnung vor Umwelt- oder Sachschäden

HINWEIS

Es besteht eine mögliche Schädigung für Umwelt, Geräte oder Daten.

Information zum Umgang mit dem Produkt



Diese Information beinhaltet z. B.:

Handlungsempfehlungen, Hilfestellungen oder weiterführende Informationen zum Produkt.



1.3 Hinweise zur Informationssicherheit

Die Produkte der Beckhoff Automation GmbH & Co. KG (Beckhoff) sind, sofern sie online zu erreichen sind, mit Security-Funktionen ausgestattet, die den sicheren Betrieb von Anlagen, Systemen, Maschinen und Netzwerken unterstützen. Trotz der Security-Funktionen sind die Erstellung, Implementierung und ständige Aktualisierung eines ganzheitlichen Security-Konzepts für den Betrieb notwendig, um die jeweilige Anlage, das System, die Maschine und die Netzwerke gegen Cyber-Bedrohungen zu schützen. Die von Beckhoff verkauften Produkte bilden dabei nur einen Teil des gesamtheitlichen Security-Konzepts. Der Kunde ist dafür verantwortlich, dass unbefugte Zugriffe durch Dritte auf seine Anlagen, Systeme, Maschinen und Netzwerke verhindert werden. Letztere sollten nur mit dem Unternehmensnetzwerk oder dem Internet verbunden werden, wenn entsprechende Schutzmaßnahmen eingerichtet wurden.

Zusätzlich sollten die Empfehlungen von Beckhoff zu entsprechenden Schutzmaßnahmen beachtet werden. Weiterführende Informationen über Informationssicherheit und Industrial Security finden Sie in unserem https://www.beckhoff.de/secguide.

Die Produkte und Lösungen von Beckhoff werden ständig weiterentwickelt. Dies betrifft auch die Security-Funktionen. Aufgrund der stetigen Weiterentwicklung empfiehlt Beckhoff ausdrücklich, die Produkte ständig auf dem aktuellen Stand zu halten und nach Bereitstellung von Updates diese auf die Produkte aufzuspielen. Die Verwendung veralteter oder nicht mehr unterstützter Produktversionen kann das Risiko von Cyber-Bedrohungen erhöhen.

Um stets über Hinweise zur Informationssicherheit zu Produkten von Beckhoff informiert zu sein, abonnieren Sie den RSS Feed unter https://www.beckhoff.de/secinfo.



2 Übersicht

Die IO-Functions-Bibliothek beinhaltet Funktionsbausteine, mit denen Dienste/Funktionen auf den IO-Geräten (Feldbus-Master oder Slaves) ausgeführt werden können.

Allgemeine Gerätefunktionen

Name	Beschreibung
IOF DeviceReset [▶ 13]	Reset eines IO-Gerätes
IOF GetBoxAddrByName [▶ 14]	Die Feldbusadresse der Box über die Geräte-Id und die Box-Bezeichnung ermitteln
IOF GetBoxAddrByNameEx	Die Feldbusadresse der Box über die Geräte-Bezeichnung und die Box-
[<u>15</u>]	Bezeichnung ermitteln
IOF GetBoxCount [▶ 16]	Die Anzahl der Boxen lesen
IOF GetBoxNameByAddr [17]	Die Box-Bezeichnung anhand der Feldbusadresse der Box und der Geräte-Id lesen
IOF GetBoxNetId [▶ 18]	Die AmsNetId einer Box anhand der Feldbusadresse der Box und der Geräte-Id lesen
IOF GetDeviceCount [▶ 19]	Die Anzahl der IO-Geräte lesen
IOF GetDeviceIDByName	Die Geräte-Id anhand der Geräte-Bezeichnung ermitteln
[<u>_20]</u>	
IOF GetDeviceIDs [▶ 21]	Alle Geräte-Ids lesen
IOF_GetDeviceName [▶ 24]	Die Geräte-Bezeichnung anhand der Geräte-Id lesen
IOF_GetDeviceNetId [▶ 25]	Die AmsNetId anhand der Geräte-Id lesen
IOF GetDeviceType [▶ 26]	Den Geräte-Typ anhand der Geräte-Id lesen
IOF GetDeviceInfoByName	Die Geräte-Id und die AmsNetId anhand der Geräte-Bezeichnung ermitteln
[<u>23]</u>	

Feldbusspezifische und gerätespezifische Funktionen

CANopen

Name	Beschreibung
IOF CAN Layer2Command	Ein Layer 2 Kommando ausführen
[<u>\begin{align*}68]</u>	

Beckhoff Lightbus

Name	Beschreibung
IOF LB BreakLocationTest	Bruchstellen-Test des Lichtwellenleiterrings
[<u>_48]</u>	
IOF LB ParityCheck [▶ 49]	Parity-Zähler lesen
IOF LB ParityCheckWithReset	Parity-Zähler lesen und zurücksetzen
[<u>> 51</u>]	

SERCOS

Name	Beschreibung
IOF SER GetPhase [▶ 99]	Die aktuelle Phase lesen
IOF SER ResetErr [▶ 101]	Reset des Fehlerpuffers
IOF_SER_SaveFlash [▶ 100]	Parameter im Flash speichern
IOF SER SetPhase [▶ 102]	Die aktuelle Phase setzen

10 Version: 1.5.3 TE1000



Name	Beschreibung
IOF SER IDN Read [▶ 104]	Sercos-Drive-Parameter lesen
IOF SER IDN Write [▶ 105]	Sercos-Drive-Parameter schreiben
IOF SER DRIVE Backup [▶ 107]	Backup und Restore der Sercos-Drive-Parameter in/aus einer Datei
IOF SER DRIVE BackupEx [▶ 109]	Backup und Restore der Sercos-Drive-Parameter in/aus einer Datei (erweiterte Funktionalität)
IOF SER DRIVE Reset [▶ 112]	Drive-Reset eines Sercos-Drives per Kommando auf Parameter S-0-0099 (IDN99)

Profibus DPV1 (Sinamics

Name	Beschreibung
F CreateDpv1ReadReqPkg	DPV1 Telegramm für Parameterlesen erzeugen
[<u>> 78]</u>	
FB Dpv1Read [▶ 82]	DPV1 Telegramm für Parameterlesen senden
F SplitDpv1ReadResPkg [▶ 80]	DPV1 Antwort-Telegramm für Parameterlesen auswerten
F CreateDpv1WriteReqPkg	DPV1 Telegramm für Parameterschreiben erzeugen
[<u>> 79]</u>	
FB_Dpv1Write [84]	DPV1 Telegramm für Parameterschreiben senden
F SplitDpv1WriteResPkg [▶ 81]	DPV1 Antwort-Telegramm für Parameterschreiben auswerten

Profinet DPV1 (Sinamics)

Name	Beschreibung
F_CreateDpv1ReadReqPkgPNET	DPV1 Telegramm für Parameterlesen erzeugen
[<u>\begin{align*}87]</u>	
FB Dpv1ReadPNET [▶ 90]	DPV1 Telegramm für Parameterlesen senden
F_SplitDpv1ReadResPkgPNET	DPV1 Antwort-Telegramm für Parameterlesen auswerten
[<u>\begin{align*}89]</u>	
F_CreateDpv1WriteReqPkgPNET	DPV1 Telegramm für Parameterschreiben erzeugen
[<u>\\$88]</u>	
FB Dpv1WritePNET [▶ 93]	DPV1 Telegramm für Parameterschreiben senden
F_SplitDpv1WriteResPkgPNET	DPV1 Antwort-Telegramm für Parameterschreiben auswerten
[<u>\bar{89}</u>	

NOV/DP-RAM

Name	Beschreibung
FB NovRamReadWrite [> 70]	Daten in das NOV-RAM schreiben oder aus dem NOV-RAM lesen
FB_NovRamReadWriteEx [▶ 72]	Daten in das NOV-RAM schreiben oder aus dem NOV-RAM lesen. Überprüft ob ein spezielle Zugriffsart auf den Speicher notwendig ist und kopiert dementsprechend die Daten auf die korrekte Art (z-B. beim Zugriff auf das CX_9000 NOV-RAM).
FB_GetDPRAMInfo [> 74]	Den Adresspointer und die konfigurierte Größe vom NOV/DP-RAM lesen
FB GetDPRAMInfoEx [▶ 77]	Den Adresspointer und die konfigurierte Größe vom NOV/DP-RAM lesen (Erweiterung)

AX200x Profibus

Funktionsbausteine für den Zugriff auf den AX200X über den Profibus: <u>Übersicht [▶ 39]</u>.



ASI Master Terminal

Funktionsbausteine für den Zugriff auf eine ASI-Masterklemme: Übersicht [▶ 27].

Beckhoff USV (unter Windows USV Service)

Name	Beschreibung	
FB GetUPSStatus [> 52]	Den Status der USV aus der SPS lesen.	

Drittherstellergeräte

INTERBUS Phoenix IBS SC/I-T Funktionen

Phoenix IBS SC/I-T Funktionen: <u>Übersicht</u> [▶ 115].

Name	Beschreibung
SCIT ActivateConfiguration	Führt den Befehl Activate_Configuration aus
[<u>117</u>]	
SCIT DeactivateConfiguration	Führt den Befehl Deactivate_Configuration aus
[<u>118</u>]	
SCIT StartDataTransfer [119]	Führt den Befehl Start_Data_Transfer aus
SCIT_StopDataTransfer [▶ 120]	Führt den Befehl Stop_Data_Transfer aus
SCIT_AlarmStop [▶ 121]	Führt den Befehl Alarm_Stop aus
SCIT ControlActiveConfiguratio	Dient zur Beeinflussung der aktiven Konfiguration der Interbus-Teilnehmer.
<u>n [▶ 123]</u>	Dieses Kommando kann sowohl im Zustand PAR_READY als auch im Zustand ACTIVE und RUN ausgeführt werden. Hierüber können einzelne, abhängige und gruppierte Teilnehmer aktiviert und deaktiviert werden.
SCIT GetErrorInfo [▶ 124]	Liefert Fehlerart und Fehlerort eines Interbus-Teilnehmers nach einem Busfehler
SCIT_ConfDevErrAll [▶ 125]	Peripheriestörungen aller Geräte quittieren

ads-tec

Name	Beschreibung	
FB_ReadAdsTecSysData [> 127]	Liest die Systemdaten/Diagnosedaten	

RAID_Controller

Following function blocks are available for RAID controller services.

Name	Description
1 B 10 (1 B 1 11 1 G C 1 1 C 1 1 C 1 C 1 C 1 C 1 C	liefert die RAID-Controller Anzahl und die entsprechenden RAID-Controller IDs zurück
	liefert ein RAID Info, das die Anzahl der RAID-Controller-Sets enthält und die maximale Anzahl der RAID Antriebe pro Set.
	liefert den RAID-Set-Index, den RAID Typ, den RAID Status, die Anzahl der RAID Antriebe und den Status der RAID Antriebe zurück.

3 Funktionsbausteine

3.1 Allgemeine IO FBs

3.1.1 IOF_DeviceReset

```
IOF_DeviceReset

NETID T_AmsNetId BOOL BUSY

DEVICEID UDINT BOOL ERR

RESET BOOL UDINT ERRID

TMOUT TIME
```

Der Funktionsbaustein IOF_DeviceReset führt ein Reset eines IO-Gerätes (z.B. einer Feldbuskarte) durch. Die Funktion entspricht der Online-Reset-Funktion aus dem TwinCAT->I/O->Geräte->Gerät xyz-Kontextmenü.

Eingänge

```
VAR_INPUT

NETID : T_AmsNetId;

DEVICEID : UDINT;

RESET : BOOL;

TMOUT : TIME := DEFAULT_ADS_TIMEOUT;

END_VAR
```

Name	Тур	Beschreibung
NETID	T_AmsNetId	Hier kann die Netzwerkadresse des TwinCAT-Rechners angegeben werden, auf dem das ADS-Kommando ausgeführt werden soll. Für den lokalen Rechner kann auch ein Leerstring angegeben werden.
DEVICEID	UDINT	Über die Deviceld (Geräte-ID) wird das IO-Gerät spezifiziert, auf dem die Funktion ausgeführt werden soll. Die Geräte-IDs werden während der Hardware-Konfiguration von TwinCAT-System festgelegt.
RESET	BOOL	Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.
TMOUT	TIME	Gibt die Timeout-Zeit an, die bei der Ausführung des Kommandos nicht überschritten werden darf.

Ausgänge

```
VAR_OUTPUT
BUSY : BOOL;
ERR : BOOL;
ERRID : UDINT;
END_VAR
```

Name	Тур	Beschreibung
BUSY	BOOL	Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
ERR	BOOL	Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der BUSY-Ausgang zurückgesetzt wurde.
ERRID	UDINT	Liefert bei einem gesetzten ERR-Ausgang die ADS-Fehlernummer [• 163].



Entwicklungsumge- bung	Zielplattform		Einzubindende SPS-Bibliotheken (Kategorie- gruppe)
	PC oder CX (x86, x64, ARM)	Alle IO-Geräte	Tc2_loFunctions (IO)

3.1.2 IOF_GetBoxAddrByName

```
IOF_GetBoxAddrByName

NETID T_AmsNetId BOOL BUSY

DEVICEID UDINT BOOL ERR

BOXNAME T_MaxString UDINT ERRID

START BOOL

TMOUT TIME
```

Der Funktionsbaustein IOF_GetBoxAddrByName ermittelt die Feldbusadresse einer Box (Box = Slave, Modul, Station) anhand der Box-Bezeichnung und der Geräte-ID. Ist eine Feldbusadresse nicht vorhanden, dann liefert der Funktionsbaustein eine logische oder physikalische Adresse zurück (bei Beckhoff Lightbus ist es z.B. die physikalische Boxnummer im Lichtwellenleiter-Ring und bei Profibus die Stationsadresse). Die Box-Bezeichnung wird als ein String an den Funktionsbaustein übergeben und kann während der Konfiguration in TwinCAT System vom Benutzer festgelegt werden.

Eingänge

```
VAR_INPUT

NETID : T_AmsNetId;

DEVICEID : UDINT;

BOXNAME : T_MaxString;

START : BOOL;

TMOUT : TIME := DEFAULT_ADS_TIMEOUT;

END_VAR
```

Name	Тур	Beschreibung
NETID	T_AmsNetId	Hier kann die Netzwerkadresse des TwinCAT-Rechners angegeben werden, auf dem das ADS-Kommando ausgeführt werden soll. Für den lokalen Rechner kann auch ein Leerstring angegeben werden.
DEVICEID	UDINT	Über die Deviceld (Geräte-ID) wird das IO-Gerät spezifiziert, auf dem die Funktion ausgeführt werden soll. Die Geräte-IDs werden während der Hardware-Konfiguration von TwinCAT-System festgelegt.
BOXNAME	T_MaxString	Die Box-Bezeichnung als String.
START	BOOL	Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.
TMOUT	TIME	Gibt die Timeout-Zeit an, die bei der Ausführung des Kommandos nicht überschritten werden darf.

Ausgänge

```
VAR_OUTPUT

BUSY : BOOL;

ERR : BOOL;

ERRID : UDINT;

BOXADDR : UINT;
```

Name	Тур	Beschreibung	
BUSY		Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.	



Name	Тур	Beschreibung
ERR		Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der BUSY-Ausgang zurückgesetzt wurde.
ERRID	UDINT	Liefert bei einem gesetzten ERR-Ausgang die ADS-
		<u>Fehlernummer [▶ 163]</u> .
BOXADDR	UINT	Die Feldbusadresse oder logische Adresse der Box.

Entwicklungsumge- bung	Zielplattform		Einzubindende SPS-Bibliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Alle IO-Geräte	Tc2_loFunctions (IO)

3.1.3 IOF_GetBoxAddrByNameEx

```
IOF_GetBoxAddrByNameEx

NETID T_AmsNetId BOOL BUSY

DEVICENAME T_MaxString BOOL ERR

BOXNAME T_MaxString UDINT ERRID

START BOOL UINT BOXADDR

TMOUT TIME
```

Der Funktionsbaustein IOF_GetBoxAddrByNameEx ermittelt die Feldbusadresse einer Box (Box = Slave, Modul, Station) anhand der Box-Bezeichnung und der Geräte-Bezeichnung. Ist eine Feldbusadresse nicht vorhanden, dann liefert der Funktionsbaustein eine logische oder physikalische Adresse zurück (bei Beckhoff Lightbus ist es z.B. die physikalische Boxnummer im Lichtwellenleiter-Ring und bei Profibus die Stationsadresse). Die Box-Bezeichnung und Geräte-Bezeichnung werden als Strings an den Funktionsbaustein übergeben und können während der Konfiguration in TwinCAT System vom Benutzer festgelegt werden.

Eingänge

```
VAR_INPUT

NETID : T_AmsNetId;

DEVICENAME : T_MaxString;

BOXNAME : T_MaxString;

START : BOOL;

TMOUT : TIME := DEFAULT_ADS_TIMEOUT;

END VAR
```

Name	Тур	Beschreibung
NETID	T_AmsNetId	Hier kann die Netzwerkadresse des TwinCAT-Rechners angegeben werden, auf dem das ADS-Kommando ausgeführt werden soll. Für den lokalen Rechner kann auch ein Leerstring angegeben werden.
DEVICENAME	T_MaxString	Die Geräte-Bezeichnung eines IO-Gerätes als String.
BOXNAME	T_MaxString	Die Box-Bezeichnung als String.
START	BOOL	Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.
TMOUT	TIME	Gibt die Timeout-Zeit an, die bei der Ausführung des Kommandos nicht überschritten werden darf.

Ausgänge

7	AR_OUTPUT			
		:	: BOOL;	
	ERR	:	: BOOL;	



ERRID : UDINT; BOXADDR : UINT; END_VAR

Name	Тур	Beschreibung
BUSY	BOOL	Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
ERR	BOOL	Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der BUSY-Ausgang zurückgesetzt wurde.
ERRID	UDINT	Liefert bei einem gesetzten ERR-Ausgang die <u>ADS-</u> <u>Fehlernummer [</u> ▶ 163].
BOXADDR	UINT	Die Feldbusadresse oder logische Adresse der Box.

Voraussetzungen

Entwicklungsumge- bung	Zielplattform		Einzubindende SPS-Bibliotheken (Kategorie-gruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Alle IO-Geräte	Tc2_loFunctions (IO)

3.1.4 IOF_GetBoxCount



Der Funktionsbaustein IOF_GetBoxCount liest die Anzahl der konfigurierten und aktiven Boxen (Box = Slave, Modul, Station) eines IO-Gerätes.

Eingänge

VAR_INPUT

NETID : T_AmsNetId;

DEVICEID : UDINT;

START : BOOL;

TMOUT : TIME := DEFAULT_ADS_TIMEOUT;

END_VAR

Name	Тур	Beschreibung
NETID	T_AmsNetId	Hier kann die Netzwerkadresse des TwinCAT-Rechners angegeben werden, auf dem das ADS-Kommando ausgeführt werden soll. Für den lokalen Rechner kann auch ein Leerstring angegeben werden.
DEVICEID	UDINT	Über die Deviceld (Geräte-ID) wird das IO-Gerät spezifiziert, auf dem die Funktion ausgeführt werden soll. Die Geräte-IDs werden während der Hardware-Konfiguration von TwinCAT-System festgelegt.
START	BOOL	Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.
TMOUT	TIME	Gibt die Timeout-Zeit an, die bei der Ausführung des Kommandos nicht überschritten werden darf.



Ausgänge

```
VAR_OUTPUT
BUSY: BOOL;
ERR: BOOL;
ERRID: UDINT;
BOXCOUNT: UDINT;
END_VAR
```

Name	Тур	Beschreibung
BUSY	BOOL	Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
ERR	BOOL	Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der BUSY-Ausgang zurückgesetzt wurde.
ERRID	UDINT	Liefert bei einem gesetzten ERR-Ausgang die <u>ADS-</u> <u>Fehlernummer [▶ 163]</u> .
BOXCOUNT	UINT	Anzahl der Boxen

Voraussetzungen

Entwicklungsumge- bung	Zielplattform		Einzubindende SPS-Bibliotheken (Kategorie-gruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Alle IO-Geräte	Tc2_loFunctions (IO)

3.1.5 IOF_GetBoxNameByAddr

```
IOF_GetBoxNameByAddr

— NETID T_AmsNetId BOOL BUSY —
DEVICEID UDINT BOOL ERR —
BOXADDR UINT UDINT ERRID —
START BOOL T_MaxString BOXNAME —
TMOUT TIME
```

Der Funktionsbaustein IOF_GetBoxNameByAddr ermittelt die Box-Bezeichnung anhand der Geräte-Id und der Feldbusadresse einer Box (Box = Slave, Modul, Station). Ist eine Feldbusadresse nicht vorhanden, dann kann als Feldbusadresse an den Funktionsbaustein eine logische oder physikalische Adresse übergeben werden (bei Beckhoff Lightbus z. B. die physikalische Boxnummer im Lichtwellenleiter-Ring). Beim Erfolg liefert der Funktionsbaustein die im TwinCAT konfigurierte Box-Bezeichnung als String zurück.

Eingänge

```
VAR_INPUT

NETID : T_AmsNetId;

DEVICEID : UDINT;

BOXADDR : UINT;

START : BOOL;

TMOUT : TIME := DEFAULT_ADS_TIMEOUT;

END_VAR
```

Name	Тур	Beschreibung
NETID	_	Hier kann die Netzwerkadresse des TwinCAT-Rechners angegeben werden, auf dem das ADS-Kommando ausgeführt werden soll. Für den lokalen Rechner kann auch ein Leerstring angegeben werden.



Name	Тур	Beschreibung
DEVICEID	UDINT	Über die Deviceld (Geräte-ID) wird das IO-Gerät spezifiziert, auf dem die Funktion ausgeführt werden soll. Die Geräte-IDs werden während der Hardware-Konfiguration von TwinCAT-System festgelegt.
BOXADDR	UINT	Die Feldbusadresse der Box.
START	BOOL	Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.
TMOUT	TIME	Gibt die Timeout-Zeit an, die bei der Ausführung des Kommandos nicht überschritten werden darf.

Ausgänge

VAR_OUTPUT

BUSY : BOOL;
ERR : BOOL;
ERRID : UDINT;
BOXNAME : T_MaxString;

END VAR

Name	Тур	Beschreibung
BUSY	BOOL	Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
ERR	BOOL	Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der BUSY-Ausgang zurückgesetzt wurde.
ERRID	UDINT	Liefert bei einem gesetzten ERR-Ausgang die <u>ADS-</u> <u>Fehlernummer [▶ 163]</u> .
BOXNAME	T_MaxString	Die Box-Bezeichnung als String.

Voraussetzungen

Entwicklungsumge- bung	Zielplattform		Einzubindende SPS-Bibliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Alle IO-Geräte	Tc2_loFunctions (IO)

3.1.6 IOF_GetBoxNetId

```
IOF_GetBoxNetId

NETID T_AmsNetId BOOL BUSY

DEVICEID UDINT BOOL ERR

BOXADDR WORD UDINT ERRID

START BOOL T_AmsNetId BoxNetId

TMOUT TIME
```

Einigen Boxen (Slave-Module) kann eine AmsNetld während der Konfiguration im TwinCAT zugewiesen werden. Die AmsNetld kann dann benutzt werden, um auf der Box Firmware-Funktionen ausführen zu können. Der Funktionsbaustein IOF_GetBoxNetld ermittelt die TwinCAT Netzwerkadresse anhand der Geräte-ID des Masters und der Feldbusadresse oder logischen Adresse im Feldbus. Die Geräte-IDs werden während der Konfiguration vom TwinCAT-System festgelegt und können nicht vom Benutzer konfiguriert werden.

Eingänge

VAR_INPUT

NETID : T_AmsNetId;
DEVICEID : UDINT;



BOXADDR : WORD;
START : BOOL;
TMOUT : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR

Name	Тур	Beschreibung
NETID	T_AmsNetId	Hier kann die Netzwerkadresse des TwinCAT-Rechners angegeben werden, auf dem das ADS-Kommando ausgeführt werden soll. Für den lokalen Rechner kann auch ein Leerstring angegeben werden.
DEVICEID	UDINT	Über die Deviceld (Geräte-ID) wird das IO-Gerät spezifiziert, auf dem die Funktion ausgeführt werden soll. Die Geräte-IDs werden während der Hardware-Konfiguration von TwinCAT-System festgelegt.
BOXADDR	WORD	Die Feldbusadresse oder logische Adresse der Box (Slave-Modul) deren AmsNetld gelesen werden soll.
START	BOOL	Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.
TMOUT	TIME	Gibt die Timeout-Zeit an, die bei der Ausführung des Kommandos nicht überschritten werden darf.

Ausgänge

VAR_OUTPUT

BUSY : BOOL;
ERR : BOOL;
ERRID : UDINT;
BOXNetId : T_AmsNetId;

END_VAR

Name	Тур	Beschreibung
BUSY	BOOL	Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
ERR	BOOL	Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der BUSY-Ausgang zurückgesetzt wurde.
ERRID	UDINT	Liefert bei einem gesetzten ERR-Ausgang die <u>ADS-</u> <u>Fehlernummer [▶ 163]</u> .
BoxNetId	T AmsNetID	Die TwinCAT Netzwerkadresse der Box als String.

Voraussetzungen

Entwicklungsumge- bung	Zielplattform		Einzubindende SPS-Bibliotheken (Kategorie-gruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Alle IO-Geräte	Tc2_loFunctions (IO)

3.1.7 IOF_GetDeviceCount



Der Funktionsbaustein IOF_GetDeviceCount liest die Anzahl der konfigurierten und aktiven IO-Geräte.



Eingänge

```
VAR_INPUT

NETID : T_AmsNetId;
START : BOOL;
TMOUT : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

Name	Тур	Beschreibung
NETID	T_AmsNetId	Hier kann die Netzwerkadresse des TwinCAT-Rechners angegeben werden, auf dem das ADS-Kommando ausgeführt werden soll. Für den lokalen Rechner kann auch ein Leerstring angegeben werden.
START	BOOL	Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.
TMOUT	TIME	Gibt die Timeout-Zeit an, die bei der Ausführung des Kommandos nicht überschritten werden darf.

Ausgänge

VAR_OUTPUT
BUSY : BOOL;
ERR : BOOL;
ERRID : UDINT;
DEVICECOUNT : UDINT;

END VAR

Name	Тур	Beschreibung
BUSY	BOOL	Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
ERR	BOOL	Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der BUSY-Ausgang zurückgesetzt wurde.
ERRID	UDINT	Liefert bei einem gesetzten ERR-Ausgang die <u>ADS-Fehlernummer</u> [• 163].
DEVICECOUNT	UDINT	Anzahl der IO-Geräte

Voraussetzungen

Entwicklungsumge- bung	Zielplattform		Einzubindende SPS-Bibliotheken (Kategorie-gruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Alle IO-Geräte	Tc2_loFunctions (IO)

3.1.8 IOF_GetDeviceIDByName

	IOF_GetDeviceIDByName	
_	NETID T_AmsNetId BOOL BUSY	H
_	DEVICENAME T MaxString BOOL ERR	L
_	START BOOL UDINT ERRID	L
_	TMOUT TIME UDINT DEVICEID	H

Der Funktionsbaustein IOF_GetDeviceIDByName ermittelt die Geräte-Id eines IO-Gerätes anhand der Geräte-Bezeichnung. Beim Erfolg liefert der Funktionsbaustein die vom TwinCAT-System während der Konfiguration festgelegte Geräte-ID. Die Geräte-Ids können vom Benutzer nicht konfiguriert werden.



Eingänge

```
VAR_INPUT
   NETID : T_AmsNetId;
   DEVICENAME : T_MaxString;
   START : BOOL;
   TMOUT : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

Name	Тур	Beschreibung
NETID	T_AmsNetId	Hier kann die Netzwerkadresse des TwinCAT-Rechners angegeben werden, auf dem das ADS-Kommando ausgeführt werden soll. Für den lokalen Rechner kann auch ein Leerstring angegeben werden.
DEVICENAME	T_MaxString	Die Geräte-Bezeichnung eines IO-Gerätes.
START	BOOL	Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.
TMOUT	TIME	Gibt die Timeout-Zeit an, die bei der Ausführung des Kommandos nicht überschritten werden darf.

Ausgänge

VAR_OUTPUT

BUSY : BOOL;
ERR : BOOL;
ERRID : UDINT;
DEVICEID : UDINT;

END_VAR

Name	Тур	Beschreibung
BUSY	BOOL	Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
ERR	BOOL	Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der BUSY-Ausgang zurückgesetzt wurde.
ERRID	UDINT	Liefert bei einem gesetzten ERR-Ausgang die <u>ADS-Fehlernummer [• 163]</u> .
DEVICEID	UDINT	Die Geräte-ID eines IO-Gerätes.

Voraussetzungen

Entwicklungsumge- bung	Zielplattform		Einzubindende SPS-Bibliotheken (Kategorie-gruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Alle IO-Geräte	Tc2_loFunctions (IO)

3.1.9 IOF_GetDeviceIDs



Der Funktionsbaustein IOF_GetDeviceIDs liest die Geräte-IDs aller konfigurierten und aktiven IO-Geräte in einen Datenpuffer ein. Der Datenpuffer kann als ein Array von Word-Variablen definiert werden. Beim Erfolg liefert der Funktionsbaustein im ersten Daten-Word die gesamte Anzahl der vorhandenen Geräte-IDs und in



den weiteren Daten-Worten die entsprechenden Geräte-IDs der einzelnen IO-Geräte. Die Geräte-IDs werden während der Konfiguration vom TwinCAT-System festgelegt und können nicht vom Benutzer konfiguriert werden.

Eingänge

```
VAR_INPUT

NETID : T_AmsNetId;

LEN : UDINT;

DESTADDR : PVOID;

START : BOOL;

TMOUT : TIME := DEFAULT_ADS_TIMEOUT;

END_VAR
```

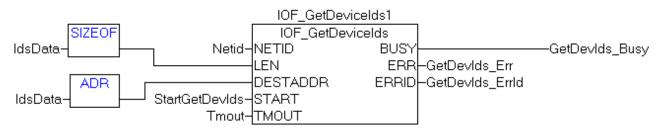
Name	Тур	Beschreibung
NETID	T_AmsNetId	Hier kann die Netzwerkadresse des TwinCAT-Rechners angegeben werden, auf dem das ADS-Kommando ausgeführt werden soll. Für den lokalen Rechner kann auch ein Leerstring angegeben werden.
LEN	UINT	Länge des Datenpuffers in Bytes in den die Geräte-IDs eingelesen werden sollen.
DESTADDR	PVOID	Adresse des Datenpuffers in den die Geräte-IDs eingelesen werden sollen.
START	BOOL	Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.
TMOUT	TIME	Gibt die Timeout-Zeit an, die bei der Ausführung des Kommandos nicht überschritten werden darf.

Beispiel:

```
PROGRAM MAIN

VAR

IOF_GetDeviceIds1: IOF_GetDeviceIDs;
IdsData : ARRAY[1..201] OF WORD;
StartGetDevIds : BOOL;
GetDevIds_Busy : BOOL;
GetDevIds_Err : BOOL;
GetDevIds_ErrId : UDINT;
END VAR
```



Ausgänge

VAR_OUTPUT

BUSY : BOOL;

ERR : BOOL;

ERRID : UDINT;

END_VAR

Name	Тур	Beschreibung
BUSY	BOOL	Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
ERR	BOOL	Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der BUSY-Ausgang zurückgesetzt wurde.



Name	Тур	Beschreibung
ERRID	UDINT	Liefert bei einem gesetzten ERR-Ausgang die ADS-
		<u>Fehlernummer [▶ 163]</u> .

	Zielplattform		Einzubindende SPS-Bibliotheken (Kategorie-
bung			gruppe)
	PC oder CX (x86, x64, ARM)	Alle IO-Geräte	Tc2_loFunctions (IO)

3.1.10 IOF_GetDeviceInfoByName

```
IOF_GetDeviceInfoByName

NETID T_AmsNetId BOOL BUSY

DEVICENAME T_MaxString BOOL ERR

START BOOL
TMOUT TIME UDINT DEVICEID

T_AmsNetId DEVICENETID
```

Der Funktionsbaustein IOF_GetDeviceInfoByName ermittelt die Geräte-ID eines IO-Gerätes und dessen TwinCAT Netzwerkadresse anhand der Geräte-Bezeichnung. Die Geräte-Ids können vom Benutzer nicht konfiguriert werden.

Eingänge

```
VAR_INPUT

NETID : T_AmsNetId;

DEVICENAME : T_MaxString;

START : BOOL;

TMOUT : TIME := DEFAULT_ADS_TIMEOUT;

END_VAR
```

Name	Тур	Beschreibung
NETID		Hier kann die Netzwerkadresse des TwinCAT-Rechners angegeben werden, auf dem das ADS-Kommando ausgeführt werden soll. Für den lokalen Rechner kann auch ein Leerstring angegeben werden.
DEVICENAME	T_MaxString	Die Geräte-Bezeichnung eines IO-Gerätes.
START	BOOL	Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.
TMOUT	TIME	Gibt die Timeout-Zeit an, die bei der Ausführung des Kommandos nicht überschritten werden darf.

Ausgänge

```
VAR_OUTPUT

BUSY : BOOL;

ERR : BOOL;

ERRID : UDINT;

DEVICEID : UDINT;

DEVICENETID : T_AmsNetId;

END_VAR
```

Name	Тур	Beschreibung
BUSY		Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.



Name	Тур	Beschreibung
ERR		Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der BUSY-Ausgang zurückgesetzt wurde.
ERRID	UDINT	Liefert bei einem gesetzten ERR-Ausgang die ADS-
		<u>Fehlernummer [▶ 163]</u> .
DEVICEID	UDINT	Die Geräte-ID eines IO-Gerätes.
DEVICENETID	T_AmsNetID	Die Netzwerkadresse eines IO-Gerätes.

Entwicklungsumge- bung	Zielplattform		Einzubindende SPS-Bibliotheken (Kategoriegruppe)
	PC oder CX (x86, x64, ARM)	Alle IO-Geräte	Tc2_loFunctions (IO)

3.1.11 IOF_GetDeviceName

```
IOF_GetDeviceName

NETID T_AmsNetId BOOL BUSY

DEVICEID UDINT BOOL ERR

START BOOL UDINT ERRID

TMOUT TIME T_MaxString DEVICENAME
```

Der Funktionsbaustein IOF_GetDeviceName liest die Gerätebezeichnung eines IO-Gerätes. Die Gerätebezeichnung kann während der Konfiguration in TwinCAT-System vom Benutzer festgelegt werden. Beim Systemstart wird diese dann als String in den IO-Treiber gesendet und kann über die ADS-Kommandos gelesen werden. Über die Eingangsvariable DEVICEID wird das IO-Gerät spezifiziert, dessen Gerätebezeichnung gelesen werden soll.

Eingänge

```
VAR_INPUT

NETID : T_AmsNetId;

DEVICEID : UDINT;

START : BOOL;

TMOUT : TIME := DEFAULT_ADS_TIMEOUT;

END_VAR
```

Name	Тур	Beschreibung
NETID	T_AmsNetId	Hier kann die Netzwerkadresse des TwinCAT-Rechners angegeben werden, auf dem das ADS-Kommando ausgeführt werden soll. Für den lokalen Rechner kann auch ein Leerstring angegeben werden.
DEVICEID	UDINT	Über die Deviceld (Geräte-ID) wird das IO-Gerät spezifiziert, auf dem die Funktion ausgeführt werden soll. Die Geräte-IDs werden während der Hardware-Konfiguration von TwinCAT-System festgelegt.
START	BOOL	Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.
TMOUT	TIME	Gibt die Timeout-Zeit an, die bei der Ausführung des Kommandos nicht überschritten werden darf.

Ausgänge

VAR_OUTPUT			
BUSY	: BOOL;		
ERR	: BOOL;		



ERRID : UDINT;
DEVICENAME : T_MaxString;
END VAR

Name	Тур	Beschreibung
BUSY	BOOL	Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
ERR	BOOL	Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der BUSY-Ausgang zurückgesetzt wurde.
ERRID	UDINT	Liefert bei einem gesetzten ERR-Ausgang die <u>ADS-</u> <u>Fehlernummer [▶ 163]</u> .
DEVICENAME	T_MaxString	Die Geräte-Bezeichnung eines IO-Gerätes als String.

Voraussetzungen

Entwicklungsumge- bung	Zielplattform		Einzubindende SPS-Bibliotheken (Kategorie-gruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Alle IO-Geräte	Tc2_loFunctions (IO)

3.1.12 IOF_GetDeviceNetId

	IOF_	_GetDeviceNetId
_	NETID T_AmsNetId	BOOL BUSY
_	DEVICEID UDINT	BOOL ERR
_	START BOOL	UDINT ERRID
_	TMOUT TIME	T_AmsNetId DeviceNetId

Einigen IO-Geräten kann eine TwinCAT Netzwerkadresse während der Konfiguration im TwinCAT-System zugewiesen werden (z.B. FC310x Profibuskarte oder CP9030-Karte). Die Netzwerkadresse kann dann benutzt werden, um auf dem Gerät Firmware-Funktionen ausführen zu können. Der Funktionsbaustein IOF_GetDeviceNetId ermittelt die Netzwerkadresse anhand der Geräte-ID. Die Geräte-IDs werden während der Konfiguration vom TwinCAT-System festgelegt und können nicht vom Benutzer konfiguriert werden.

Eingänge

VAR_INPUT

NETID : T_AmsNetId;

DEVICEID : UDINT;

START : BOOL;

TMOUT : TIME := DEFAULT_ADS_TIMEOUT;

END_VAR

Name	Тур	Beschreibung
NETID	T_AmsNetId	Hier kann die Netzwerkadresse des TwinCAT-Rechners angegeben werden, auf dem das ADS-Kommando ausgeführt werden soll. Für den lokalen Rechner kann auch ein Leerstring angegeben werden.
DEVICEID	UDINT	Über die Deviceld (Geräte-ID) wird das IO-Gerät spezifiziert, auf dem die Funktion ausgeführt werden soll. Die Geräte-IDs werden während der Hardware-Konfiguration von TwinCAT-System festgelegt.
START	BOOL	Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.
TMOUT	TIME	Gibt die Timeout-Zeit an, die bei der Ausführung des Kommandos nicht überschritten werden darf.



Ausgänge

VAR_OUTPUT

BUSY : BOOL;
ERR : BOOL;
ERRID : UDINT;
DeviceNetId : T_AmsNetId;
END VAR

Name	Тур	Beschreibung
BUSY	BOOL	Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
ERR	BOOL	Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der BUSY-Ausgang zurückgesetzt wurde.
ERRID	UDINT	Liefert bei einem gesetzten ERR-Ausgang die <u>ADS-</u> <u>Fehlernummer [▶ 163]</u> .
DeviceNetId	T_AmsNetID	Die Netzwerkadresse eines IO-Gerätes als String.

Voraussetzungen

Entwicklungsumge- bung	Zielplattform		Einzubindende SPS-Bibliotheken (Kategorie-gruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Alle IO-Geräte	Tc2_loFunctions (IO)

3.1.13 IOF_GetDeviceType

```
IOF_GetDeviceType

NETID T_AmsNetId BOOL BUSY—
DEVICEID UDINT BOOL ERR—
START BOOL UDINT ERRID—
TMOUT TIME IODEVICETYPES IODEViceType—
```

Der Funktionsbaustein "IOF_GetDeviceType" ermittelt den Geräte-Typ anhand der Geräte-ID. Die Geräte-IDs werden während der Konfiguration vom TwinCAT-System festgelegt und können nicht vom Benutzer konfiguriert werden.

Eingänge

VAR_INPUT

NETID : T_AmsNetId;

DEVICEID : UDINT;

START : BOOL;

TMOUT : TIME := DEFAULT_ADS_TIMEOUT;

END_VAR

Name	Тур	Beschreibung
NETID	T_AmsNetId	Hier kann die Netzwerkadresse des TwinCAT-Rechners angegeben werden, auf dem das ADS-Kommando ausgeführt werden soll. Für den lokalen Rechner kann auch ein Leerstring angegeben werden.
DEVICEID	UDINT	Über die Deviceld (Geräte-ID) wird das IO-Gerät spezifiziert, auf dem die Funktion ausgeführt werden soll. Die Geräte-IDs werden während der Hardware-Konfiguration von TwinCAT-System festgelegt.
START	BOOL	Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.



Name	Тур	Beschreibung
TMOUT	TIME	Gibt die Timeout-Zeit an, die bei der Ausführung des
		Kommandos nicht überschritten werden darf.

Ausgänge

VAR_OUTPUT

BUSY : BOOL;
ERR : BOOL;
ERRID : UDINT;
IODeviceType : IODEVICETYPES;

END VAR

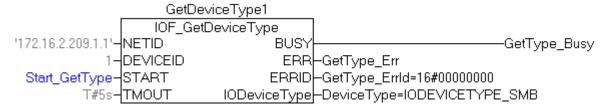
Name	Тур	Beschreibung
BUSY	BOOL	Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
ERR	BOOL	Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der BUSY-Ausgang zurückgesetzt wurde.
ERRID	UDINT	Liefert bei einem gesetzten ERR-Ausgang die <u>ADS-</u> <u>Fehlernummer [</u> ▶ 163].
IODeviceType	IODEVICETYPES [▶ 137]	Die Geräte-Typ-Konstante.

Beispiel:

```
PROGRAM MAIN

VAR

GetDeviceType1 : IOF_GetDeviceType;
Start_GetType : BOOL;
GetType_Busy : BOOL;
GetType_Err : BOOL;
GetType_Err : BOOL;
GetType_Errid : UDINT;
DeviceType : IODEVICETYPES;
END_VAR
```



Voraussetzungen

Entwicklungsumge- bung	Zielplattform		Einzubindende SPS-Bibliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Alle IO-Geräte	Tc2_loFunctions (IO)

3.2 ASI-Masterklemme

3.2.1 Übersicht

Funktionsbausteine für den Zugriff auf die ASI-Masterklemme.

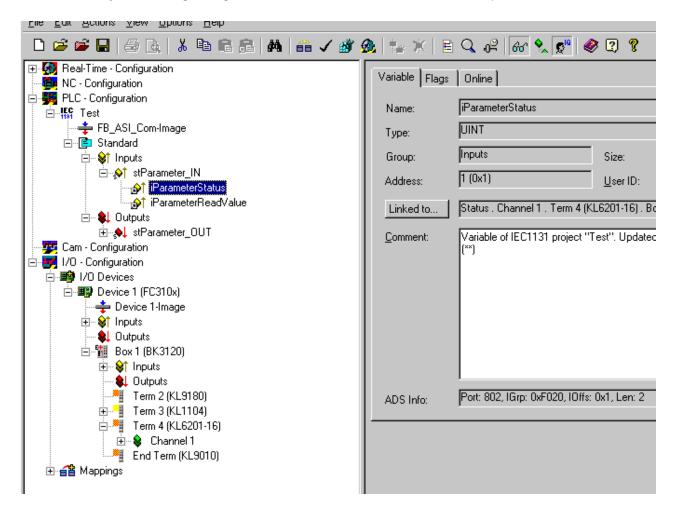


Funktionsbausteine:

Name	Beschreibung
FB_ASI_Addressing [> 29]	Festlegen oder Verändern von Adressen der ASI- Slaves
FB ASI SlaveDiag [> 30]	Zyklische Slave-Diagnose (z.B. Zählerstände)
FB ASI ReadParameter [▶ 31]	Universeller FB zum Auslesen aller Parameter eines ASI-Slaves
FB_ASI_WriteParameter [> 32]	Universeller FB zum Setzen aller Parameter eines ASI-Slaves
FB ReadInput analog [> 36]	Lesen von analogen Werten
FB WriteOutput analog [▶ 37]	Schreiben von analogen Werten
FB ASI Processdata digital [> 34]	Lesen/Schreiben von digitalen Werten
FB ASI ParameterControl [▶ 35]	Hintergrundkommunikation Dieser Baustein muss immer zyklisch aufgerufen werden!!!

Einbinden in den System Manager

Die Bibliothek besitzt eine Eingangsstruktur: ST_Parameter_IN und eine Ausgangsstruktur: ST_Parameter_OUT. Diese müssen instanziert und adressiert werden, um sie einerseits dem FB_ParameterControl als VAR_IN_OUT übergeben zu können und andererseits im System Manager verknüpft zu werden. Die Prozessdaten der Klemme beinhalten 6Byte und 16Byte, je nachdem, welches ASI-Modul im System Manager eingebunden wurde. Diese können direkt verknüpft werden.





FB_ASI_Addressing 3.2.2

FB_ASI_Addressing iOldAddress BYTE BOOL bBusy iNewAddress BYTE BOOL bErr bStart BOOL DWORD iErrornumber stParameterBuffer ST_ParameterBuffer

Eingänge

VAR_INPUT

iOldAddress : BYTE; (*old address*)
iNewAddress : BYTE; (*new address*)
bStart : BOOL; (*START*)
END_VAR

Name	Тур	Beschreibung
iOldAddress	BYTE	Alte Adresse des zu adressierenden Slaves (neue Slaves haben die Adresse 0).
iNewAddress	BYTE	Neue Adresse des zu adressierenden Slaves.
bStart	BOOL	Mit einer positiven Flanke an diesem boolschen Eingang wird die Adressierung vorgenommen.

▼/ ■ Ein-/Ausgänge

stParameterBuffer : ST_ParameterBuffer; END_VAR

Name	Тур	Beschreibung
stParameterBuffer	ST_ParameterBuffer	Datenpuffer für die Hintergrundkommunikation.
	[<u>\bar{144}</u>]	

Ausgänge

VAR OUTPUT

bBusy : BOOL; bErr : BOOL; bErrornumber : DWORD;

END VAR

Name	Тур	Beschreibung
bBusy	BOOL	Dieser Ausgang bleibt solange auf TRUE, bis der Baustein eine Befehlsanforderung ausführt. Während Busy = TRUE wird an den Eingängen kein neuer Befehl angenommen. Beachten Sie, dass nicht die Ausführung des Dienstes, sondern nur dessen Annahme zeitlich überwacht wird.
bErr	BOOL	Dieser Ausgang zeigt den Fehlerstatus an.
bErrornumber	DWORD	Enthält den befehlsspezifischen Fehlercode des zuletzt ausgeführten Befehls. Wird durch das Ausführen eines Befehls an den Eingängen auf 0 zurückgesetzt.

Befehlsspezifischer Fehlercode (dezimal)	Beschreibung
1	Kommunikations-Timeout
2	ASI-Slaveadresse nicht vorhanden
3 - 10	Reserviert
11	ASI-Slave ist nicht aktiviert (Slave ist nicht in LAS).
12	Bei der Kommunikation ist ein Fehler aufgetreten.
13	Datenaustauschbit (CN.4) nicht gesetzt



Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	PC oder CX (x86)	ASI Masterklemme	Tc2_loFunctions (IO)

3.2.3 FB_ASI_SlaveDiag

```
FB_ASI_SlaveDiag

iSlaveaddress BYTE
iCounter INT
bCounterReset BOOL
bErr
bReadLES BOOL
bReadLAS BOOL
bReadLAS BOOL
bReadLAS BOOL
bCycleMode BOOL
bStart BOOL
stParameterBuffer
```

Eingänge

```
VAR_INPUT

iSlaveaddress: BYTE;
iCounter: INT;
bReadLES: BOOL;
bReadLAS: BOOL;
bCyleMode: BOOL;
bStart: BOOL;
END_VAR
```

Name	Тур	Beschreibung
iSlaveaddress	BYTE	Slave Adresse
iCounter	INT	1:PhysicalFaultCounter, 2:TimeoutCounter, 3:ResponseCounter, 4:Leave-DataExchCounter, 5:DataExch-FailedCounter.
bCounterReset	BOOL	Rücksetzen des aktuellen Zählers.
bReadLES	BOOL	Liste der erkannten ASI-Slaves(LES).
bReadLAS	BOOL	Liste der aktivierten ASI-Slaves(LAS).
bCycleMode	BOOL	0=continuous reading, 1= reading once.
bStart	BOOL	Mit einer positiven Flanke an diesem boolschen Eingang wird der entsprechende Auftrag ausgeführt.

▼/ Ein-/Ausgänge

```
VAR_IN_OUT
stParameterBuffer : ST_ParameterBuffer;
END_VAR
```

Name	Тур	Beschreibung
stParameterBuffer	ST_ParameterBuffer	Datenpuffer für die Hintergrundkommunikation.
	[<u>\bar{144}</u>]	

Ausgänge

```
VAR_OUTPUT

bBusy : BOOL;
bErr : BOOL;
iErrornumber : DWORD; (* Error code of ASI-Master *)
iCounterValue : WORD; (*Counter of a slave*)
iSlaveList : DWORD; (*LES or LAS of all Slaves*)

END_VAR
```



Name	Тур	Beschreibung
bBusy	BOOL	Dieser Ausgang bleibt solange auf TRUE, bis der Baustein eine Befehlsanforderung ausführt. Während Busy = TRUE wird an den Eingängen kein neuer Befehl angenommen. Beachten Sie, dass nicht die Ausführung des Dienstes, sondern nur dessen Annahme zeitlich überwacht wird.
bErr	BOOL	Dieser Ausgang zeigt den Fehlerstatus an.
bErrornumber	DWORD	Enthält den befehlsspezifischen Fehlercode des zuletzt ausgeführten Befehls. Wird durch das Ausführen eines Befehls an den Eingängen auf 0 zurückgesetzt.

Befehlsspezifischer Fehlercode (dezimal)	Beschreibung
1	Kommunikations-Timeout
2	ASI-Slaveadresse nicht vorhanden
3 - 10	Reserviert
11	ASI-Slave ist nicht aktiviert (Slave ist nicht in LAS).
12	Bei der Kommunikation ist ein Fehler aufgetreten.
13	Datenaustauschbit (CN.4) nicht gesetzt

Name	Тур	Beschreibung
iCountervalue	WORD	Zählerstand
iSlaveList	DWORD	LES bzw. LAS

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	PC oder CX (x86)	ASI Masterklemme	Tc2 IoFunctions (IO)

3.2.4 FB_ASI_ReadParameter

	FB_ASI_ReadParameter			
_	iParameternumber	WORD	BOOL bBusy -	_
_	bCycleMode BOOL		BOOL bErr	_
_	bStart BOOL		DWORD iErrornumber	_
_	stParameterBuffer	ST_ParameterBuffer	DWORD iParameterReadvalue	_

Eingänge

VAR_INPUT
iParameternumber : WORD;
bCycleMode : BOOL; (*0: Acyclic , 1:Cyclic (permanent Read/Write) *)
bStart : BOOL;
END_VAR

Name	Тур	Beschreibung
iParameterNumber	WORD	Parameternummer.
bCycleMode	BOOL	0: Acyclic , 1:Cyclic (permanent Read/Write) Ist dieses Bit gesetzt, wird der Ausgang bBusy erst zurückgenommen, wenn der Eingang bStart auf FALSE gezogen wird. Wird der Eingang bStart zu früh auf FALSE gezogen, steht noch kein aktueller Wert am Ausgang an.
bStart	BOOL	Mit einer positiven Flanke an diesem boolschen Eingang wird der entsprechende Auftrag ausgeführt.



▼/ ■ Ein-/Ausgänge

VAR_IN_OUT ____stParameterBuffer : ST_ParameterBuffer; END_VAR

Name	Тур	Beschreibung
stParameterBuffer	ST ParameterBuffer	Datenpuffer für die Hintergrundkommunikation.
	[<u>\bar{144}</u>	

Ausgänge

VAR_OUTPUT

bBusy : BOOL; bErr

bErr : BOOL; iErrornumber : DWORD; (* Error code of ASI-Master *)

iParameterReadvalue : BYTE;

END_VAR

Name	Тур	Beschreibung
bBusy	BOOL	Dieser Ausgang bleibt solange auf TRUE, bis der Baustein eine Befehlsanforderung ausführt. Während Busy = TRUE wird an den Eingängen kein neuer Befehl angenommen. Beachten Sie, dass nicht die Ausführung des Dienstes, sondern nur dessen Annahme zeitlich überwacht wird.
bErr	BOOL	Dieser Ausgang zeigt den Fehlerstatus an.
bErrornumber	DWORD	Enthält den befehlsspezifischen Fehlercode des zuletzt ausgeführten Befehls. Wird durch das Ausführen eines Befehls an den Eingängen auf 0 zurückgesetzt.

Befehlsspezifischer Fehlercode (dezimal)	Beschreibung
1	Kommunikations-Timeout
2 ASI-Slaveadresse nicht vorhanden	
3 - 10	Reserviert
11	ASI-Slave ist nicht aktiviert (Slave ist nicht in LAS).
12	Bei der Kommunikation ist ein Fehler aufgetreten.
13	Datenaustauschbit (CN.4) nicht gesetzt

Name	Тур	Beschreibung
iParameterReadvalu	BYTE	E/A-Kennung bzw. ID-Code des angesprochenen Slaves.
е		

Voraussetzungen

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	PC oder CX (x86)	ASI Masterklemme	Tc2_loFunctions (IO)

3.2.5 FB_ASI_WriteParameter

	FB_ASI_WriteParameter		
_	iParameternumber WORD	BOOL bBusy	—
_	iParametervalue DWORD	BOOL bErr	—
_	-bStart BOOL DWORL	iErrornumber –	—
_	-stParameterBuffer ST_ParameterBuffer		



Eingänge

VAR_INPUT

____iParameternumber : WORD; iParametervalue : DWORD; bStart : BOOL; OVAR END_VAR

Name	Тур	Beschreibung	
iParameterNumber	WORD	Parameternummer	
iParametervalue	DWORD	Parameterwert	
bStart		Mit einer positiven Flanke an diesem boolschen Eingang wird der entsprechende Auftrag ausgeführt.	

▼/ ■ Ein-/Ausgänge

VAR_IN_OUT stParameterBuffer : ST_ParameterBuffer; END_VAR

Name	Тур	Beschreibung
stParameterBuffer	ST_ParameterBuffer	Datenpuffer für die Hintergrundkommunikation.
	[<u>\bar{144}</u>	

Ausgänge

VAR_OUTPUT

bBusy : BOOL; bErr : BOOL; bErrornumber : DWORD;

END_VAR

Name	Тур	Beschreibung	
bBusy	BOOL	Dieser Ausgang bleibt solange auf TRUE, bis der Baustein eine Befehlsanforderung ausführt. Während Busy = TRUE wird an den Eingängen kein neuer Befehl angenommen. Beachten Sie, dass nicht die Ausführung des Dienstes, sondern nur dessen Annahme zeitlich überwacht wird.	
bErr	BOOL	Dieser Ausgang zeigt den Fehlerstatus an.	
bErrornumber	DWORD	Enthält den befehlsspezifischen Fehlercode des zuletz ausgeführten Befehls. Wird durch das Ausführen eines Befehls an den Eingängen auf 0 zurückgesetzt.	

Befehlsspezifischer Fehlercode (dezimal)	Beschreibung	
1	Kommunikations-Timeout	
2	ASI-Slaveadresse nicht vorhanden	
3 - 10	Reserviert	
11	ASI-Slave ist nicht aktiviert (Slave ist nicht in LAS).	
Bei der Kommunikation ist ein Fehler aufgetreten.		
13	Datenaustauschbit (CN.4) nicht gesetzt	

Voraussetzungen

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	PC oder CX (x86)	ASI Masterklemme	Tc2_loFunctions (IO)



3.2.6 FB_ASI_Processdata_digital

```
FB_ASI_Processdata_digital

iSlaveaddress BYTE

iSlavevalue BYTE

bParametermode BOOL

bCycleMode BOOL

bCommMode BOOL

bCommMode BOOL

bRegComm BOOL

bmaskAccess BOOL

bStart BOOL

stParameterBuffer

FB_ASI_Processdata_digital

BOOL bBusy

BOOL bErr

DWORD iErrornumber

BYTE iReadValue

WORD iParametergroup

bRegComm BOOL

bstart BOOL

stParameterBuffer ST_ParameterBuffer
```

Eingänge

```
VAR_INPUT

iSlaveaddress : BYTE;
iSlavevalue : WORD;
bParametermode : BOOL;
bCycleMode : BOOL;
bCommMode : BOOL;
bRegComm : BOOL;
bmaskAccess : BOOL;
bStart : BOOL;
```

Name	Тур	Beschreibung	
iSlaveaddress	BYTE	Slaveadresse	
iSlavevalue	WORD	Prozesswert	
bParametermode	BOOL	0: Read, 1: Write	
bCycleMode	BOOL	0: Acyclic, 1:Cyclic (permanent Read/Write) Ist dieses Bit gesetzt, wird der Ausgang bBusy erst zurückgenommen, wenn der Eingang bStart auf FALSE gezogen wird. Wird der Eingang bStart zu früh auf FALSE gezogen, steht noch kein aktueller Wert am Ausgang an.	
bCommMode	BOOL	0: Parameterzugriff, 1: ADS (z.Z. immer 0)	
bRegComm	BOOL	Registerkommunikation: 0: Parameterzugriff, 1: Registerkommunikation (z.Z. immer 0)	
bmaskAccess	BOOL	0:normaler Zugriff, 1:maskierter Zugriff	
bStart	BOOL	Mit einer positiven Flanke an diesem boolschen Eingang wird der entsprechende Auftrag ausgeführt.	

▼/ ■ Ein-/Ausgänge

```
VAR_IN_OUT
stParameterBuffer : ST_ParameterBuffer;
END_VAR
```

Name	Тур	Beschreibung
stParameterBuffer	ST ParameterBuffer	Datenpuffer für die Hintergrundkommunikation.
	[<u>\bar{144}</u>]	

Ausgänge

```
VAR_OUTPUT

bBusy : BOOL;
bErr : BOOL;
iErrornumber : DWORD; (* Error code of ASI-Master *)
iReadValue : WORD;
iParametergroup : WORD;
END_VAR
```



Name	Тур	Beschreibung
bBusy	BOOL	Dieser Ausgang bleibt solange auf TRUE, bis der Baustein eine Befehlsanforderung ausführt. Während Busy = TRUE wird an den Eingängen kein neuer Befehl angenommen. Beachten Sie, dass nicht die Ausführung des Dienstes, sondern nur dessen Annahme zeitlich überwacht wird.
bErr	BOOL	Dieser Ausgang zeigt den Fehlerstatus an.
bErrornumber	DWORD	Enthält den befehlsspezifischen Fehlercode des zuletzt ausgeführten Befehls. Wird durch das Ausführen eines Befehls an den Eingängen auf 0 zurückgesetzt.

Befehlsspezifischer Fehlercode (dezimal)	Beschreibung
1	Kommunikations-Timeout
2	ASI-Slaveadresse nicht vorhanden
3 - 10	Reserviert
11	ASI-Slave ist nicht aktiviert (Slave ist nicht in LAS).
12	Bei der Kommunikation ist ein Fehler aufgetreten.
13	Datenaustauschbit (CN.4) nicht gesetzt

Name	Тур	Beschreibung
iReadvalue	WORD	Prozesswert
iParametergroup	WORD	Parametergruppe

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	PC oder CX (x86)	ASI Masterklemme	Tc2 IoFunctions (IO)

3.2.7 FB_ASI_ParameterControl

	FB_ASI_ParameterControl		
_	stParameter_IN ST_Parameter_IN		
_	stParameter_OUT ST_Parameter_OUT		
_	stParameterBuffer ST_ParameterBuffer		

Der FB_ASI_ParameterControl realisiert die Hintergrundkommunikation zwischen der ASI-Masterklemme und den einzelnen Bausteinen der Lib.



Aufruf des Bausteins



Dieser Baustein muss immer zyklisch aufgerufen werden.

🕏 🕒 Ein-/Ausgänge

```
VAR_IN_OUT
stParameterBuffer : ST_ParameterBuffer;
stParameter_IN : ST_Parameter_IN;
stParameter_OUT : ST_Parameter_OUT;
END_VAR
```

Name	Тур	Beschreibung
stParameterBuffer	ST ParameterBuffer	Datenpuffer für die Hintergrundkommunikation.
	[> 144]	
stParameter_IN	ST Parameter IN [▶ 142]	Eingangsdaten von der ASI-Klemme.



Name	Тур	Beschreibung
stParameter_OUT	ST Parameter OUT [▶ 143]	Eingangsdaten von der ASI-Klemme.

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	PC oder CX (x86)	ASI Masterklemme	Tc2 IoFunctions (IO)

3.2.8 FB_ReadInput_analog

```
FB_ReadInput_analog

iSlaveaddress BYTE

iChannel BYTE

bCycleMode BOOL

bStart BOOL

stParameterBuffer ST_ParameterBuffer

FB_ReadInput_analog

BOOL bBusy

BOOL bErr

BYTE iErrornumber

BOOL bValid

BOOL bOverflow

WORD iReadValue
```

Eingänge

```
VAR_INPUT
iSlaveaddress: BYTE;
iChannel: BYTE;
bCycleMode: BOOL;(*0: Acyclic, 1:Cyclic (permanent Read/Write) *)
bStart: BOOL;
END_VAR
```

Name	Тур	Beschreibung
iSlaveaddress	BYTE	Slaveadresse
iChannel	BYTE	Kanal des Slaves
bCycleMode	BOOL	0: Acyclic, 1:Cyclic (permanent Read/Write) Ist dieses Bit gesetzt, wird der Ausgang bBusy erst zurückgenommen, wenn der Eingang bStart auf FALSE gezogen wird. Wird der Eingang bStart zu früh auf FALSE gezogen, steht noch kein aktueller Wert am Ausgang an.
bStart	BOOL	Mit einer positiven Flanke an diesem boolschen Eingang wird der entsprechende Auftrag ausgeführt.

▼/ ■ Ein-/Ausgänge

```
VAR_IN_OUT
stParameterBuffer : ST_ParameterBuffer;
END_VAR
```

Name	Тур	Beschreibung
stParameterBuffer	ST_ParameterBuffer	Datenpuffer für die Hintergrundkommunikation.
	[<u>144</u>]	

Ausgänge

```
VAR_OUTPUT

bBusy : BOOL;
bErr : BOOL;
iErrornumber : DWORD; (* Error code of ASI-Master *)
bValid : BOOL;
bOverflow : BOOL;
iReadValue : WORD;
END_VAR
```



Name	Тур	Beschreibung
bBusy	BOOL	Dieser Ausgang bleibt solange auf TRUE, bis der Baustein eine Befehlsanforderung ausführt. Während Busy = TRUE wird an den Eingängen kein neuer Befehl angenommen. Beachten Sie, dass nicht die Ausführung des Dienstes, sondern nur dessen Annahme zeitlich überwacht wird.
bErr	BOOL	Dieser Ausgang zeigt den Fehlerstatus an.
bErrornumber	DWORD	Enthält den befehlsspezifischen Fehlercode des zuletzt ausgeführten Befehls. Wird durch das Ausführen eines Befehls an den Eingängen auf 0 zurückgesetzt.

Befehlsspezifischer Fehlercode (dezimal)	Beschreibung
1	Kommunikations-Timeout
2	ASI-Slaveadresse nicht vorhanden
3 - 10	Reserviert
11	ASI-Slave ist nicht aktiviert (Slave ist nicht in LAS).
12	Bei der Kommunikation ist ein Fehler aufgetreten.
13	Datenaustauschbit (CN.4) nicht gesetzt

Name	Тур	Beschreibung
bValid	BOOL	Gültigkeit der gelesenen Werte.
bOverflow	BOOL	Slave hat einen Wert außerhalb seines Wertebereiches.
iReadvalue	WORD	Prozesswert

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
	II.	9 ,

3.2.9 FB_WriteOutput_analog

Eingänge

```
VAR_INPUT

iSlaveaddress : BYTE;
iChannel : BYTE;
iSlavevalue : WORD;
bCycleMode : BOOL; (*0: Acyclic , 1:Cyclic (permanent Read/Write) *)
bStart : BOOL;
END_VAR
```

Name	Тур	Beschreibung
iSlaveaddress	BYTE	Slaveadresse
iChannel	BYTE	Kanal des Slaves
iSlavevalue	WORD	Zu schreibende Daten



Name	Тур	Beschreibung
bCycleMode	BOOL	0: Acyclic, 1:Cyclic (permanent Read/Write) Ist dieses Bit gesetzt, wird der Ausgang bBusy erst zurückgenommen, wenn der Eingang bStart auf FALSE gezogen wird. Wird der Eingang bStart zu früh auf FALSE gezogen, steht noch kein aktueller Wert am Ausgang an.
bStart	BOOL	Mit einer positiven Flanke an diesem boolschen Eingang wird der entsprechende Auftrag ausgeführt.

₹/ Ein-/Ausgänge

VAR_IN_OUT stParameterBuffer : ST_ParameterBuffer; END_VAR

Name	Тур	Beschreibung
stParameterBuffer	ST_ParameterBuffer	Datenpuffer für die Hintergrundkommunikation.
	[<u>\bar{144}</u>]	

Ausgänge

VAR_OUTPUT

bBusy : BOOL;
bErr : BOOL;
bErrornumber : DWORD;

END_VAR

Name	Тур	Beschreibung
bBusy	BOOL	Dieser Ausgang bleibt solange auf TRUE, bis der Baustein eine Befehlsanforderung ausführt. Während Busy = TRUE wird an den Eingängen kein neuer Befehl angenommen. Beachten Sie, dass nicht die Ausführung des Dienstes, sondern nur dessen Annahme zeitlich überwacht wird.
bErr	BOOL	Dieser Ausgang zeigt den Fehlerstatus an.
bErrornumber	DWORD	Enthält den befehlsspezifischen Fehlercode des zuletzt ausgeführten Befehls. Wird durch das Ausführen eines Befehls an den Eingängen auf 0 zurückgesetzt.

Befehlsspezifischer Fehlercode (dezimal)	Beschreibung
1	Kommunikations-Timeout
2	ASI-Slaveadresse nicht vorhanden
3 - 10	Reserviert
11	ASI-Slave ist nicht aktiviert (Slave ist nicht in LAS).
12	Bei der Kommunikation ist ein Fehler aufgetreten.
13	Datenaustauschbit (CN.4) nicht gesetzt

Voraussetzungen

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	PC oder CX (x86)	ASI Masterklemme	Tc2_loFunctions (IO)



3.3 AX200x Profibus

3.3.1 Übersicht

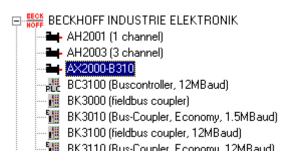
Funktionsbausteine für den Zugriff auf den AX20XX über den Profibus. Voraussetzung für den Betrieb am Profibus ist die Verwendung einer FC310x mit einer Firmwareversion größer 1.20.

Funktionsbausteine

Name	Beschreibung
FB_AX2000_Parameter [▶ 40]	Schreiben/Lesen der Daten zur Parametrierung des Antriebs. Beachten Sie, dass während des Schreibens eines Parameters zum Wechsel der Betriebsart der Eingang "STOP" des Bausteins AX2000AXACT auf TRUE gehalten werden muss.
FB AX2000 AXACT [▶ 42]	Starten von Achsaktionen (muss immer zyklisch aufgerufen werden)
FB AX2000 Jogmode [• 43]	Tippbetrieb
FB_AX2000_Reference [▶ 44]	Setzen des Referenzpunktes bzw. starten einer Referenzfahrt
FB AX200X Profibus [▶ 46]	Dieser Baustein fasst die drei vorangegangenen Bausteine zusammen. Er bietet die komplette Schnittstelle zum AX2000 mit Zugriff auf sämtliche Funktionen (ausg. Parameter).

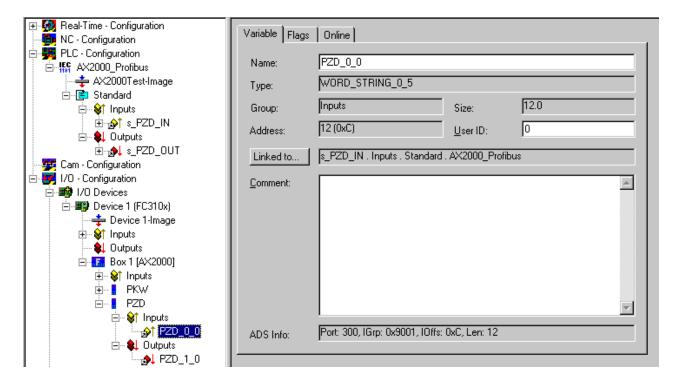
Einbindung in den System Manager

Im TwinCAT System Manager wird in der I/O-Konfiguration unter der entsprechenden ProfibusKarte direkt die Box "AX2000" angefügt.



Im Modul "PZD" (Prozessdaten) der AX2000-Box können nun die I/O-Variablen direkt mit den entsprechenden I/O-Variablen der SPS-Applikation verknüpft werden. Das Modul "PKW" wird nicht verknüpft, da die PKW-Daten per ADS übertragen werden.

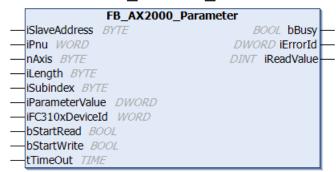




Hinweise zur Benutzung der Funktionsbausteine

- Die E/A-Strukturen stPZD_IN und stPZD_OUT müssen instanziiert und adressiert werden, um sie im System Manager mit der Achse verknüpfen zu können.
- Der Antrieb befindet sich nach dem Einschalten in einem Sicherheits-Betriebsmodus, d.h. vor der ersten Achsaktion muss der Betriebsmodus "Positionierung" oder "Drehzahl digital" eingestellt werden. Dies geschieht durch setzen des Eingangs "blnit" und "bMode_DigitalSpeed" (bei Drehzahlmodus) am Baustein AX200X Profibus.
- Die Fahrtrichtung im Tippbetrieb wird durch das Vorzeichen der "JogModeBasicVelo" festgelegt.
- Jede Referenzfahrt und jedes Setzen des Referenzpunktes **muss** mit einem bStop =TRUE beendet werden.

3.3.2 FB_AX2000_Parameter



Der Funktionsbaustein FB_AX2000_Parameter dient zum Schreiben/Lesen der Daten zur Parametrierung des Antriebs.



Beachten Sie, dass während des Schreibens eines Parameters zum Wechsel der Betriebsart der Eingang "STOP" des Bausteins AX2000AXACT auf TRUE gehalten werden muss.



Eingänge

```
VAR_INPUT

iSlaveAddress : BYTE := 0; (* Station Address of the Slave *)

iPnu : WORD := 16#03A2; (* Parameter-Number *)

nAxis : BYTE := 1; (* Number of Axis *)

iLength : BYTE := 4; (* Length of the parameter (2 or 4) *)

iSubIndex : BYTE;

iParameterValue : DWORD := 2; (* Parameter value *)

iFC310xDeviceId : WORD := 1; (* Device-ID of the FCxxxx *)

bStartRead : BOOL; (* StartFlag to start the PKW-Read *)

bStartWrite : BOOL; (* StartFlag to start the PKW-Write *)

tTimeOut : TIME := DEFAULT_ADS_TIMEOUT;

END_VAR
```

Name	Тур	Beschreibung
iSlaveAddress	BYTE	Stationsadresse
iPnu	WORD	Auswahl des zu schreibenden / zu lesenden Parameters.
		Liste mit den verfügbaren <u>Parameternummern [▶ 159]</u> .
nAxis	BYTE	Achsen-Id
iLength	BYTE	Länge der Parameter (2 oder 4)
iSubIndex	BYTE	
iParameterValue	DWORD	Wert des zu schreibenden / zu lesenden Parameters.
iFC310xDeviceId	WORD	Device-Id
bStartRead	BOOL	Mit einer positiven Flanke an diesem boolschen Eingang wird ein Startbefehl zum Lesen des mit 'Pnu' gewählten Parameters an die Achse gesendet.
bStartWrite	BOOL	Mit einer positiven Flanke an diesem boolschen Eingang wird ein Startbefehl zum Schreiben des mit 'Pnu' gewählten Parameters an die Achse gesendet. Bei Betriebsartenwechsel ist der Schreibbefehl nur bei Stop=TRUE an dem Baustein FB AX2000 AXACT [• 42] wirksam.
tTimeOut	BOOL	Maximale Zeit die bei der Ausführung des Befehls nicht überschritten werden soll.

Ausgänge

VAR_OUTPUT
bBusy :BOOL;
iErrorId :DWORD;
iReadValue :DINT;
END VAR

Name	Тур	Beschreibung
bBusy	BOOL	Dieser Ausgang bleibt solange auf TRUE, bis der Baustein eine Befehlsanforderung ausführt. Während Busy = TRUE wird an den Eingängen kein neuer Befehl angenommen. Beachten Sie, dass nicht die Ausführung des Dienstes, sondern nur dessen Annahme zeitlich überwacht wird.
iErrorld	DWORD	Enthält den befehlsspezifischen Fehlercode des zuletzt ausgeführten Befehls. Wird durch das Ausführen eines Befehls an den Eingängen auf 0 zurückgesetzt.
iReadValue	DINT	Parameterwert als Antwort auf den Befehl 'StartRead'.

Voraussetzungen

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	PC oder CX (x86)	AX2000 Profibus box	Tc2_loFunctions (IO)



3.3.3 FB_AX2000_AXACT

```
FB_AX2000_AXACT
iVelocity
                                bBusy
iPosition
                                bError
imotion_tasknumber
                          bTimeOutErr
imotion_blocktype
                              stPZDIN
bStart
                            stPZDOUT
bStop
bShortStop
bErrorResume
tTimeOut
stPZDIN
stPZDOUT
```

Der Funktionsbaustein FB_AX2000_AXACT dient zum Starten von Achsaktionen (muss immer zyklisch aufgerufen werden).

Eingänge

```
VAR_INPUT

iVelocity : DWORD; (*Velocity*)
iPosition : DINT; (*Position*)
imotion_tasknumber : WORD; (*number of EEPROM-saved motion-task*)
imotion_blocktype : WORD; (*optional Parameters of motion tasks*)
bStart : BOOL; (*START*)
bStop : BOOL; (*STOP*)
bShortStop : BOOL; (*1: break of motion task, 0: continue same motion task*)
bErrorResume : BOOL; (*Error resume*)
tTimeOut : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

Name	Тур	Beschreibung
iVelocity	DWORD	Der Parameter enthält die geforderte Fahrgeschwindigkeit für einen nachfolgenden Fahrauftrag z.B. µm/s.
iPosition	DINT	Zielposition in physikalischen Größen z.B. µm, Grad.
imotion_tasknumber	WORD	Fahrsatznummer. Mit diesem Eingang kann ein vorher im Speicher des Antriebes abgelegter Fahrsatz ausgewählt werden.
imotion_blocktype	WORD	Fahrsatzart (optional) Mit diesem Eingang können Eigenschaften eines Direktfahrauftrages verändert werden.
bStart	BOOL	Mit einer positiven Flanke an diesem boolschen Eingang wird ein Startbefehl an die Achse gesendet.
bStop	BOOL	Mit einer positiven Flanke an diesem boolschen Eingang wird ein Stoppbefehl an die Achse gesendet. Die Achse hält und geht in den Zustand "disabled".
bShortStop	BOOL	Mit einer positiven Flanke an diesem boolschen Eingang wird ein Stoppbefehl an die Achse gesendet. Die Achse hält, bleibt aber im Zustand "enabled".
bErrorResume	BOOL	Mit einer positiven Flanke an diesem boolschen Eingang wird ein "AX200X-Fehler" zurückgesetzt (kein TimeOut-Fehler).
tTimeOut	BOOL	Maximale Zeit die bei der Ausführung des Befehls nicht überschritten werden soll.

▼/ Ein-/Ausgänge

```
VAR_IN_OUT
stPZDIN : ST_PZD_IN;
stPZDOUT : ST_PZD_OUT;
END VAR
```



Name	Тур	Beschreibung
stPZDIN	ST PZD IN [▶ 146]	Datenwörter vom Antrieb zur PLC.
stPZDOUT	ST PZD OUT [▶ 146]	Datenwörter von der PLC zum Antrieb.

Ausgänge

VAR_OUTPUT

bBusy : BOOL; bError : BOOL; (*Errorstatus of Servo*)

bTimeOutErr : BOOL;

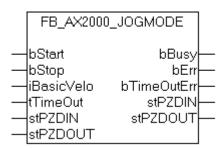
END VAR

Name	Тур	Beschreibung
bBusy		Dieser Ausgang bleibt solange auf TRUE, bis der Baustein eine Befehlsanforderung ausführt. Während Busy = TRUE wird an den Eingängen kein neuer Befehl angenommen. Beachten Sie, dass nicht die Ausführung des Dienstes, sondern nur dessen Annahme zeitlich überwacht wird.
bError	BOOL	Dieser Ausgang zeigt den Fehlerstatus an.
bTimeOutErr	BOOL	TimeOut-Fehler

Voraussetzungen

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	PC oder CX (x86)	AX2000 Profibus box	Tc2_loFunctions (IO)

3.3.4 FB_AX2000_JogMode



Der Funktionsbaustein FB_AX2000_JogMode dient zum Tippbetrieb.

Eingänge

VAR_INPUT bStart : BOOL; bStop : BOOL;

iBasicVelo : INT; (*BasicVelocity*) tTimeOut : TIME := DEFAULT_ADS_TIMEOUT;

END_VAR

Name	Тур	Beschreibung
bStart	BOOL	Starten des Tippbetriebes
bStop	BOOL	Stoppen des Tippbetriebes
iBasicVelo	INT	Basisgeschwindigkeit, die tatsächliche Geschwindigkeit ergibt sich aus der Basisgeschwindigkeit und dem Faktor "v-Tippbetrieb" des Antriebes.
tTimeOut	BOOL	Maximale Zeit die bei der Ausführung des Befehls nicht überschritten werden soll.



▼/ Ein-/Ausgänge

```
VAR_IN_OUT
stPZDIN : ST_PZD_IN;
stPZDOUT : ST_PZD_OUT;
END VAR
```

Name	Тур	Beschreibung
stPZDIN	ST PZD IN [▶ 146]	Datenwörter vom Antrieb zur PLC.
stPZDOUT	ST_PZD_OUT [▶ 146]	Datenwörter von der PLC zum Antrieb.

Ausgänge

```
VAR_OUTPUT

bBusy : BOOL;
bErr : BOOL;
bTimeOutErr : BOOL;
END_VAR
```

Name	Тур	Beschreibung
bBusy	BOOL	Dieser Ausgang bleibt solange auf TRUE, bis der Baustein eine Befehlsanforderung ausführt. Während Busy = TRUE wird an den Eingängen kein neuer Befehl angenommen. Beachten Sie, dass nicht die Ausführung des Dienstes, sondern nur dessen Annahme zeitlich überwacht wird.
bErr	BOOL	Dieser Ausgang zeigt den Fehlerstatus an.
bTimeOutErr	BOOL	TimeOut-Fehler

Voraussetzungen

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	PC oder CX (x86)	AX2000 Profibus box	Tc2_loFunctions (IO)

3.3.5 FB_AX2000_Reference

```
FB_AX2000_REFERENCE

— bSetRefPoint bBusy—
bCalibrStart bErr—
bCalibrStop stPZDIN—
iCalVelo stPZDOUT—
stPZDIN
— stPZDOUT
```

Der Funktionsbaustein FB_AX2000_Reference dient zum Setzen des Referenzpunktes bzw. Starten einer Referenzfahrt

Eingänge

```
VAR_INPUT

bSetRefPoint : BOOL; (* set Reference Point*)

bCalibrStart : BOOL; (* start home running*)

bCalibrStop : BOOL; (* stop home running*)

iCalVelo : WORD; (* basic velocity of Calibration*)

END VAR
```

Name	Тур	Beschreibung
bSetRefPoint	BOOL	Setzen des Referenzpunktes.
bCalibrStart	BOOL	Starten der Referenzfahrt.



Name	Тур	Beschreibung
bCalibrStop	BOOL	Stoppen der Referenzfahrt.
iCalVelo	WORD	Basisgeschwindigkeit der Referenzfahrt. Die Endgeschwindigkeit setzt sich aus der Basisgeschwindigkeit und dem Faktor "v-Tippbetrieb" des Antriebes zusammen.

Ein-/Ausgänge

VAR_IN_OUT stPZDIN : ST_PZD_IN; stPZDOUT : ST_PZD_OUT; END_VAR

Name	Тур	Beschreibung
stPZDIN	ST_PZD_IN [▶ 146]	Datenwörter vom Antrieb zur PLC.
stPZDOUT	ST PZD OUT [▶ 146]	Datenwörter von der PLC zum Antrieb.

Ausgänge

VAR_OUTPUT bBusy : BOOL; bErr : BOOL; END_VAR

Name	Тур	Beschreibung
bBusy		Dieser Ausgang bleibt solange auf TRUE, bis der Baustein eine Befehlsanforderung ausführt. Während Busy = TRUE wird an den Eingängen kein neuer Befehl angenommen. Beachten Sie, dass nicht die Ausführung des Dienstes, sondern nur dessen Annahme zeitlich überwacht wird.
bErr	BOOL	Dieser Ausgang zeigt den Fehlerstatus an.

Voraussetzungen

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	PC oder CX (x86)	AX2000 Profibus box	Tc2 IoFunctions (IO)



3.3.6 FB_AX200X_Profibus

```
FB AX200X Profibus
                                           BOOL bBusy
bMode_DigitalSpeed BOOL
                                           BOOL bError
iDigitalSpeed DINT
                                          DWORD iErrID
iVelocity DWORD
                                     BOOL bTimeOutErr
                                          BOOL bInitOK
iPosition DINT
iRunningMode BYTE
                                       DINT iactPosition
imotion_tasknumber WORD
imotion blocktype WORD
iJogModeBasicValue INT
iCalVelo WORD
bSetRefPoint BOOL
bStart BOOL
bStop BOOL
bShortStop BOOL
iSlaveAddress BYTE
iFC310xDeviceId WORD
bErrorResume BOOL
tTimeOut TIME
stPZD_IN ST_PZD_IN
stPZD_OUT ST_PZD_OUT
```

Der Funktionsbaustein FB_AX200X_Profibus fasst die drei vorangegangenen Bausteine zusammen. Er bietet die komplette Schnittstelle zum AX2000 mit Zugriff auf sämtliche Funktionen (ausg. Parameter).

Eingänge

```
VAR_INPUT

bInit : BOOL; (*Initialization*)

bMode_DigitalSpeed : BOOL; (*OP-Mode digital speed instead of Positioning*)

iDigitalSpeed : DWORD; (*digital speed if OP-Mode = digital speed*)

iVelocity : DWORD; (*Velocity*)

iPosition : DINT; (*Position*)

iRunningMode : BYTE; (*0:digital speed, 1: motiontask, 2: JogMode, 3: Calibration*)

imotion_tasknumber : WORD; (*number of EEPROM-saved motion-task*)

imotion_blocktype : WORD:=16#2000; (*optional Parameters of motion tasks, default:SI-values*)

iJogModeBasicValue : INT; (*BasicVelocity for JogMode*)

iCalVelo : WORD; (* basic velocity of Calibration*)

bSetRefPoint : BOOL; (* set Reference Point*)

bStart : BOOL; (* set Reference Point*)

bStop : BOOL; (*START*)

bStop : BOOL; (* break of motion task*)

iSlaveAddress : BYTE; (* Station Address of the Slave *)

iFC310xDeviceId : WORD; (* Device-ID of the FCxxxx *)

bErrorResume : BOOL; (*Error resume*)

tTimeOut : TIME := DEFAULT_ADS_TIMEOUT;

END_VAR
```

Name	Тур	Beschreibung
blnit	BOOL	Initialisierung des Antriebes. Bei blnit TRUE wird im Antrieb die Betriebsart 2 'Positionierung' eingestellt.
bMode_DigitalSpeed	BOOL	Wird gesetzt, wenn der Antrieb bei der Initialisierung in die Betriebsart 'Drehzahl digital' versetzt werden soll.
iDigitalSpeed	DWORD	Drehzahl in der Betriebsart 'Drehzahl digital'.
iVelocity	DWORD	Der Parameter enthält die geforderte Fahrgeschwindigkeit für einen nachfolgenden Fahrauftrag z.B. µm/s.
iPosition	DINT	Zielposition
iRunningMode	BYTE	0: Drehzahl digital, 1: Motiontask, 2: JogMode, 3: Calibration.
imotion_tasknumber	WORD	Fahrsatznummer. Mit diesem Eingang kann ein vorher im Speicher des Antriebes abgelegter Fahrsatz ausgewählt werden.



Name	Тур	Beschreibung	
imotion_blocktype	WORD	Fahrsatzart (optional) Mit diesem Eingang können Eigenschaften eines Direktfahrauftrages verändert werden.	
iJogModeBasicValu e	INT	Basisgeschwindigkeit für den Tippbetrieb, die tatsächliche Geschwindigkeit ergibt sich aus der Basisgeschwindigkeit und dem Faktor "v-Tippbetrieb" des Antriebes.	
iCalVelo	WORD	Basisgeschwindigkeit der Referenzfahrt. Die Endgeschwindigkeit setzt sich aus der Basisgeschwindigkeit und dem Faktor "v-Tippbetrieb" de Antriebes zusammen.	
bSetRefPoint	BOOL	Setzen des Referenzpunktes	
bStart	BOOL	Starten der Aktion je nach Zustand von iRunningMode.	
bStop	BOOL	Stoppen der Aktion je nach Zustand von iRunningMode.	
bShortStop	BOOL		
iSlaveAddress	BYTE	Stationsadresse	
iFC310xDeviceId	WORD	Device-Id	
bErrorResume	BOOL	Mit einer positiven Flanke an diesem boolschen Eingang wird ein "AX200X-Fehler" zurückgesetzt (kein TimeOut-Fehler).	
tTimeOut	BOOL	Maximale Zeit die bei der Ausführung des Befehls nicht überschritten werden soll.	

▼/ ■ Ein-/Ausgänge

VAR_IN_OUT STPZDIN : ST_PZD_IN; StPZDOUT : ST_PZD_OUT; END_VAR

Name	Тур	Beschreibung
stPZDIN	ST_PZD_IN [▶ 146]	Datenwörter vom Antrieb zur PLC.
stPZDOUT	ST_PZD_OUT [▶ 146]	Datenwörter von der PLC zum Antrieb.

Ausgänge

VAR_OUTPUT OUTPUT

bBusy : BOOL;

bError : BOOL; (*Errorstatus of Servo*)

iErrID : DWORD;

bTimeOutErr : BOOL;

bInitOK : BOOL; (*Initialization OK*)

iactPosition : DINT; (*actual Position SI-value*)

END_VAR

Name	Тур	Beschreibung
bBusy	BOOL	Dieser Ausgang bleibt solange auf TRUE, bis der Baustein eine Befehlsanforderung ausführt. Während Busy = TRUE wird an den Eingängen kein neuer Befehl angenommen. Beachten Sie, dass nicht die Ausführung des Dienstes, sondern nur dessen Annahme zeitlich überwacht wird.
bError	BOOL	Dieser Ausgang zeigt den Fehlerstatus an.
iErrID	DWORD	Enthält den befehlsspezifischen Fehlercode des zuletzt ausgeführten Befehls. Wird durch das Ausführen eines Befehls an den Eingängen auf 0 zurückgesetzt.
bTimeOutErr	BOOL	TimeOut-Fehler
blnitOK	BOOL	Initialisierungszustand des Antriebes, bInit:= TRUE: Antrieb ist Initialisiert und in der Betriebsart 2 'Positionierung'.
iactPosition	DINT	Aktuelle Positionsanzeige im RunningMode 1: Motiontask.



Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	PC oder CX (x86)	AX2000 Profibus box	Tc2_loFunctions (IO)

3.4 Beckhoff Lightbus

3.4.1 IOF_LB_BreakLocationTest

```
IOF_LB_BreakLocationTest

NETID T_AmsNetId BOOL BUSY

DEVICEID UDINT BOOL ERR

START BOOL UDINT ERRID

TMOUT TIME BOOL BREAK

WORD BOXNO
```

Der Funktionsbaustein IOF_LB_BreakLocationTest führt ein Bruchstellentest in einem Beckhoff Lightbus-Lichtwellenleiterring durch und kann eventuelle Bruchstellen lokalisieren. Wurde während des Tests keine Bruchstelle im Ring erkannt, dann liefert die Ausgangsvariable **BOXNO** die aktuelle Anzahl der Lightbus-Module im Ring. Bei einer erkannten Bruchstelle vor dem NN-ten Modul vor dem Empfängereingang wird das Flag **BREAK** gesetzt und die Modulnummer über die Ausgangsvariable **BOXNO** ausgegeben. Liefert die **BOXNO**-Variable einen **0xFF**-Wert, dann liegt die Bruchstelle direkt vor dem Empfängereingang und kann nicht lokalisiert werden.

Eingänge

```
VAR_INPUT

NETID : T_AmsNetId;

DEVICEID : UDINT;

START : BOOL;

TMOUT : TIME := DEFAULT_ADS_TIMEOUT;

END_VAR
```

Name	Тур	Beschreibung
NETID	T_AmsNetId	Hier kann die Netzwerkadresse des TwinCAT-Rechners angegeben werden, auf dem das ADS-Kommando ausgeführt werden soll. Für den lokalen Rechner kann auch ein Leerstring angegeben werden.
DEVICEID	UDINT	Über die Deviceld (Geräte-ID) wird das IO-Gerät spezifiziert, auf dem die Funktion ausgeführt werden soll. Die Geräte-IDs werden während der Hardware-Konfiguration von TwinCAT-System festgelegt.
START	BOOL	Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.
TMOUT	TIME	Gibt die Timeout-Zeit an, die bei der Ausführung des Kommandos nicht überschritten werden darf.

Ausgänge

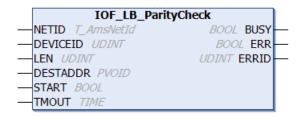
```
VAR_OUTPUT
BUSY: BOOL;
ERR: BOOL;
ERRID: UDINT;
BREAK: BOOL;
BOXNO: WORD;
END_VAR
```



Name	Тур	Beschreibung
BUSY	BOOL	Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
ERR	BOOL	Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der BUSY-Ausgang zurückgesetzt wurde.
ERRID	UDINT	Liefert bei einem gesetzten ERR-Ausgang die <u>ADS-</u> <u>Fehlernummer [▶ 163]</u> .
BREAK	BOOL	Dieses Flag wird gesetzt, wenn in dem Lichwellenleiter- Ring eine Bruchstelle erkannt wurde.
BOXNO	WORD	Die Modulnummer vor dem Empfängereingang, vor dem die Bruchstelle erkannt wurde.

Entwicklungsumgebung	•		Einzubindende SPS-Bibliotheken (Kategorie-gruppe)
TwinCAT v3.1.0	Funktionalität wird zur Zeit von TwinCAT 3	Beckhoff Lightbus Master: C1220 ISA; FC200x PCI	Tc2_loFunctions (IO)

3.4.2 IOF_LB_ParityCheck



Der Funktionsbaustein IOF_LB_ParityCheck liest die Parityfehler-Zähler der Beckhoff Lightbus-Module (z.B. BK2000). Im Gegensatz zu dem IOF LB ParityCheckWithReset [▶ 51] Funktionsbaustein werden die Zählerstände nicht zurückgesetzt. Für jedes Modul hält der Master einen 8-Bit Fehlerzähler bereit. Die Zähler arbeiten ohne Überlauf. Es können maximal 256-Byte Daten und somit 256 Zähler gelesen werden. Die Anzahl der zu lesenden Fehlerzähler wird durch die Eingangsvariablen: LEN und DESTADDR festgelegt. Existieren z.B. nur 5 Module im Ring, dann kann für den DESTADDR-Parameter eine Adresse auf einen 5 Byte großen Datenpuffer übergeben werden und dem LEN-Parameter der Wert 5.

Eingänge

```
VAR_INPUT

NETID : T_AmsNetId;

DEVICEID : UDINT;

LEN : UDINT;

DESTADDR : PVOID;

START : BOOL;

TMOUT : TIME := DEFAULT_ADS_TIMEOUT;

END_VAR
```

Name	Тур	Beschreibung
NETID	T_AmsNetId	Hier kann die Netzwerkadresse des TwinCAT-Rechners angegeben werden, auf dem das ADS-Kommando ausgeführt werden soll. Für den lokalen Rechner kann auch ein Leerstring angegeben werden.



Name	Тур	Beschreibung
DEVICEID	UDINT	Über die Deviceld (Geräte-ID) wird das IO-Gerät spezifiziert, auf dem die Funktion ausgeführt werden soll. Die Geräte-IDs werden während der Hardware-Konfiguration von TwinCAT-System festgelegt.
LEN	UINT	Länge in Bytes der zu lesenden Daten.
DESTADDR	PVOID	Die Adresse des Datenpuffers, in den die Paritydaten geschrieben werden sollen.
START	BOOL	Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.
TMOUT	TIME	Gibt die Timeout-Zeit an, die bei der Ausführung des Kommandos nicht überschritten werden darf.

Ausgänge

VAR OUTPUT

BUSY : BOOL; ERR : BOOL; ERRID : UDINT;

END VAR

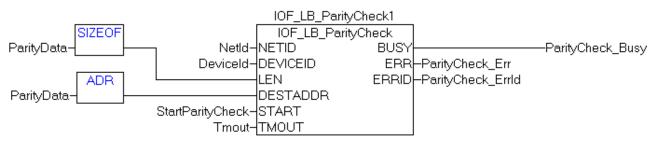
Name	Тур	Beschreibung
BUSY	BOOL	Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
ERR	BOOL	Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der BUSY-Ausgang zurückgesetzt wurde.
ERRID	UDINT	Liefert bei einem gesetzten ERR-Ausgang die ADS-Fehlernummer [▶ 163].

Beispiel:

```
PROGRAM MAIN

VAR

IOF_LB_ParityCheck1 : IOF_LB_ParityCheck;
ParityData : ARRAY[1..256] OF BYTE;
StartParityCheck : BOOL;
ParityCheck_Busy : BOOL;
ParityCheck_Err : BOOL;
ParityCheck_ErrId : UDINT;
END_VAR
```



Voraussetzungen

Entwicklungsumgebung	· -		Einzubindende SPS-Bibliotheken (Kategorie-gruppe)
TwinCAT v3.1.0	Funktionalität wird zur Zeit von TwinCAT 3	Lightbus Master:	Tc2_loFunctions (IO)



3.4.3 IOF_LB_ParityCheckWithReset

```
IOF_LB_ParityCheckWithReset

NETID T_AmsNetId BOOL BUSY

DEVICEID UDINT BOOL ERR

LEN UDINT UDINT ERRID

DESTADDR PVOID

START BOOL

TMOUT TIME
```

Der Funktionsbaustein IOF_LB_ParityCheckWithReset liest die Parityfehler-Zähler der Beckhoff Lightbus-Module (z.B. BK2000). Anschließend werden die Zähler zurückgesetzt. Für jedes Modul hält der Master einen 8-Bit Fehlerzähler bereit. Die Zähler arbeiten ohne Überlauf. Es können maximal **256**-Byte Daten und somit **256** Zähler gelesen werden. Die Anzahl der zu lesenden Fehlerzähler wird durch die Eingangsvariablen: **LEN** und **DESTADDR** festgelegt. Existieren z.B. nur 5 Module im Ring, dann kann für den **DESTADDR**-Parameter eine Adresse auf einen 5 Byte großen Datenpuffer übergeben werden und dem **LEN**-Parameter der Wert 5.

Eingänge

```
VAR_INPUT

NETID : T_AmsNetId;
DEVICEID : UDINT;
LEN : UDINT;
DESTADDR : PVOID;
START : BOOL;
TMOUT : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

Name	Тур	Beschreibung
NETID	T_AmsNetId	Hier kann die Netzwerkadresse des TwinCAT-Rechners angegeben werden, auf dem das ADS-Kommando ausgeführt werden soll. Für den lokalen Rechner kann auch ein Leerstring angegeben werden.
DEVICEID	UDINT	Über die Deviceld (Geräte-ID) wird das IO-Gerät spezifiziert, auf dem die Funktion ausgeführt werden soll. Die Geräte-IDs werden während der Hardware-Konfiguration von TwinCAT-System festgelegt.
LEN	UINT	Länge in Bytes der zu lesenden Daten.
DESTADDR	PVOID	Die Adresse des Datenpuffers, in den die Paritydaten geschrieben werden sollen.
START	BOOL	Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.
TMOUT	TIME	Gibt die Timeout-Zeit an, die bei der Ausführung des Kommandos nicht überschritten werden darf.

Ausgänge

```
VAR_OUTPUT

BUSY : BOOL;

ERR : BOOL;

ERRID : UDINT;

END_VAR
```

Name	Тур	Beschreibung
BUSY	BOOL	Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
ERR	BOOL	Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der BUSY-Ausgang zurückgesetzt wurde.



Name	Тур	Beschreibung	
ERRID	UDINT	Liefert bei einem gesetzten ERR-Ausgang die ADS-	
		<u>Fehlernummer [▶ 163]</u> .	

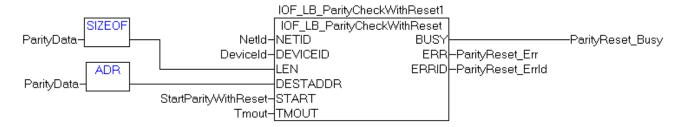
Beispiel:

```
PROGRAM MAIN

VAR

IOF_LB_ParityCheckWithReset1 : IOF_LB_ParityCheckWithReset;
ParityData : ARRAY[1..256] OF BYTE;
StartParityWithReset : BOOL;
ParityReset_Busy : BOOL;
ParityReset_Err : BOOL;
ParityReset_Err : BOOL;
ParityReset_ErrId : UDINT;

END VAR
```



Voraussetzungen

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bibliotheken (Kategorie-gruppe)
TwinCAT v3.1.0	Funktionalität wird zur Zeit von TwinCAT 3	Beckhoff Lightbus Master: C1220 ISA; FC200x PCI	Tc2_loFunctions (IO)

3.5 Beckhoff USV (konfiguriert mit Windows USV service)

3.5.1 FB_GetUPSStatus



Voraussetzungen:

- Die Beckhoff USV Softwarekomponenten wurden installiert:
 - Windows 7, Windows Embedded Standard 7 und höher: Konfigurationsdialog unter "Start->Programme->Beckhoff->UPS Software Components".
 - NT4, Win2K, WinXP, WinXP embedded: Zusätzlicher Reiter unter "Systemsteuerung->Energieoptionen->Beckhoff UPS Configuration" oder "Systemsteuerung->Energieoptionen->USV".
 - Beckhoff CE Geräte mit 24V USV-Unterstützung werden mit einem speziellen Beckhoff Battery Driver für Windows CE ausgeliefert. Der Treiber ist bei diesen Geräten in dem Standard CE Image enthalten.
- Die USV wurde aktiviert und konfiguriert. Weitere Informationen zur USV-Konfiguration finden Sie in der entsprechenden weiterführenden USV-Software und Gerätedokumentation.
 - Windows 7, Windows Embedded Standard 7 und höher: Konfigurationsdialog unter "Start->Programme->Beckhoff->UPS Software Components".



- NT4, Win2K, WinXP, WinXP embedded: Konfigurationsdialog unter "Systemsteuerung->Energieoptionen->Beckhoff UPS Configuration".
- Windows CE: Die USV-Funktion ist standardmäßig deaktiviert und muss über ein RegFile aktiviert werden. Neuere Images beinhalten ein Konfigurationsdialog unter "Start->Systemsteuerung->BECKHOFF UPS Configuration".

Der Funktionsbaustein FB_GetUPSStatus liest aus der SPS den Status der USV-Hardware. Der Baustein wird Levelgetriggert, d.h. nur bei dem gesetzten *bEnable* -Eingang werden die Statusinformationen der USV zyklisch gelesen. Um dabei die Systemauslastung niedrig zu halten, werden die Statusinformationen alle ~4,5s neu gelesen. Bei einem gesetzten *bValid*-Ausgang sind die zuletzt gelesenen Daten gültig. D.h. der letzte Lesezyklus wurde fehlerfrei durchgeführt. Beim Auftreten eines Fehlers wird der Lesezyklus wiederholt und der Fehler automatisch zurückgesetzt, sobald die Fehlerursache behoben wurde (z.B. keine Kommunikation zur USV).

Eingänge

```
VAR_INPUT
    sNetId : T_AmsNetId;
    nPort : T_AmsPort; (* 0 = Windows UPS service / Windows Battery Driver *)
    bEnable : BOOL;
END VAR
```

Name	Тур	Beschreibung
sNetId	T_AmsNetID	Hier kann ein String mit der Netzwerkadresse des TwinCAT-Rechners angegeben werden, dessen USV- Status gelesen werden soll. Für den lokalen Rechner kann auch ein Leerstring angegeben werden.
nPort	T_AmsPort	Die ADS-Portnummer. Setzen Sie diesen Wert auf Null. Andere Portnummern sind für zukünftige Anwendungen reserviert.
bEnable	BOOL	Bei einem gesetzten Eingang wird der USV-Status zyklisch gelesen.

Ausgänge

```
VAR_OUTPUT
bValid :BOOL;
bError :BOOL;
nErrId :UDINT;
stStatus :ST_UPSStatus;
END_VAR
```

Name	Тур	Beschreibung
bValid	BOOL	Wenn dieser Ausgang gesetzt ist sind die Daten in der ST_UPSStatus-Struktur gültig (beim letzten Lesezyklus ist kein Fehler aufgetreten).
bError	BOOL	Sollte ein Fehler bei der Ausführung der Funktion erfolgen, dann wird dieser Ausgang gesetzt.
nErrld	UDINT	Liefert bei einem gesetzten <i>bError</i> -Ausgang die <u>ADS-Fehlernummer [* 163]</u> oder einen Befehlsspezifischen Fehlercode zurück (Tabelle).

Fehlercodes	Fehlerbeschreibung
0x0000	Kein Fehler
0x8001	USV-Konfigurationsfehler. Möglicherweise ist die USV nicht richtig oder gar keine USV konfiguriert.
0x8002	Kommunikationsfehler. Die Kommunikation zu der USV wurde unterbrochen.
0x8003	Fehler beim Lesen der Statusdaten.

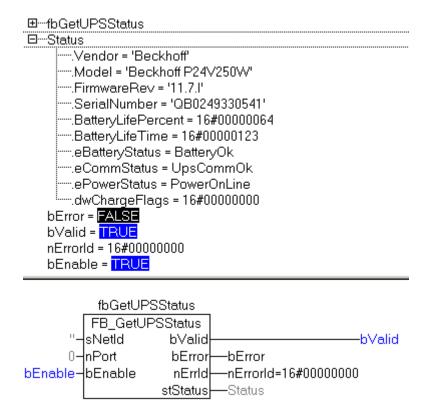


Name	Тур	Beschreibung
stStatus	ST_UPSStatus [▶ 150]	Struktur mit den Statusinformationen der USV.

Nicht alle USV-Geräte können alle Statusinformationen liefern. Einige Geräte können z. B. keine *BatteryLifeTime* oder keinen *BatteryReplace*-Status liefern.

Beispiel:

Online-Daten mit Statusinformationen einer USV:



Voraussetzungen

Entwicklungsum- gebung	Zielplattform	U	SV Hardware	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	•	Beckhoff CU81x0-0xx0	Tc2_loFunctions
		•	Beckhoff BAPI v1;	(IO)
		•	Beckhoff P24Vxxxx;	
		•	Beckhoff CP903x-Karte (PCI/ISA);	
		•	Beckhoff CX2100-09x4 Modelle (z.B. CX2100-0904 oder CX2100-0914 + "Smart Battery" CX2900-0192);	
		•	Die mit Beckhoff Industrie-PC ausgelieferten APC-Geräte die das Smartprotokoll unterstützen und mit dem Windows USV- Dienst konfiguriert werden können;	



3.6 Busklemmen-Konfiguration

3.6.1 FB_KL1501Config

```
FB_KL1501Config
bConfigurate BOOL
                                                    BOOL bBusy
bReadConfig BOOL
                                                    BOOL bError
iSetCounterType INT
                                                  UDINT iErrorId
bSetBackwardCounting BOOL
                                                   USINT iState
tTimeout TIME
                                                   UINT iDataIn0
stInData ST_KL1501InData
                                                  UINT iDataIn1
stOutData ST_KL1501OutData
                                                  UDINT iDataIn
                                            WORD iTerminalType
                                              WORD iSpecialType
                                          WORD iFirmwareVersion
                                             STRING sDescription
                                            STRING sCounterType
                                        BOOL bBackwardCounting
```

Der Funktionsbaustein FB_KL1501Config dient zur Parametrierung eine KL1501: 1-Kanal Zählerklemme.



Dieser Baustein berücksichtigt nicht das alternative Ausgabeformat, da sich bei Umstellung auf dieses Format das Prozessabbild verschiebt.

Eingänge

VAR_INPUT
bConfigurate : BOOL;
bReadConfig : BOOL;
iSetCounterType : INT;
bSetBackwardCounting : BOOL;
tTimeout : TIME;
END VAR

Name	Тур	Beschreibung
bConfigurate	BOOL	Eine steigende Flanke startet die Konfigurationssequenz. Zunächst werden die allgemeinen Klemmendaten "Klemmbezeichnung", "Sondervariante" und "Firmwarestand" ausgelesen. Danach werden die angegebenen Einstellungen in die entsprechenden Register geschrieben und abschließend zur Sicherheit und Information noch einmal ausgelesen. Die gelesenen Informationen werden an den Bausteinausgängen angezeigt. Während des Ablaufs dieser Sequenz steht der Ausgang bBusy auf TRUE und es wird kein weiterer Befehl, wie etwa bReadConfig, entgegengenommen.
bReadConfig	BOOL	Eine steigende Flanke startet eine Lesesequenz. Es werden die allgemeinen Klemmendaten "Klemmbezeichnung", "Sondervariante" und "Firmwarestand" ausgelesen und im Anschluss daran die eingestellten Konfigurationsparameter. Die gelesenen Informationen werden an den Bausteinausgängen angezeigt. Während der Lesesequenz steht der Ausgang bBusy auf TRUE und es wird kein weiterer Befehl, wie etwa bConfigurate, entgegengenommen.
iSetCounterType	INT	Eingabe des Zählertyps. Die Einstellung erfolgt nach unten aufgeführter Tabelle.
bSetBackwardCount ing	BOOL	Ein TRUE an diesem Eingang kehrt die Zählrichtung um.



Name	Тур	Beschreibung
tTimeout		Innerhalb der hier eingetragenen Zeit muss die Konfiguration der Klemme bzw. das Auslesen der Konfiguration abgeschlossen sein. Anderenfalls wird ein Fehler mit entsprechender Fehlernummer an den Ausgängen bError und iErrorld ausgegeben.

iSetCounterType	Zählertyp
0	32-Bit-Vorwärts/Rückwärts-Zähler
1	2 x 16-Bit Vorwärts-Zähler
2	32-bit Gated-Counter, Gate-Eingang Low sperrt den Zähler
3	32-bit Gated-Counter, Gate-Eingang High sperrt den Zähler

▼/ ■ Ein-/Ausgänge

VAR_IN_OUT

stInData : ST_KL1501InData; stOutData : ST_KL1501OutData;

END VAR

Name	Тур	Beschreibung
stInData	ST_KL1501InData [▶ 153]	Verweis auf die Struktur des Eingangsprozessabbildes.
stOutData	ST_KL1501OutData [▶ 154]	Verweis auf die Struktur des Ausgangsprozessabbildes.

Ausgänge

VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
iErrorId : UDINT;
iState : USINT;
iDataIn0 : UINT;
iDataIn1 : UINT;
iTerminalType : WORD;
iSpecialType : WORD;
iFirmwareVersion : WORD;
sDescription : STRING;
sCounterType : STRING;
bBackwardCounting : BOOL;
END_VAR

Name	Тур	Beschreibung
bBusy	BOOL	Solange eine Lese- oder Konfigurationssequenz abgearbeitet wird, steht dieser Ausgang auf TRUE.
bError	BOOL	Dieser Ausgang wird auf TRUE geschaltet, wenn bei der Ausführung eines Befehls (Konfigurieren oder Lesen) ein Fehler aufgetreten ist. Der befehlsspezifische Fehlercode ist in <i>iErrorld</i> enthalten.
iErrorld	UDINT	Enthält den befehlsspezifischen Fehlercode des zuletzt ausgeführten Befehls. Wird durch das erneute Aktivieren des Bausteins über die Eingänge bConfigurate oder bReadConfig wieder auf 0 zurückgesetzt. Siehe Fehlercodes [• 162].
iState	USINT	Entspricht der Statusvariablen der Prozessdaten stInData.iState, siehe VAR_IN_OUT. Während der Befehlsausführung (bBusy = TRUE) ist der Ausgang auf 0 gesetzt. Damit eignet sich dieser Ausgang zur Statusbeurteilung im Normalbetrieb der Klemme. Störende Zustände während des Konfigurierens und Lesens durch die Registerkommunikation werden ausgeblendet.



Name	Тур	Beschreibung
iDataIn0	UINT	Entspricht der Datenvariablen der Prozessdaten stInData.arrDataIn[0], siehe VAR_IN_OUT. Während der Befehlsausführung (bBusy = TRUE) behält dieser Ausgang den Wert, den er vor dem Befehlsaufruf innehatte. Damit eignet sich der Ausgang zur direkten Prozessdatenverarbeitung im Normalbetrieb der Klemme. Störende Zustände während des Konfigurierens und Lesens durch die Registerkommunikation werden ausgeblendet.
iDataIn1	UINT	Entspricht der Datenvariablen der Prozessdaten stlnData.arrDataln[1], siehe VAR_IN_OUT. Während der Befehlsausführung (bBusy = TRUE) behält dieser Ausgang den Wert, den er vor dem Befehlsaufruf innehatte. Damit eignet sich der Ausgang zur direkten Prozessdatenverarbeitung im Normalbetrieb der Klemme. Störende Zustände während des Konfigurierens und Lesens durch die Registerkommunikation werden ausgeblendet.
iDataIn	UDINT	Entspricht der Datenvariablen der Prozessdaten stInData.iDataIn, siehe VAR_IN_OUT. Während der Befehlsausführung (bBusy = TRUE) behält dieser Ausgang den Wert, den er vor dem Befehlsaufruf innehatte. Damit eignet sich der Ausgang zur direkten Prozessdatenverarbeitung im Normalbetrieb der Klemme. Störende Zustände während des Konfigurierens und Lesens durch die Registerkommunikation werden ausgeblendet.
iTerminalType	WORD	Inhalt des Registers 8 (Klemmenbezeichnung). Bei Anwendung mit der richtigen Klemme sollte der Inhalt 0x05DD (1501dez) sein.
iSpecialType	WORD	Inhalt des Registers 29 (Sondervariante)
iFirmwareVersion	WORD	Inhalt des Registers 9 (Firmware-Stand)
sDescription	STRING	Klemmenbezeichnung, Sondervariante und Version der Firmware als String (z. B. 'Terminal KL1501-0000 / Firmware 1C').
sCounterType	STRING	Eingestellter Zählermodus als Klartext.
bBackwardCounting	BOOL	TRUE: Die Zählrichtung wurde umgekehrt.

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bi- bliotheken
TwinCAT v3.1.4018.26	PC/CX	Tc2_loFunctions ab v3.3.5.0



3.6.2 FB_KL27x1Config

```
FB_KL27x1Config
bConfigurate BOOL
                                                    BOOL bBusy
bReadConfig BOOL
                                                    BOOL bError
bSetDimRampAbsolute BOOL
                                                  UDINT iErrorId
iSetRampTime INT
                                                    USINT iState
bSetWatchdogDisable BOOL
                                           WORD iTerminalType
iSetWatchdogTimeout UINT
                                             WORD iSpecialType
iSetTimeoutOnValue UINT
                                       WORD iFirmwareVersion
iSetTimeoutOffValue UINT
                                             STRING sDescription
iSetDimmerMode I///
                                      BOOL bDimRampAbsolute
bSetOnAfterShortCircuit BOOL
                                     STRING sRampTime
BOOL bWatchdogDisable
                                             STRING sRampTime
bSetLineFrequency60Hz BOOL
                                       TIME tWatchdogTimeout
tTimeout TIME
stInData ST_KL27x1InData
                                         UINT iTimeoutOnValue
stOutData ST_KL27x1OutData
                                          UINT iTimeoutOffValue
                                           STRING sDimmerMode
                                        STRING sAfterShortCircuit
                                          STRING sLineFrequency
```

Der Funktionsbaustein FB_KL27x1Config dient zur Parametrierung einer <u>KL2751 / KL2761</u>: 1-Kanal Dimmerklemme.

Eingänge

```
VAR_INPUT

bConfigurate : BOOL;
bReadConfig : BOOL;
bSetDimRampAbsolute : BOOL;
iSetRampTime : INT;
bSetWatchdogDisable : BOOL;
iSetWatchdogTimeout : UINT;
iSetTimeoutOnValue : UINT;
iSetTimeoutOffValue : UINT;
iSetDimmerMode : INT;
bSetOnAfterShortCircuit : BOOL;
bSetLineFrequency60Hz : BOOL;
tTimeout VAR

END VAR
```

Name	Тур	Beschreibung
bConfigurate	BOOL	Eine steigende Flanke startet die Konfigurationssequenz. Zunächst werden die allgemeinen Klemmendaten "Klemmbezeichnung", "Sondervariante" und "Firmwarestand" ausgelesen. Danach werden die angegebenen Einstellungen in die entsprechenden Register geschrieben und abschließend zur Sicherheit und Information noch einmal ausgelesen. Die gelesenen Informationen werden an den Bausteinausgängen angezeigt. Während des Ablaufs dieser Sequenz steht der Ausgang bBusy auf TRUE und es wird kein weiterer Befehl, wie etwa bReadConfig, entgegengenommen.
bReadConfig	BOOL	Eine steigende Flanke startet eine Lesesequenz. Es werden die allgemeinen Klemmendaten "Klemmbezeichnung", "Sondervariante" und "Firmwarestand" ausgelesen und im Anschluss daran die eingestellten Konfigurationsparameter. Die gelesenen Informationen werden an den Bausteinausgängen angezeigt. Während der Lesesequenz steht der Ausgang bBusy auf TRUE und es wird kein weiterer Befehl, wie etwa bConfigurate, entgegengenommen.



Name	Тур	Beschreibung
bSetDimRampAbsol ute	BOOL	FALSE: Die eingestellte Rampenzeit <i>iSetRampTime</i> bezieht sich auf den kompletten Datenbereich (0 - 32767). Je kleiner der Sprung, desto kürzer die Rampenzeit. TRUE: Jeder Schaltschritt, egal wie groß, benötigt dieselbe Zeit, die unter iSetRampTime eingetragen ist.
iSetRampTime	INT	Eingabe der Rampenzeit. Die Einstellung erfolgt nach unten aufgeführter Tabelle.
bSetWatchdogDisab le	BOOL	Der interne Watchdog wird deaktiviert.
iSetWatchdogTimeo ut	ÚINT	Einstellung der Watchdog-Zeit als Vielfaches von 10ms.
iSetTimeoutOnValue	UINT	Dieser Eingang legt den Lichtwert fest, der bei einem Feldbusfehler und aktuellen Prozessdaten > 0 ausgegeben wird.
iSetTimeoutOffValue	UINT	Dieser Eingang legt den Lichtwert fest, der bei einem Feldbusfehler und aktuellen Prozessdaten = 0 ausgegeben wird.
iSetDimmerMode	INT	An diesem Eingang ist der Dimmermodus einzustellen. Die Einstellung erfolgt nach unten aufgeführter Tabelle.
bSetOnAfterShortCir cuit	BOOL	FALSE: Nach einem Kurzschluss bleibt das Licht ausgeschaltet. TRUE: Das Licht wird nach einem Kurzschluss wieder eingeschaltet.
bSetLineFrequency6 0Hz	BOOL	FALSE: Netzfrequenz = 50 Hz. TRUE: Netzfrequenz = 60 Hz.
tTimeout	TIME	Innerhalb der hier eingetragenen Zeit muss die Konfiguration der Klemme bzw. das Auslesen der Konfiguration abgeschlossen sein. Anderenfalls wird ein Fehler mit entsprechender Fehlernummer an den Ausgängen bError und iErrorld ausgegeben.

iSetRampTime	Element
0	50 ms
1	100 ms
2	200 ms
3	500 ms
4	1 s
5	2 s
6	5 s
7	10 s
iSetDimmerMode	Element
0	Automatische Erkennung
1	Phasenabschnitt
2	Phasenanschnitt
3	Gleichrichterbetrieb, positiv (positive Halbwelle mit Phasenanschnitt)
4	Gleichrichterbetrieb, negativ (negative Halbwelle mit Phasenanschnitt)

▼/ ■ Ein-/Ausgänge

VAR_IN_OUT stInData : ST_KL1501InData;
 stOutData : ST_KL1501OutData;
D VAR

END_VAR



Name	Тур	Beschreibung
stInData	ST_KL1501InData [▶ 153]	Verweis auf die Struktur des Eingangsprozessabbildes.
stOutData	ST KL1501OutData [▶ 154]	Verweis auf die Struktur des Ausgangsprozessabbildes.

Ausgänge

```
VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
iErrorId : UDINT;
iState : USINT;
iTerminalType : WORD;
iSpecialType : WORD;
iFirmwareVersion : WORD;
sDescription : STRING;
bDimRampAbsolute : BOOL;
sRampTime : STRING;
bWatchdogDisable : BOOL;
tWatchdogTimeout : TIME;
iTimeoputOffValue : UINT;
iTimeoputOffValue : UINT;
sDimmerMode : STRING;
aAfterShortCirciut : STRING;
sLineFrequency : STRING;
```

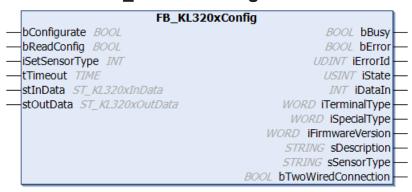
Name	Тур	Beschreibung
bBusy	BOOL	Solange eine Lese- oder Konfigurationssequenz abgearbeitet wird, steht dieser Ausgang auf TRUE.
bError	BOOL	Dieser Ausgang wird auf TRUE geschaltet, wenn bei der Ausführung eines Befehls (Konfigurieren oder Lesen) ein Fehler aufgetreten ist. Der befehlsspezifische Fehlercode ist in <i>iErrorld</i> enthalten.
iErrorld	UDINT	Enthält den befehlsspezifischen Fehlercode des zuletzt ausgeführten Befehls. Wird durch das erneute Aktivieren des Bausteins über die Eingänge bConfigurate oder bReadConfig wieder auf 0 zurückgesetzt. Siehe Fehlercodes [• 162].
iState	USINT	Entspricht der Statusvariablen der Prozessdaten stInData.iState, siehe VAR_IN_OUT. Während der Befehlsausführung (bBusy = TRUE) ist der Ausgang auf 0 gesetzt. Damit eignet sich dieser Ausgang zur Statusbeurteilung im Normalbetrieb der Klemme. Störende Zustände während des Konfigurierens und Lesens durch die Registerkommunikation werden ausgeblendet.
iTerminalType	WORD	Inhalt des Registers 8 (Klemmenbezeichnung). Bei Anwendung mit der richtigen Klemme sollte der Inhalt 0x05DD (1501dez) sein.
iSpecialType	WORD	Inhalt des Registers 29 (Sondervariante)
iFirmwareVersion	WORD	Inhalt des Registers 9 (Firmware-Stand)
sDescription	STRING	Klemmenbezeichnung, Sondervariante und Version der Firmware als String (z. B. 'Terminal KL1501-0000 / Firmware 1C').
bDimRampAbsolute	BOOL	TRUE: Dimmrampe ist als absolut eingestellt, d.h. jeder Schaltschritt benötigt dieselbe Rampenzeit, welche unter iSetRampTime eingestellt ist.
sRampTime	STRING	Eingestellte Rampenzeit als Klartext.
bWatchdogDisable	BOOL	TRUE: Watchdog ist deaktiviert.
tWatchdogTimeout	TIME	Eingestellte Watchdogzeit.
iTimeoutOnValue	UINT	Eingestellter Lichtwert, der bei einem Feldbusfehler und aktuellen Prozessdaten > 0 ausgegeben wird.



Name	Тур	Beschreibung
iTimeoutOffValue	UINT	Eingestellter Lichtwert, der bei einem Feldbusfehler und aktuellen Prozessdaten = 0 ausgegeben wird.
sDimmerMode	STRING	Eingestellter Dimmermodus als Klartext.
sAfterShortCircuit	STRING	Eingestelltes Verhalten nach Kurzschluss als Klartext.
sLineFrequency	STRING	Eingestellte Netzfrequenz als Klartext.

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken
TwinCAT v3.1.4018.26	PC/CX	· · · · · · · · · · · · · · · · · · ·	Tc2_loFunctions ab v3.3.5.0

3.6.3 FB_KL320xConfig



Der Funktionsbaustein FB_KL320xConfig dient zur Parametrierung einer <u>KL3201, KL3202 oder KL3204</u>: Eingangsklemme für Widerstandssensoren.



Der Baustein parametriert nur einen Klemmenkanal. Zur Parametrierung aller Kanäle ist die entsprechende Anzahl von Bausteinen zu instanziieren. Eine Mischkonfiguration (z.B. unterschiedliche Sensortypen) ist möglich.

Eingänge

VAR_INPUT
bConfigurate : BOOL;
bReadConfig : BOOL;
iSetSensorType : INT;
tTimeout : TIME;
END VAR

Name	Тур	Beschreibung
bConfigurate	BOOL	Eine steigende Flanke startet die Konfigurationssequenz. Zunächst werden die allgemeinen Klemmendaten "Klemmbezeichnung", "Sondervariante" und "Firmwarestand" ausgelesen. Danach werden die angegebenen Einstellungen in die entsprechenden Register geschrieben und abschließend zur Sicherheit und Information noch einmal ausgelesen. Die gelesenen Informationen werden an den Bausteinausgängen angezeigt. Während des Ablaufs dieser Sequenz steht der Ausgang bBusy auf TRUE und es wird kein weiterer Befehl, wie etwa bReadConfig, entgegengenommen.
bReadConfig	BOOL	Eine steigende Flanke startet eine Lesesequenz. Es werden die allgemeinen Klemmendaten "Klemmbezeichnung", "Sondervariante" und "Firmwarestand" ausgelesen und im Anschluss daran die



Name	Тур	Beschreibung
		eingestellten Konfigurationsparameter. Die gelesenen Informationen werden an den Bausteinausgängen angezeigt. Während der Lesesequenz steht der Ausgang bBusy auf TRUE und es wird kein weiterer Befehl, wie etwa bConfigurate, entgegengenommen.
iSetSensorType	INT	An diesem Eingang ist der verwendete Sensor einzustellen. Die Einstellung erfolgt nach unten aufgeführter Tabelle.
tTimeout	TIME	Innerhalb der hier eingetragenen Zeit muss die Konfiguration der Klemme bzw. das Auslesen der Konfiguration abgeschlossen sein. Anderenfalls wird ein Fehler mit entsprechender Fehlernummer an den Ausgängen bError und iErrorld ausgegeben.

iSetSensorType	Element
0	PT100
1	NI100
2	PT1000
3	PT500
4	PT200
5	NI1000
6	NI120
7	Ausgabe 10,0 Ω - 5000,0 Ω
8	Ausgabe 10,0 Ω - 1200,0 Ω
9	PT1000 - Zwei-Leiter-Anschluss - nicht zulässig bei Verwendung einer KL3204!

▼/ ■ Ein-/Ausgänge

```
VAR_IN_OUT
stInData : ST_KL1501InData;
stOutData : ST_KL1501OutData;
END VAR
```

Name	Тур	Beschreibung
stInData	ST_KL1501InData [▶ 153]	Verweis auf die Struktur des Eingangsprozessabbildes.
stOutData	ST_KL1501OutData [▶ 154]	Verweis auf die Struktur des Ausgangsprozessabbildes.

Ausgänge

```
VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
iErrorId : UDINT;
iState : USINT;
iDataIn : INT;
iTerminalType : WORD;
iSpecialType : WORD;
iFirmwareVersion : WORD;
sDescription : STRING;
sSensorType : STRING;
bTwoWiredConnection : BOOL;
END_VAR
```

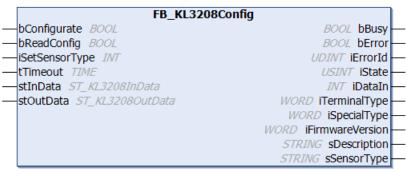
Name	Тур	Beschreibung
bBusy	BOOL	Solange eine Lese- oder Konfigurationssequenz
-		abgearbeitet wird, steht dieser Ausgang auf TRUE.



Name	Тур	Beschreibung
bError	BOOL	Dieser Ausgang wird auf TRUE geschaltet, wenn bei der Ausführung eines Befehls (Konfigurieren oder Lesen) ein Fehler aufgetreten ist. Der befehlsspezifische Fehlercode ist in <i>iErrorld</i> enthalten.
iErrorld	UDINT	Enthält den befehlsspezifischen Fehlercode des zuletzt ausgeführten Befehls. Wird durch das erneute Aktivieren des Bausteins über die Eingänge bConfigurate oder bReadConfig wieder auf 0 zurückgesetzt. Siehe
		Fehlercodes [▶ 162].
iState	USINT	Entspricht der Statusvariablen der Prozessdaten stInData.iState, siehe VAR_IN_OUT. Während der Befehlsausführung (bBusy = TRUE) ist der Ausgang auf 0 gesetzt. Damit eignet sich dieser Ausgang zur Statusbeurteilung im Normalbetrieb der Klemme. Störende Zustände während des Konfigurierens und Lesens durch die Registerkommunikation werden ausgeblendet.
iDataIn	UDINT	Entspricht der Datenvariablen der Prozessdaten stInData.iDataIn, siehe VAR_IN_OUT. Während der Befehlsausführung (bBusy = TRUE) behält dieser Ausgang den Wert, den er vor dem Befehlsaufruf innehatte. Damit eignet sich der Ausgang zur direkten Prozessdatenverarbeitung im Normalbetrieb der Klemme. Störende Zustände während des Konfigurierens und Lesens durch die Registerkommunikation werden ausgeblendet.
iTerminalType	WORD	Inhalt des Registers 8 (Klemmenbezeichnung). Bei Anwendung mit der richtigen Klemme sollte der Inhalt 0x05DD (1501dez) sein.
iSpecialType	WORD	Inhalt des Registers 29 (Sondervariante)
iFirmwareVersion	WORD	Inhalt des Registers 9 (Firmware-Stand)
sDescription	STRING	Klemmenbezeichnung, Sondervariante und Version der Firmware als String (z. B. 'Terminal KL1501-0000 / Firmware 1C').
sSensorType	STRING	Eingestellter Sensortyp als Klartext.
bTwoWiredConnecti on	BOOL	Sensortyp ist im Zweileiter-Anschluss parametriert.

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken
TwinCAT v3.1.4018.26	PC/CX	KL3201, KL3202, KL3204	Tc2_loFunctions ab v3.3.5.0

3.6.4 FB_KL3208Config





Der Funktionsbaustein FB_KL3208Config dient zur Parametrierung einer <u>KL3208-0010</u>: 8-Kanal Eingangsklemme für Widerstandssensoren.



Der Baustein parametriert nur einen Klemmenkanal. Zur Parametrierung aller Kanäle ist die entsprechende Anzahl von Bausteinen zu instanziieren. Eine Mischkonfiguration (z.B. unterschiedliche Sensortypen) ist möglich.

Eingänge

VAR_INPUT
bConfigurate : BOOL;
bReadConfig : BOOL;
iSetSensorType : INT;
tTimeout : TIME;
END_VAR

Name	Тур	Beschreibung
bConfigurate	BOOL	Eine steigende Flanke startet die Konfigurationssequenz. Zunächst werden die allgemeinen Klemmendaten "Klemmbezeichnung", "Sondervariante" und "Firmwarestand" ausgelesen. Danach werden die angegebenen Einstellungen in die entsprechenden Register geschrieben und abschließend zur Sicherheit und Information noch einmal ausgelesen. Die gelesenen Informationen werden an den Bausteinausgängen angezeigt. Während des Ablaufs dieser Sequenz steht der Ausgang bBusy auf TRUE und es wird kein weiterer Befehl, wie etwa bReadConfig, entgegengenommen.
bReadConfig	BOOL	Eine steigende Flanke startet eine Lesesequenz. Es werden die allgemeinen Klemmendaten "Klemmbezeichnung", "Sondervariante" und "Firmwarestand" ausgelesen und im Anschluss daran die eingestellten Konfigurationsparameter. Die gelesenen Informationen werden an den Bausteinausgängen angezeigt. Während der Lesesequenz steht der Ausgang bBusy auf TRUE und es wird kein weiterer Befehl, wie etwa bConfigurate, entgegengenommen.
iSetSensorType	INT	An diesem Eingang ist der verwendete Sensor einzustellen. Die Einstellung erfolgt nach unten aufgeführter Tabelle.
tTimeout	TIME	Innerhalb der hier eingetragenen Zeit muss die Konfiguration der Klemme bzw. das Auslesen der Konfiguration abgeschlossen sein. Anderenfalls wird ein Fehler mit entsprechender Fehlernummer an den Ausgängen bError und iErrorld ausgegeben.

iSetSensorType	Element
0	PT1000
1	NI1000
2	RSNI1000 (NI1000 nach Landis&Staefa-Charakteristik: 1000 Ω bei 0 °C und 1500 Ω bei 100 °C.)
3	NTC1K8
4	NTC1K8_TK
5	NTC2K2
6	NTC3K
7	NTC5K
8	NTC10K
9	NTC10KPRE
10	NTC10K_3204



iSetSensorType	Element
11	NTC10KTYP2
12	NTC10KTYP3
13	NTC10KDALE
14	NTC10K3A221
15	NTC20K
16	Poti, Auflösung 0,1 Ω
17	Poti, Auflösung 1 Ω
18	NTC100K

▼/ ■ Ein-/Ausgänge

VAR_IN_OUT

stInData : ST_KL1501InData; stOutData : ST_KL1501OutData;

END_VAR

Name	Тур	Beschreibung
stInData	ST_KL1501InData [▶ 153]	Verweis auf die Struktur des Eingangsprozessabbildes.
stOutData	ST_KL1501OutData [▶ 154]	Verweis auf die Struktur des Ausgangsprozessabbildes.

Ausgänge

VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
iErrorId : UDINT;
iState : USINT;
iDataIn : INT;
iTerminalType : WORD;
iSpecialType : WORD;
iFirmwareVersion : WORD;
sDescription : STRING;
sSensorType : STRING;

Name	Тур	Beschreibung
bBusy	BOOL	Solange eine Lese- oder Konfigurationssequenz abgearbeitet wird, steht dieser Ausgang auf TRUE.
bError	BOOL	Dieser Ausgang wird auf TRUE geschaltet, wenn bei der Ausführung eines Befehls (Konfigurieren oder Lesen) ein Fehler aufgetreten ist. Der befehlsspezifische Fehlercode ist in <i>iErrorld</i> enthalten.
iErrorld	UDINT	Enthält den befehlsspezifischen Fehlercode des zuletzt ausgeführten Befehls. Wird durch das erneute Aktivieren des Bausteins über die Eingänge bConfigurate oder bReadConfig wieder auf 0 zurückgesetzt. Siehe Fehlercodes [• 162].
iState	USINT	Entspricht der Statusvariablen der Prozessdaten stlnData.iState, siehe VAR_IN_OUT. Während der Befehlsausführung (bBusy = TRUE) ist der Ausgang auf 0 gesetzt. Damit eignet sich dieser Ausgang zur Statusbeurteilung im Normalbetrieb der Klemme. Störende Zustände während des Konfigurierens und Lesens durch die Registerkommunikation werden ausgeblendet.
iDataIn	UDINT	Entspricht der Datenvariablen der Prozessdaten stlnData.iDataln, siehe VAR_IN_OUT. Während der Befehlsausführung (bBusy = TRUE) behält dieser Ausgang den Wert, den er vor dem Befehlsaufruf innehatte. Damit eignet sich der Ausgang zur direkten Prozessdatenverarbeitung im Normalbetrieb der Klemme.



Name	Тур	Beschreibung
		Störende Zustände während des Konfigurierens und Lesens durch die Registerkommunikation werden ausgeblendet.
iTerminalType	WORD	Inhalt des Registers 8 (Klemmenbezeichnung). Bei Anwendung mit der richtigen Klemme sollte der Inhalt 0x05DD (1501dez) sein.
iSpecialType	WORD	Inhalt des Registers 29 (Sondervariante)
iFirmwareVersion	WORD	Inhalt des Registers 9 (Firmware-Stand)
sDescription	STRING	Klemmenbezeichnung, Sondervariante und Version der Firmware als String (z. B. 'Terminal KL1501-0000 / Firmware 1C').
sSensorType	STRING	Eingestellter Sensortyp als Klartext.

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken
TwinCAT v3.1.4018.26	PC/CX	KL3201, KL3202, KL3204	Tc2_loFunctions ab v3.3.5.0

3.6.5 FB_KL3228Config



Der Funktionsbaustein FB_KL3228Config dient zur Parametrierung einer <u>KL3228</u>: 8-Kanal Eingangsklemme für Widerstandssensoren.



Der Baustein parametriert nur einen Klemmenkanal. Zur Parametrierung aller Kanäle ist die entsprechende Anzahl von Bausteinen zu instanziieren. Eine Mischkonfiguration (z.B. unterschiedliche Sensortypen) ist möglich.

Eingänge

VAR_INPUT
bConfigurate : BOOL;
bReadConfig : BOOL;
iSetSensorType : INT;
tTimeout : TIME;
END_VAR

Name	Тур	Beschreibung
bConfigurate	BOOL	Eine steigende Flanke startet die Konfigurationssequenz. Zunächst werden die allgemeinen Klemmendaten "Klemmbezeichnung", "Sondervariante" und "Firmwarestand" ausgelesen. Danach werden die angegebenen Einstellungen in die entsprechenden Register geschrieben und abschließend zur Sicherheit und Information noch einmal ausgelesen. Die gelesenen Informationen werden an den Bausteinausgängen



Name	Тур	Beschreibung
		angezeigt. Während des Ablaufs dieser Sequenz steht der Ausgang <i>bBusy</i> auf TRUE und es wird kein weiterer Befehl, wie etwa <i>bReadConfig</i> , entgegengenommen.
bReadConfig	BOOL	Eine steigende Flanke startet eine Lesesequenz. Es werden die allgemeinen Klemmendaten "Klemmbezeichnung", "Sondervariante" und "Firmwarestand" ausgelesen und im Anschluss daran die eingestellten Konfigurationsparameter. Die gelesenen Informationen werden an den Bausteinausgängen angezeigt. Während der Lesesequenz steht der Ausgang bBusy auf TRUE und es wird kein weiterer Befehl, wie etwa bConfigurate, entgegengenommen.
iSetSensorType	INT	An diesem Eingang ist der verwendete Sensor einzustellen. Die Einstellung erfolgt nach unten aufgeführter Tabelle.
tTimeout	TIME	Innerhalb der hier eingetragenen Zeit muss die Konfiguration der Klemme bzw. das Auslesen der Konfiguration abgeschlossen sein. Anderenfalls wird ein Fehler mit entsprechender Fehlernummer an den Ausgängen bError und iErrorld ausgegeben.

iSetSensorType	Element
0	PT1000
1	NI1000
	RSNI1000 (NI1000 nach Landis&Staefa-Charakteristik: 1000 Ω bei 0 °C und 1500 Ω bei 100 °C.)

▼/ Ein-/Ausgänge

VAR_IN_OUT
stInData : ST_KL1501InData;
stOutData : ST_KL1501OutData;
END_VAR

Name	Тур	Beschreibung
stInData	ST_KL1501InData [▶ 153]	Verweis auf die Struktur des Eingangsprozessabbildes.
stOutData	ST KL1501OutData [▶ 154]	Verweis auf die Struktur des Ausgangsprozessabbildes.

Ausgänge

VAR_OUTPUT		
bBusy	:	BOOL;
bError	:	BOOL;
iErrorId	:	UDINT;
iState	:	USINT;
iDataIn	:	INT;
iTerminalType	:	WORD;
iSpecialType	:	WORD;
iFirmwareVersion	:	WORD;
sDescription	:	STRING;
sSensorType	:	STRING;
END_VAR		

Name	Тур	Beschreibung
bBusy	BOOL	Solange eine Lese- oder Konfigurationssequenz abgearbeitet wird, steht dieser Ausgang auf TRUE.
bError		Dieser Ausgang wird auf TRUE geschaltet, wenn bei der Ausführung eines Befehls (Konfigurieren oder Lesen) ein Fehler aufgetreten ist. Der befehlsspezifische Fehlercode ist in <i>iErrorld</i> enthalten.



Name	Тур	Beschreibung
iErrorld	UDINT	Enthält den befehlsspezifischen Fehlercode des zuletzt ausgeführten Befehls. Wird durch das erneute Aktivieren des Bausteins über die Eingänge bConfigurate oder bReadConfig wieder auf 0 zurückgesetzt. Siehe Fehlercodes [• 162].
iState	USINT	Entspricht der Statusvariablen der Prozessdaten stInData.iState, siehe VAR_IN_OUT. Während der Befehlsausführung (bBusy = TRUE) ist der Ausgang auf 0 gesetzt. Damit eignet sich dieser Ausgang zur Statusbeurteilung im Normalbetrieb der Klemme. Störende Zustände während des Konfigurierens und Lesens durch die Registerkommunikation werden ausgeblendet.
iDataIn	UDINT	Entspricht der Datenvariablen der Prozessdaten stInData.iDataIn, siehe VAR_IN_OUT. Während der Befehlsausführung (bBusy = TRUE) behält dieser Ausgang den Wert, den er vor dem Befehlsaufruf innehatte. Damit eignet sich der Ausgang zur direkten Prozessdatenverarbeitung im Normalbetrieb der Klemme. Störende Zustände während des Konfigurierens und Lesens durch die Registerkommunikation werden ausgeblendet.
iTerminalType	WORD	Inhalt des Registers 8 (Klemmenbezeichnung). Bei Anwendung mit der richtigen Klemme sollte der Inhalt 0x05DD (1501dez) sein.
iSpecialType	WORD	Inhalt des Registers 29 (Sondervariante)
iFirmwareVersion	WORD	Inhalt des Registers 9 (Firmware-Stand)
sDescription	STRING	Klemmenbezeichnung, Sondervariante und Version der Firmware als String (z. B. 'Terminal KL1501-0000 / Firmware 1C').
sSensorType	STRING	Eingestellter Sensortyp als Klartext.

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken
TwinCAT v3.1.4018.26	PC/CX	KL3201, KL3202, KL3204	Tc2_loFunctions ab v3.3.5.0

3.7 CANopen

3.7.1 IOF_CAN_Layer2Command



Der Funktionsbaustein IOF_CAN_Layer2Command sendet ein 10 Byte langes Kommando an die Schicht 2 eines CAN-Masters.



Eingänge

VAR INPUT NETID : T AmsNetId; DEVICEID : UDINT; BOXADDR : WORD; START : BOOL; TMOUT : TIME := DEFAULT_ADS_TIMEOUT; VAR

END VAR

Name	Тур	Beschreibung	
NETID	T_AmsNetId	Hier kann die Netzwerkadresse des TwinCAT-Rechners angegeben werden, auf dem die Funktion ausgeführt werden soll. Für den lokalen Rechner kann auch ein Leerstring angegeben werden.	
DEVICEID	UDINT	Über die Deviceld (Geräte-ID) wird das Gerät (CAN-Master) spezifiziert, auf dem die Funktion ausgeführt werden soll. Die Geräte-IDs werden während der Hardware-Konfiguration von TwinCAT-System festgelegt.	
LEN	UDINT	Die Bytelänge des Layer 2 Kommandos.	
SRCADDR	PVOID	Die Adresse von dem ersten Datenwort des CAN-Layer 2 Kommandos.	
START	BOOL	Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.	
TMOUT	TIME	Gibt die Timeout-Zeit an, die bei der Ausführung des Kommandos nicht überschritten werden darf.	

Beispiel:

```
PROGRAM MAIN
Layer2Command1 : IOF_CAN_Layer2Command

Layer2Data : ARRAY[1..5] OF WORD;

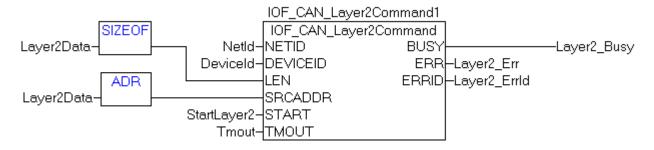
StartLayer2 : BOOL;

Layer2_Busy : BOOL;

Layer2_Err : BOOL;

Layer2_ErrId : UDINT;

END_VAR
         IOF CAN Layer2Command1 : IOF CAN Layer2Command;
```



Ausgänge

VAR OUTPUT

: BOOL; BUSY : BOOL; ERR ERRID : UDINT; END VAR

Name	Тур	Beschreibung
BUSY		Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
ERR		Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der

TE1000 Version: 1.5.3 69

BUSY-Ausgang zurückgesetzt wurde.



Name	Тур	Beschreibung	
ERRID	UDINT	Liefert bei einem gesetzten ERR-Ausgang die ADS-	
		<u>Fehlernummer [▶ 163]</u> .	

Entwicklungsumge- bung	Zielplattform		Einzubindende SPS-Bibliotheken (Kategorie-gruppe)
TwinCAT v3.1.0	Funktionalität	HILSCHER CIF3xx COM master card	Tc2_loFunctions (IO)

3.8 NOV/DP-RAM

3.8.1 FB_NovRamReadWrite

Die folgende Beschreibung bezieht sich auf die TwinCAT Version 2.8. Ab der TwinCAT Version 2.9 [Build 927] wird kein Funktionsbaustein zum Schreiben bzw. Lesen von zu sichernden SPS-Daten ins NOVRAM mehr benötigt.

```
FB_NovRamReadWrite

nDevId UDINT BOOL bBusy
bRead BOOL BOOL bError
bWrite BOOL UDINT nErrId
cbSrcLen UDINT UDINT cbRead
cbDestLen UDINT UDINT cbWrite
pSrcAddr PVOID
pDestAddr PVOID
tTimeOut TIME
```

Der Funktionsbaustein FB_NovRamReadWrite greift aus einem SPS-Programm auf das NOV-RAM der FCxxxx-000**2** Feldbuskarten zu. Bei einer steigenden Flanke am *bRead* oder *bWrite*-Eingang wird der Funktionsbaustein aktiviert und eine entsprechende Anzahl Datenbytes aus dem NOV-RAM gelesen bzw. in das NOV-RAM geschrieben. Wurden gleichzeitig beide Eingänge: *bRead* und *bWrite* gesetzt, dann werden die Daten zuerst in das NOV-RAM geschrieben und dann zurückgelesen.

Bemerkungen:

Um den Addresspointer auf das NOV-RAM zu ermitteln, wird von dem FB_NovRamReadWrite-Funktionsbaustein intern eine Instanz des ADSREAD-Funktionsbausteines verwendet. Dieser Addresspointer wird aber nur beim ersten Aufruf des FB_NovRamReadWrite-Funktionsbausteins und bei einer Änderung von nDevId neu ermittelt. Dafür werden mehrere SPS-Zyklen benötigt. Um die Daten in das NOV-RAM zu schreiben oder aus dem NOV-RAM zu lesen wird die Funktion MEMCPY verwendet. Dadurch können die Daten im gleichen SPS-Zyklus geschrieben bzw. gelesen werden. Intern wird auch die maximale Bytelänge des NOV-RAM ermittelt, und die maximale Länge der Daten, die gelesen oder geschrieben werden können, auf diese Länge begrenzt.

Eingänge

```
VAR_INPUT

nDevId : UDINT;

bRead : BOOL;

bWrite : BOOL;

cbSrcLen : UDINT;

cbDestLen : UDINT;

pSrcAddr : PVOID;

pDestAddr : PVOID;

tTimeOut : TIME := DEFAULT_ADS_TIMEOUT;

END_VAR
```



Name	Тур	Beschreibung	
nDevId	UDINT	Die Geräte-Id einer NOVRAM-Karte. Über die Id wird das NOVRAM einer FCxxxx-0002 Karte spezifiziert, auf das mit dem Funktionsbaustein schreibend oder lesend zugegriffen werden soll. Die Geräte-Ids werden während der Hardware-Konfiguration vom TwinCAT-System festgelegt.	
bRead	BOOL	Bei einer positiven Flanke an diesem Eingang wird der Baustein aktiviert und <i>cbDestLen</i> -Daten aus dem NOVRAM (ab dem Addressoffset NULL) in den Puffer mit der Adresse <i>pDestAddr</i> hineinkopiert.	
bWrite	BOOL	Bei einer positiven Flanke an diesem Eingang wird der Baustein aktiviert und cbSrcLen-Daten aus dem Puffer mit der Adresse pSrcAddr in das NOVRAM (ab dem Addressoffset NULL) hineinkopiert.	
cbSrcLen	UDINT	Die Bytelänge der Daten, die in das NOVRAM geschriebe werden sollen.	
cbDestLen	UDINT	Die Bytelänge der Daten, die aus dem NOVRAM gelesen werden sollen.	
pSrcAddr	PVOID	Der Addresspointer auf einen Datenpuffer mit den Daten, die in das NOV-RAM geschrieben werden sollen. Der Addresspointer kann mit dem ADR-Operator ermittelt werden.	
pDestAddr	PVOID	Der Addresspointer auf einen Datenpuffer, in den die gelesenen NOV-RAM-Daten hineinkopiert werden sollen.	
tTimeOut	TIME	Gibt die Timeout-Zeit an, die bei der Ausführung des Kommandos/Funktion nicht überschritten werden darf.	

Ausgänge

VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
nErrId : UDINT;
cbRead : UDINT;
cbWrite : UDINT;
END_VAR

Name	Тур	Beschreibung
bBusy	BOOL	Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis die Ausführung abgeschlossen wurde.
bError	BOOL	Sollte ein Fehler bei der Ausführung erfolgen, dann wird dieser Ausgang gesetzt.
nErrld	UDINT	Liefert bei einem gesetzten <i>bError</i> -Ausgang die <u>ADS-</u> <u>Fehlernummer [▶ 163]</u> .
cbRead	UDINT	Anzahl der erfolgreich gelesenen Datenbytes.
cbWrite	UDINT	Anzahl der erfolgreich geschriebenen Datenbytes.

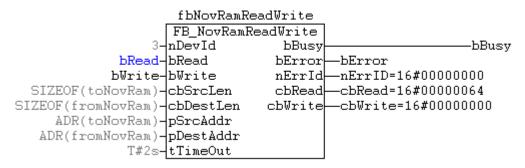
Beispiel:

```
PROGRAM MAIN

VAR

fbNovRamReadWrite: FB_NovRamReadWrite;
bRead: BOOL;
bWrite: BOOL;
fromNovRam: ARRAY[1..100] OF BYTE;
toNovRam: ARRAY[1..100] OF BYTE;
bBusy: BOOL;
bError: BOOL;
nErrID: UDINT;
cbRead: UDINT;
cbWrite: UDINT;
```



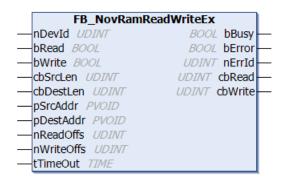


Im Beispiel wurden bei einer steigenden Flanke am *bRead*-Eingang 100 Byte Daten aus dem NOV-RAM ausgelesen und in das Array *fromNovRam* hineinkopiert.

Voraussetzungen

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	\ /	FCxxxx cards mit NOV- RAM (FCxxxx-0002)	Tc2_loFunctions (IO)

3.8.2 FB_NovRamReadWriteEx



Der Funktionsbaustein FB_NovRamReadWriteEx greift aus einem SPS-Programm auf das NOV-RAM zu (z.B. der FCxxxx-0002 Feldbuskarten, CX9000 NOV-RAM etc.). Bei einer steigenden Flanke am *bRead* oder *bWrite*-Eingang wird der Funktionsbaustein aktiviert und eine entsprechende Anzahl Datenbytes aus dem NOV-RAM gelesen bzw. in das NOVRAM geschrieben. Wurden gleichzeitig beide Eingänge: *bRead* und *bWrite* gesetzt, dann werden die Daten zuerst in das NOVRAM geschrieben und dann zurück gelesen. Der Unterschied zum Baustein FB_NovRamReadWrite ist, dass der Adressoffset im NOVRAM für Schreib- und Lesezugriffe angegeben werden kann. Außerdem überprüft der Baustein die erlaubte Zugriffsart auf den NOVRAM Speicher und kopiert wenn nötig die Daten byteweise in den NOVRAM Speicher, anstatt ein MEMCPY zu verwenden. Das NOVRAM vom CX9000 erlaubt z.B. nur Byte-Zugriffe und der Baustein FB_NovRamReadWrite würde in diesem Fall einen Fehler zurückliefern.

Bemerkungen:

Um den Addresspointer auf das NOVRAM zu ermitteln, wird von dem FB_NovRamReadWriteEx-Funktionsbaustein intern eine Instanz des ADSREAD-Funktionsbausteines verwendet. Dieser Addresspointer wird aber nur beim ersten Aufruf des FB_NovRamReadWriteEx-Funktionsbausteins und bei einer Änderung von nDevId neu ermittelt. Dafür werden mehrere SPS-Zyklen benötigt. Um die Daten in das NOVRAM zu schreiben oder aus dem NOVRAM zu lesen wird direkt auf den NOVRAM Speicher zugegriffen. Dadurch können die Daten im gleichen SPS-Zyklus geschrieben bzw. gelesen werden. Intern wird auch die maximale Bytelänge des NOVRAM ermittelt, und die maximale Länge der Daten, die gelesen oder geschrieben werden können, auf diese Länge begrenzt.



Eingänge

VAR_INPUT

nDevId : UDINT;
bRead : BOOL;
bWrite : BOOL;
cbSrcLen : UDINT;
cbDestLen : UDINT;
pSrcAddr : PVOID;
pDestAddr : PVOID;
nReadOffs : UDINT;
nWriteOffs : UDINT;
tTimeOut : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR

Name	Тур	Beschreibung
nDevId	UDINT	Die Geräte-Id einer NOVRAM-Karte. Über die Id wird das NOVRAM einer FCxxxx-0002 Karte spezifiziert, auf das mit dem Funktionsbaustein schreibend oder lesend zugegriffen werden soll. Die Geräte-Ids werden während der Hardware-Konfiguration vom TwinCAT-System festgelegt.
bRead	BOOL	Bei einer positiven Flanke an diesem Eingang wird der Baustein aktiviert und <i>cbDestLen</i> -Daten aus dem NOVRAM (ab dem Addressoffset NULL) in den Puffer mit der Adresse <i>pDestAddr</i> hineinkopiert.
bWrite	BOOL	Bei einer positiven Flanke an diesem Eingang wird der Baustein aktiviert und cbSrcLen-Daten aus dem Puffer mit der Adresse pSrcAddr in das NOVRAM (ab dem Addressoffset NULL) hineinkopiert.
cbSrcLen	UDINT	Die Bytelänge der Daten, die in das NOVRAM geschrieben werden sollen.
cbDestLen	UDINT	Die Bytelänge der Daten, die aus dem NOVRAM gelesen werden sollen.
pSrcAddr	PVOID	Der Addresspointer auf einen Datenpuffer mit den Daten, die in das NOV-RAM geschrieben werden sollen. Der Addresspointer kann mit dem ADR-Operator ermittelt werden.
pDestAddr	PVOID	Der Addresspointer auf einen Datenpuffer, in den die gelesenen NOV-RAM-Daten hineinkopiert werden sollen.
nReadOffs	UDINT	Der Adressoffset im NOVRAM ab dem gelesen werden soll.
nWriteOffs	UDINT	Der Adressoffset im NOVRAM ab dem geschrieben werden soll.
tTimeOut	TIME	Gibt die Timeout-Zeit an, die bei der Ausführung des Kommandos/Funktion nicht überschritten werden darf.

Ausgänge

VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
nErrId : UDINT;
cbRead : UDINT;
cbWrite : UDINT;
END_VAR

Name	Тур	Beschreibung
bBusy	BOOL	Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis die Ausführung abgeschlossen wurde.
bError	BOOL	Sollte ein Fehler bei der Ausführung erfolgen, dann wird dieser Ausgang gesetzt.



Name	Тур	Beschreibung
nErrld	UDINT	Liefert bei einem gesetzten bError-Ausgang die ADS-
		Fehlernummer [163].
cbRead	UDINT	Anzahl der erfolgreich gelesenen Datenbytes.
cbWrite	UDINT	Anzahl der erfolgreich geschriebenen Datenbytes.

Beispiel:

```
PROGRAM MAIN
   fbNovRamReadWrite : FB NovRamReadWriteEx;
          : BOOL;
: BOOL;
   bRead
   bWrite
   fromNovRam : ARRAY[1..100] OF BYTE;
   toNovRam
                    : ARRAY[1..100] OF BYTE;
   bBusy
                     : BOOL;
   bError
                    : BOOL;
   nErrID
                     : UDINT;
                    : UDINT;
   cbRead
                    : UDINT;
   cbWrite
   readOffs
                     : UDINT :=0;
   writeOffs
                    : UDINT:=0;
   cbToWrite
                     : UDINT := 100;
: UDINT := 100;
   cbToRead
END VAR
```

fbNovRamReadWrite FB_NovRamReadWriteEx 1-nDevid bBusy -bBusy bRead-bRead bError -bError bWrite-bWrite nErrId -nErrld cbToWrite-cbSrcLen -cbRead cbRead cbToRead-cbDestLen cbWrite| -cbWrite ADR(toNovRam)-pSrcAddr ADR(fromNovRam)-pDestAddr readOffs-InReadOffs writeOffs-InWriteOffs t#5s-|tTimeOut

Im Beispiel wurden bei einer steigenden Flanke am *bRead*-Eingang 100 Byte Daten aus dem NOV-RAM ausgelesen und in das Array *fromNovRam* hineinkopiert.

Voraussetzungen

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	- ()	FCxxxx cards mit NOV- RAM (FCxxxx-0002)	Tc2_loFunctions (IO)

3.8.3 FB_GetDPRAMInfo

```
FB_GetDPRAMInfo

nDevId UDINT BOOL bBusy bExecute BOOL bError UDINT nErrid ST_NovRamAddrInfo stInfo
```

Der Funktionsbaustein FB_GetDPRAMInfo ermittelt den Addresspointer und die konfigurierte Größe des NOV/DP-RAM einer Feldbuskarte. Der Addresspointer kann z.B. dazu benutzt werden, um direkt auf das NOV-RAM der FCxxx-0002 (Beckhoff) Karten oder DPRAM der von TwinCAT nicht unterstützten Karten (Dritthersteller) schreibend oder lesend zugreifen zu können. Die Karte muss vorher in TwinCAT System als



allgemeines NOV/DP-RAM konfiguriert werden. Im SPS-Programm können dann die MEMCPY-, MEMSEToder MEMCMP-Funktionen benutzt werden um auf einen beliebigen Speicheroffset schreibend/lesend zugreifen zu können.

Wenn das NOV/DP-RAM einen speziellen Zugriff (z.B. BYTE, aligned WORD) erfordert gibt dieser Funktionsbaustein einen *Service Not Supported* (16#701) Fehler zurück. In diesem Fall kann der Baustein FB_NovRamReadWriteEx verwendet werden um Daten aus dem NOV/DP-RAM zu lesen oder Daten in das NOV/DP-RAM zu schreiben. Z.B. das NOV-RAM vom *CX9000* erlaubt nur BYTE-Zugriffe, so dass in diesem Falle der Baustein FB_NovRamReadWriteEx verwendet werden sollte.

Eingänge

```
VAR_INPUT

nDevId : UDINT;

bExecute : BOOL;

tTimeOut : TIME := DEFAULT_ADS_TIMEOUT;

END VAR
```

Name	Тур	Beschreibung
nDevId	UDINT	Die Geräte-Id einer NOV/DPRAM-Karte. Über die Id wird die Karte spezifiziert, deren Informationen gelesen werden sollen. Die Geräte-Ids werden während der Hardware-Konfiguration vom TwinCAT-System festgelegt.
bExecute	BOOL	Bei einer positiven Flanke an diesem Eingang wird der Baustein aktiviert.
tTimeOut	TIME	Gibt die Timeout-Zeit an, die bei der Ausführung des Kommandos/Funktion nicht überschritten werden darf.

Ausgänge

```
VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
nErrId : UDINT;
stInfo : ST_NovRamAddrInfo;
END_VAR
```

Name	Тур	Beschreibung
bBusy		Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis die Ausführung abgeschlossen wurde.
bError		Sollte ein Fehler bei der Ausführung erfolgen, dann wird dieser Ausgang gesetzt.
nErrld	UDINT	Liefert bei einem gesetzten <i>bError</i> -Ausgang die <u>ADS-</u> <u>Fehlernummer [▶ 163]</u> .
stInfo	ST_NovRamAddrInfo [▶ 142]	Eine Struktur mit der Adresse und Größe des NOV/DP-RAM.

Beispiel:

In MAIN wird ein Pointer auf eine Instanz der ST_NOVRAM-Strukturvariablen deklariert. Dieser Pointer wird beim Programmstart mit der Adresse des NOV-/DP-RAM initialisiert. Durch Referenzieren des Pointers kann auf die einzelnen Strukturelemente (und das NOV-/DP-RAM) schreibend oder lesend zugegriffen werden.

```
TYPE ST_NOVRAM:
STRUCT

dwParam_0 : DWORD;
dwParam_1 : DWORD;
dwParam_2 : DWORD;
dwParam_3 : DWORD;

cbSize : DWORD;
wCounter : WORD;
func : BYTE;
sMsg : STRING(20);
```



```
(* ...other NOV-/DP-RAM variables *)
END STRUCT
END TYPE
PROGRAM MAIN
VAR
   pNOVRAM
                 : POINTER TO ST_NOVRAM;
    cbNoVRAM : DWORD;
fbGetInfo : FB_GetDPRAMInfo;
state : INT := 0;
    state
                 : BOOL := FALSE;
   bInit
                 : TON;
    timer
                : BOOL;
   bReset
END VAR
CASE state OF
    IF NOT bInit THEN
       state := 1;
    END IF
1:
    fbGetInfo( bExecute := FALSE ); (* reset fb *)
 fbGetInfo( nDevId:= 3,
 bExecute:= TRUE, (* start fb execution *) tTimeOut:= T#10s);
    state := 2;
2:
    fbGetInfo( bExecute := FALSE );
    IF NOT fbGetInfo.bBusy THEN
        IF NOT fbGetInfo.bError THEN
             \  \  \, \hbox{IF fbGetInfo.stInfo.pCardAddress} \, <> \, \hbox{O THEN} \\
                 pNOVRAM := fbGetInfo.stInfo.pCardAddress;
                 cbNOVRAM := fbGetInfo.stInfo.iCardMemSize;
                 bInit := TRUE;
                 state := 0; (* ready, go to the idle step *)
         ELSE
                 state := 100; (* error *)
         END IF
        ELSE
           state := 100; (* error *)
     END IF
    END IF
100:
    ; (* error, stay here *)
END CASE
IF bInit THEN
 (* read from or write to NOV-/DP-RAM*)
 IF bReset THEN (* reset all bytes to 0 *)
     bReset := FALSE;
       MEMSET ( pNOVRAM, 0, cbNOVRAM );
    END IF
    timer( IN := TRUE, PT := T#1s );
    IF timer.Q THEN
        timer( IN := FALSE );
        pNOVRAM^.dwParam 0 := pNOVRAM^.dwParam 0 + 1;
        pNOVRAM^.dwParam_1 := pNOVRAM^.dwParam_1 - 1;
    END IF
END_IF
```

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	PC oder CX (x86)	FCxxxx cards mit NOV- RAM (FCxxxx-0002)	Tc2_loFunctions (IO)



3.8.4 FB_GetDPRAMInfoEx

FB_GetDPRAMInfoEx — nDevId UDINT BOOL bBusy — bExecute BOOL BOOL bError — tTimeOut TIME UDINT nErrid — ST_NovRamAddrInfoEx stInfo

Der Funktionsbaustein FB_GetDPRAMInfoEx ermittelt den Addresspointer und die konfigurierte Größe des NOV/DP-RAM einer Feldbuskarte. Der Addresspointer kann z.B. dazu benutzt werden, um direkt auf das NOV-RAM der Fcxxx-0002 (Beckhoff) Karten oder DPRAM der von TwinCAT nicht unterstützen Karten (Dritthersteller) schreiben oder lesend zugreifen zu können. Die Karte muss vorher im TwinCAT-System als allgemeines NOV/DP-RAM konfiguriert werden.

Der Ausgang stInfo liefert neben der Adresse (pCardAddress) und der Größe (iCardMemSize) des NOV/DP-RAM auch die Art des Zugriffs (eAccessType).

Wenn das NOV/DP-RAM keinen speziellen Zugriff (BYTE oder WORD aligned) erfordert, können im SPS-Programm die MEMCPY-, MEMSET-oder MEMCMP-Funktionen benutzt werden, um auf einen beliebigen Speicheroffset schreibend/lesend zugreifen zu können. Erfordert das NOV/DP-RAM einen speziellen Zugriff (BYTE-oder WORD-aligned), dann darf nur mit passender Datengröße zugegriffen werden. Hierfür sollte der FB NovRamReadWriteEx verwendet werden.

Eingänge

VAR_INPUT
nDevid : UDINT;
bExecute : BOOL;

tTimeOut : TIME := DEFAULT_ADS_TIMEOUT;

END VAR

Name	Тур	Beschreibung
nDevId	UDINT	Die Geräte-Id einer NOV/DPRAM-Karte. Über die Id wird die Karte spezifiziert, deren Informationen gelesen werden sollen. Die Geräte-Ids werden während der Hardware-Konfiguration vom TwinCAT-System festgelegt.
bExecute	BOOL	Bei einer positiven Flanke an diesem Eingang wird der Baustein aktiviert.
tTimeOut	TIME	Gibt die Timeout-Zeit an, die bei der Ausführung des Kommandos/Funktion nicht überschritten werden darf.

Ausgänge

VAR_OUTPUT

bBusy : BOOL; bError : BOOL; nErrId : UDINT;

stInfo : ST NovRamAddrInfoEx;

END_VAR

Name	Тур	Beschreibung
bBusy	BOOL	Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis die Ausführung abgeschlossen wurde.
bError	BOOL	Sollte ein Fehler bei der Ausführung erfolgen, dann wird dieser Ausgang gesetzt.
nErrld	UDINT	Liefert bei einem gesetzten <i>bError-</i> Ausgang die <u>ADS-</u> <u>Fehlernummer [▶ 163]</u> .
stInfo	ST_NovRamAddrInfoEx [• 142]	Eine Struktur mit der Adresse und Größe des NOV/DP-RAM.



Beispiel

In MAIN wird ein Pointer auf einen BYTE deklariert. Dieser Pointer wird beim Programmstart mit der Adresse des NOV-/DP-RAM initialisiert. Durch Referenzieren des Pointers kann auf das NOV-/DP-RAM schreibend oder lesend zugegriffen werden.

```
PROGRAM MAIN
VAR
                  : UDINT := 1; //device 1, see ID of NOV/DP-RAM device
    nDevId
               : POINTER TO BYTE;
    pNOVRAM
    cbNOVRAM : DWORD;
fbGetInfo : FB GetDPRAMInfoEx;
    bInit : BOOL := FALSE;
eAccessType : E_IOACCESSTYPE;
    bInit
    bByteAccess : BOOL;
    bWordAccess : BOOL;
    bDWordAccess : BOOL;
END VAR
IF NOT bInit THEN
    fbGetInfo(
        nDevId
                   := nDevId,
        bExecute := TRUE,
         tTimeOut := T#5S
    IF NOT fbGetInfo.bBusy THEN
        IF NOT fbGetInfo.bError THEN
             pNOVRAM := fbGetInfo.stInfo.pCardAddress;
cbNoVRAM := fbGetInfo.stInfo.iCardMemSize;
             eAccessType := fbGetInfo.stInfo.eAccessType;
             bDWordAccess := FALSE;
             bByteAccess := FALSE;
bWordAccess := FALSE;
             CASE eAccessType OF
                 eTOAccess Default:
                      bDWordAccess := TRUE;
                      //access via MEMCPY, MEMSET, MEMCMP possible
                  eIOAccess Byte:
                     bByteAccess := TRUE;
                      //access via POINTER to BYTE possible
                  eIOAccess WordSwap:
                      bWordAccess := TRUE;
                      //access via POINTER to WORD +
                      //swapping of high and low byte possible
             END CASE
             bInit := TRUE;
        END IF
         fbGetInfo(bExecute := FALSE);
    END IF
END IF
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	\ /	FCxxxx cards mit NOV- RAM (FCxxxx-0002)	Tc2_loFunctions (IO)

3.9 Profibus DPV1 (Sinamics)

3.9.1 F_CreateDpv1ReadReqPkg





Die Funktion F_CreateDpv1ReadReqPkg erzeugt ein DPV1 Telegramm für einen <u>FB Dpv1Read [▶82]</u> eines oder mehrerer Parameter eines Antriebs oder der Controller Unit eines Sinamics Profidrives (Profidrive Specification 3.1). Da Profidrives das Motorola-Format und IPCs das Intel-Format nutzen, führt die Funktion automatisch das Vertauschen der Bytereihenfolge im DPV1 Telegramm von Parametern mit Datentypen mit mehr als einem Byte durch.

Die Funktion liefert die aktuelle Länge des DPV1 Telegramms in Bytes (max. 240 Bytes) zurück.

FUNCTION F_CreateDpv1ReadReqPkg : USINT

Eingänge

```
VAR_INPUT

pDpv1ReqData : POINTER TO ARRAY [1..iMAX_DPV1_SIZE] OF BYTE; (* DPV1 read request *)

iNumOfParams : USINT; (* 1..39; else: reserved *)

iDriveId : USINT;

END VAR
```

Name	Тур	Beschreibung
pDpv1ReqData	ARRAY OF BYTE	Zeiger auf ein Array von 240 Bytes, das das DPV1 Lesetelegramm enthält. Dieses Telegram wird von der Funktion erstellt.
iNumOfParams	USINT	Anzahl der zu lesenden Parameter (1 bis 39). Eine weitere Begrenzung ist die Telegrammgröße von 240 Bytes.
iDriveID	USINT	Die ID ist 1 für die ControllerUnit, 2 für das Antriebsobjekt A, 3 für das Antriebsobjekt B eines Doppel/Dreifach-Antriebs. Die Drive ID wird in der Starter Software gesetzt. 116 ist möglich.

▼/ Ein-/Ausgänge

```
VAR_IN_OUT
    stDpv1Parameter : ARRAY [1..iMAX_DPV1_PARAMS] OF ST_Dpv1ParamAddrEx;(* list of parameters *)
END VAR
```

Name	Тур	Beschreibung
stDpv1Parameter		Array von 39 Parametern, die zum DPV1 Lesetelegramm zugefügt werden sollen.

Voraussetzungen

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	,	Beckhoff FC310x PCI, CX1500-M310, EL6731	Tc2_loFunctions (IO)

3.9.2 F_CreateDpv1WriteReqPkg

Die Funktion F_CreateDpv1WriteReqPkg erzeugt ein DPV1 Telegramm für einen FB_Dpv1Write [▶ 84] eines oder mehrerer Parameter eines Antriebs oder der Controller Unit eines Sinamics Profidrives (Profidrive Specification 3.1). Da Profidrives das Motorola-Format und IPCs das Intel-Format nutzen, führt die Funktion automatisch das Vertauschen der Bytereihenfolge im DPV1 Telegramm von Parametern mit Datentypen mit mehr als einem Byte durch.



Die Funktion liefert die aktuelle Länge des DPV1 Telegramms in Bytes (max. 240 Bytes) zurück.

FUNCTION F_CreateDpv1WriteReqPkg: USINT

Eingänge

```
VAR_INPUT

pDpv1ReqData : POINTER TO ARRAY [1..iMAX_DPV1_SIZE] OF BYTE; (* DPV1 read request *)

iNumOfParams : USINT; (* 1..39; else: reserved *)

iDriveId : USINT;

END_VAR
```

Name	Тур	Beschreibung
pDpv1ReqData	ARRAY OF BYTE	Zeiger auf ein Array von 240 Bytes, das das DPV1 Schreibtelegramm enthält. Dieses Telegram wird von der Funktion erstellt.
iNumOfParams	USINT	Anzahl der zu schreibenden Parameter (1 bis 39). Eine weitere Begrenzung ist die Telegrammgröße von 240 Bytes.
iDriveID	USINT	Die ID ist 1 für die ControllerUnit, 2 für das Antriebsobjekt A, 3 für das Antriebsobjekt B eines Doppel/Dreifach-Antriebs. Die Drive ID wird in der Starter Software gesetzt. 116 ist möglich.

▼/ ■ Ein-/Ausgänge

```
VAR_IN_OUT
stDpv1Parameter : ARRAY [1..iMAX_DPV1_PARAMS] OF ST_Dpv1ParamAddrEx; (* list of parameters *)
stDpv1ValueHeaderEx : ARRAY [1..iMAX_DPV1_PARAMS] OF ST_Dpv1ValueHeaderEx; (* list of values *)
END VAR
```

Name	Тур	Beschreibung
stDpv1Parameter		Array von 39 Parametern, die zum DPV1 Lesetelegramm zugefügt werden sollen.
stDpv1ValueHeader Ex		Array von 39 Parameterwerten, die zum DPV1 Schreibtelegramm zugefügt werden sollen.

Voraussetzungen

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	- ()	Beckhoff FC310x PCI, CX1500-M310, EL6731	Tc2_loFunctions (IO)

3.9.3 F_SplitDpv1ReadResPkg

```
F_SplitDpv1ReadResPkg

—pDpv1ResData POINTER TO ARRAY [1...IMAX_DPV1_SIZE] OF BYTE
—stDpv1Parameter ARRAY [1...IMAX_DPV1_PARAMS] OF ST_Dpv1ParamAddrEx
—stDpv1ValueHeaderEx ARRAY [1...IMAX_DPV1_PARAMS] OF ST_Dpv1ValueHeaderEx
```

Die Funktion F_SplitDpv1ReadResPkg wertet ein DPV1 Telegramm eines <u>FB_Dpv1Read [▶ 82]</u> eines oder mehrerer Parameter eines Antriebs oder der Controller Unit eines Sinamics Profidrives (Profidrive Specification 3.1) aus. Da Profidrives das Motorola-Format und IPCs das Intel-Format nutzen, führt die Funktion automatisch das Vertauschen der Bytereihenfolge im DPV1 Telegramm von Parametern mit Datentypen mit mehr als einem Byte durch.

Die Funktion liefert die aktuelle Länge des DPV1 Telegramms in Bytes (max. 240 Bytes) zurück.



FUNCTION F_SplitDpv1ReadResPkg: USINT

Eingänge

```
VAR_INPUT pDpv1ResData : POINTER TO ARRAY [1..iMAX_DPV1_SIZE] OF BYTE; (* DPV1 read response *) END_VAR
```

Name	Тур	Beschreibung
pDpv1ResData		Zeiger auf ein Array von 240 Bytes, das das DPV1 Leseantworttelegramm enthält. Dieses Telegramm wird von der Funktion ausgewertet.

▼/ ■ Ein-/Ausgänge

```
VAR_IN_OUT
stppv1Parameter : ARRAY [1..iMAX_DPV1_PARAMS] OF ST_Dpv1ParamAddrEx;(* list of parameters *)
stDpv1ValueHeaderEx : ARRAY [1..iMAX_DPV1_PARAMS] OF ST_Dpv1ValueHeaderEx;(* list of values *)
END_VAR
```

Name	Тур	Beschreibung
stDpv1Parameter		Array von 39 Parametern, die zum DPV1 Lesetelegramm zugefügt wurden.
stDpv1ValueHeader Ex	ST_Dpv1ValueHeaderEx [▶ 141]	Array von 39 Parameterwerten, die vom Antrieb gelesen wurden.

Voraussetzungen

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	- ()	Beckhoff FC310x PCI, CX1500-M310, EL6731	Tc2_loFunctions (IO)

3.9.4 F_SplitDpv1WriteResPkg

```
F_SplitDpv1WriteResPkg

pDpv1ResData POINTER TO ARRAY [1...IMAX_DPV1_SIZE] OF BYTE

stDpv1Parameter ARRAY [1...IMAX_DPV1_PARAMS] OF ST_Dpv1ParamAddrEx

stDpv1ValueHeaderEx ARRAY [1...IMAX_DPV1_PARAMS] OF ST_Dpv1ValueHeaderEx
```

Die Funktion F_SplitDpv1WriteResPkg wertet ein DPV1 Telegramm eines <u>FB Dpv1Write [▶84]</u> eines oder mehrerer Parameter eines Antriebs oder der Controller Unit eines Sinamics Profidrives (Profidrive Specification 3.1) aus. Da Profidrives das Motorola-Format und IPCs das Intel-Format nutzen, führt die Funktion automatisch das Vertauschen der Bytereihenfolge im DPV1 Telegramm von Parametern mit Datentypen mit mehr als einem Byte durch.

Die Funktion liefert die aktuelle Länge des DPV1 Telegramms in Bytes (max. 240 Bytes) zurück.

FUNCTION F SplitDpv1WriteResPkg: USINT

Eingänge

VAR_INPUT

pDpv1ResData : POINTER TO ARRAY [1..iMAX_DPV1_SIZE] OF BYTE; (* DPV1 read response *)

END VAR

Name	Тур	Beschreibung
pDpv1ResData		Zeiger auf ein Array von 240 Bytes, das das DPV1 Schreibantworttelegramm enthält. Dieses Telegramm wird von der Funktion ausgewertet.



▼/ Image Image

```
VAR_IN_OUT
stDpv1Parameter : ARRAY [1..iMAX_DPV1_PARAMS] OF ST_Dpv1ParamAddrEx;(* list of parameters *)
stDpv1ValueHeaderEx : ARRAY [1..iMAX_DPV1_PARAMS] OF ST_Dpv1ValueHeaderEx;(* list of values *)
END VAR
```

Name	Тур	Beschreibung
stDpv1Parameter		Array von 39 Parametern, die zum DPV1 Schreibtelegramm zugefügt wurden.
stDpv1ValueHeader Ex		Array von 39 Parameterwerten, die vom Antrieb gelesen wurden.

Voraussetzungen

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	,	Beckhoff FC310x PCI, CX1500-M310, EL6731	Tc2_loFunctions (IO)

3.9.5 FB_Dpv1Read

```
FB_Dpv1Read
bExecute BOOL
                                                                                                                 BOOL bBusy
aNetId T AmsNetId
                                                                                                                BOO/ bError
                                                                                                           BOOL bErrorValues
iProfibusSlaveAdr USINT
iDriveId USIN
                                                                                                                UDINT iErrId
pDpv1ReqData POINTER TO ARRAY [1..iMAX_DPV1_SIZE] OF BYTE
                                                                                                             BYTE iErrorClass
iDpv1ReqDataLen UDINT
                                                                                                             BYTE iErrorCode
pDpv1ResData POINTER TO ARRAY [1..iMAX_DPV1_SIZE] OF BYTE
iDpv1ResDataLen UDINT
tTmOut 77/
iRequestRef USINT
```

Der Funktionsbaustein FB_Dpv1Read liest einen oder mehrere Parameter eines Sinamics Profidrive via DPV1 (Profidrive Specification 3.1). Das DPV1 Lesetelegramm muss mit <u>F_CreateDpv1ReadReqPkg [\rightarrow 78]</u> erstellt werden, bevor an bExecute eine steigende Flanke ansteht. Das DPV1 Antworttelegramm muss mit <u>F_SplitDpv1ReadResPkg [\rightarrow 80]</u> ausgewertet werden, nachdem bBusy eine fallende Flanke anzeigt.

Die Ausführung dieses Funktionsbausteins benötigt einige Zeit, abhängig von der Anzahl der Parameter, die gelesen werden sollen. Der Funktionsbaustein sendet das DPV1 Telegramm und pollt nach einem Antworttelegramm.

Eingänge

```
VAR_INPUT

bExecute : BOOL;
aNetId : T_AmsNetId; (* NetID of Profibus Master FC310x/EL6731 *)
iProfibusSlaveAdr : USINT; (* DP address of ProfiDrive *)
iDriveId : USINT; (* 1..16 possible *)
pDpv1ReqData : POINTER TO ARRAY [1..iMAX_DPV1_SIZE] OF BYTE;
iDpv1ReqDataLen : UDINT;
pDpv1ResData : POINTER TO ARRAY [1..iMAX_DPV1_SIZE] OF BYTE;
iDpv1ResDataLen : UDINT;
tTmOut : TIME := DEFAULT_ADS_TIMEOUT;
END VAR
```

Name	Тур	Beschreibung
bExecute	BOOL	Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.
aNetId	T_AmsNetID	Die Netzwerkadresse des Profibus Master Gerätes (siehe ADS-Tab des Profibus Master Gerätes in der I/O-Konfiguration im TwinCAT-System)



Name	Тур	Beschreibung
iProfibusSlaveAdr	USINT	Die Profibus Slave DP-Adresse des Antriebs. Das ist eine Adresse für mehrere Achsen, spezifiziert im TwinCAT-System in der I/O-Konfiguration.
iDriveID	USINT	Die ID ist 1 für die ControllerUnit, 2 für das Antriebsobjekt A, 3 für das Antriebsobjekt B eines Doppel/Dreifach-Antriebs. Die Drive ID wird in der Starter Software gesetzt. 116 ist möglich.
pDpv1ReqData	POINTER TO ARRAY	Zeiger auf ein Array von 240 Bytes, das das DPV1 Lesetelegramm enthält. Dieses Telegramm muss von der Funktion F CreateDpv1ReadReqPkg [> 78] erstellt werden, bevor das DPV1 Lesen via bExecute aktiviert wird.
iDv1ReqDataLen	UDINT	Maximale Länge des DPV1 Datapuffer (240 bytes).
pDpv1ResData	POINTER TO ARRAY	Zeiger auf ein Array von 240 Bytes, das das DPV1 Leseantworttelegramm enthält. Dieses Telegramm muss von der Funktion F SplitDpv1ReadResPkg [> 80] ausgewertet werden nachdem auf bBusy eine negative Flanke erscheint.
iDpv1ResDataLen	UDINT	Maximale Länge des DPV1 Antwort-Datapuffers (240 bytes).
tTmOut	TIME	Maximale Zeit die bei der Ausführung des Kommandos nicht überschritten werden soll.

₹/ Ein-/Ausgänge

```
VAR_OUTPUT
   __iRequestRef : USINT; (* 1..127; 0: reserved *)
END_VAR
```

Name	Тур	Beschreibung
iRefRequest	USINT	Referenz, die mit jedem Telegramm automatisch hochgezählt wird. Die Referenz wird für die Zuordnung der Antworten auf die Schreib/Lese-Anforderungen benötigt.

Ausgänge

```
VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
bErrorValues : BOOL;
iErrId : UDINT;
iErrorClass : BYTE;
iErrorCode : BYTE;
END VAR
```

Name	Тур	Beschreibung
bBusy	BOOL	Der Ausgang geht auf TRUE sobald der Funktionsbaustein via bExecute aktiviert wurde und bleibt so lange TRUE, wie der Baustein keine Antwort erhalten hat.
bError	BOOL	Bei Fehlern geht der Ausgang auf TRUE und bBusy auf FALSE.
bErrorValues	BOOL	Ist TRUE wenn der DPV1 Read nicht oder nur teilweise erfolgreich war. Die Fehlerursachen werden über die Fehler-ID geliefert (sowie Class und Code).
iErrld	UDINT	Liefert die <u>ADS Fehlernummer</u> [▶ 163] oder funktionsbausteinspezifische Fehlernummern, wenn bError = TRUE.
iErrClass	BYTE	Profidrive Fehlerklasse
iErrCode	BYTE	Profidrive Fehlercode



Funktionbausteinspezifische Fehlercodes	Beschreibung
0x2	Falsche Antwortreferenz
0x3	DPV1 Read fehlerhaft oder teilweise fehlerhaft
0x4	Falsche Antwort-ID
other error IDs	Siehe ADS Fehlercodes

Fehlerklassen	Beschreibung	Fehlercode
0x0 - 0x9	Reserviert	-
0xA	Anwendungsfehler	0x0: Lesefehler 0x1: Schreibfehler 0x2: Modulfehler 0x3 - 0x7: reserviert 0x8: Versionskonflikt 0x9: nicht unterstützt 0xA - 0xF: benutzerabhängig
0xB	Zugriffsfehler	0x0: ungültiger Index (kein Datenblock DB47, Parameterzugriff wird nicht unterstützt) 0x1: Schreiblängenfehler 0x2: ungültiger Slot 0x3: Typkonflikt 0x4: ungültiger Bereich 0x5: Zustandskonflikt (Zugriff auf DB47 temporär nicht möglich wegen interner Prozesszustände) 0x6: Zugriff verweigert 0x7: ungültiger Bereich (Schreibfehler im DB47 Header) 0x8: ungültiger Parameter 0x9: ungültiger Typ 0xA - 0xF: benutzerabhängig
0xC	Resourcefehler	0x0: Lesekonflikt 0x1: Schreibkonflikt 0x2: Resource beschäftigt 0x3: Resource nicht erreichbar 0x4 - 0x7: reserviert 0x8 - 0xF: benutzerabhängig
0xD - 0xF	Benutzerdefinierte Fehler	-

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	,	Beckhoff FC310x PCI, CX1500-M310, EL6731	Tc2_loFunctions (IO)

3.9.6 FB_Dpv1Write

```
FB_Dpv1Write
bExecute BOOL
aNetId T_AmsNetId
                                                                                                                    BOOL bBusy
                                                                                                                    BOOL bError
iProfibusSlaveAdr USINT
                                                                                                               BOOL bErrorValues
iDriveId USIN7
                                                                                                                    UDINT iErrId
pDpv1ReqData POINTER TO ARRAY [1..iMAX_DPV1_SIZE] OF BYTE
                                                                                                                 BYTE iErrorClass
iDpv1ReqDataLen UDINT
                                                                                                                 BYTE iErrorCode
pDpv1ResData POINTER TO ARRAY [1..iMAX_DPV1_SIZE] OF BYTE
iDpv1ResDataLen UDINT
tTmOut 77/
iRequestRef USINT
```

Der Funktionsbaustein FB_Dpv1Write schreibt einen oder mehrere Parameter eines Sinamics Profidrive via DPV1 (Profidrive Specification 3.1). Das DPV1 Schreibtelegramm muss mit <u>F_CreateDpv1WriteReqPkg_[▶_79]</u> erstellt werden, bevor an bExecute eine steigende Flanke ansteht. Das DPV1 Antworttelegramm muss mit <u>F_SplitDpv1WriteResPkg_[▶_81]</u> ausgewertet werden, nach dem bBusy eine fallende Flanke anzeigt.



Die Ausführung dieses Funktionsbausteins dauert einige Zeit, abhängig von der Anzahl der Parameter, die gelesen werden sollen. Der Funktionsbaustein sendet das DPV1 Telegramm und pollt nach einem Antworttelegramm.

Eingänge

Name	Тур	Beschreibung
bExecute	BOOL	Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.
aNetId	T_AmsNetID	Die Netzwerkadresse des Profibus Master Gerätes (siehe ADS-Tab des Profibus Master Gerätes in der I/O-Konfiguration im TwinCAT-System)
iProfibusSlaveAdr	USINT	Die Profibus Slave DP-Adresse des Antriebs. Das ist eine Adresse für mehrere Achsen, spezifiziert im TwinCAT-System in der I/O-Konfiguration.
iDriveID	USINT	Die ID ist 1 für die ControllerUnit, 2 für das Antriebsobjekt A, 3 für das Antriebsobjekt B eines Doppel/Dreifach-Antriebs. Die Drive ID wird in der Starter Software gesetzt. 116 ist möglich.
pDpv1ReqData	POINTER TO ARRAY	Zeiger auf ein Array von 240 Bytes, das das DPV1 Schreibtelegramm enthält. Dieses Telegramm muss von der Funktion <u>F CreateDpv1WriteReqPkg</u> [▶ 79] erstellt werden, bevor das DPV1 Lesen via bExecute aktiviert wird.
iDv1ReqDataLen	UDINT	Maximale Länge des DPV1 Datapuffer (240 bytes).
pDpv1ResData	POINTER TO ARRAY	Zeiger auf ein Array von 240 Bytes, das das DPV1 Schreibantworttelegramm enthält. Dieses Telegramm muss von der Funktion <u>F SplitDpv1WriteResPkg</u> [▶ 81] ausgewertet werden nachdem auf bBusy eine negative Flanke erscheint.
iDpv1ResDataLen	UDINT	Maximale Länge des DPV1 Antwort-Datapuffers (240 bytes).
tTmOut	TIME	Maximale Zeit die bei der Ausführung des Kommandos nicht überschritten werden soll.

▼/ ■ Ein-/Ausgänge

```
VAR_OUTPUT
iRequestRef : USINT; (* 1..127; 0: reserved *)
END_VAR
```

Name	Тур	Beschreibung
iRefRequest	USINT	Referenz, die mit jedem Telegramm automatisch hochgezählt wird. Die Referenz wird für die Zuordnung der
		Antworten auf die Schreib/Lese-Anforderungen benötigt.

Ausgänge

VAR OUTPUT	
_ bBusy	: BOOL;
bError	: BOOL;



bErrorValues : BOOL;
iErrId : UDINT;
iErrorClass : BYTE;
iErrorCode : BYTE;
END_VAR

Name	Тур	Beschreibung
bBusy	BOOL	Der Ausgang geht auf TRUE sobald der Funktionsbaustein via bExecute aktiviert wurde und bleibt so lange TRUE, wie der Baustein keine Antwort erhalten hat.
bError	BOOL	Bei Fehlern geht der Ausgang auf TRUE und bBusy auf FALSE.
bErrorValues	BOOL	Ist TRUE wenn der DPV1 Read nicht oder nur teilweise erfolgreich war. Die Fehlerursachen werden über die Fehler-ID geliefert (sowie Class und Code).
iErrld	UDINT	Liefert die <u>ADS Fehlernummer</u> [▶ 163] oder funktionsbausteinspezifische Fehlernummern, wenn bError = TRUE.
iErrClass	BYTE	Profidrive Fehlerklasse
iErrCode	BYTE	Profidrive Fehlercode

Funktionbausteinspezifische Fehlercodes	Beschreibung
0x2	Falsche Antwortreferenz
0x3	DPV1 Read fehlerhaft oder teilweise fehlerhaft
0x4	Falsche Antwort-ID
other error IDs	Siehe ADS Fehlercodes

Fehlerklassen	Beschreibung	Fehlercode
0x0 - 0x9	Reserviert	-
0xA	Anwendungsfehler	0x0: Lesefehler 0x1: Schreibfehler 0x2: Modulfehler 0x3 - 0x7: reserviert 0x8: Versionskonflikt 0x9: nicht unterstützt 0xA - 0xF: benutzerabhängig
0xB	Zugriffsfehler	0x0: ungültiger Index (kein Datenblock DB47, Parameterzugriff wird nicht unterstützt) 0x1: Schreiblängenfehler 0x2: ungültiger Slot 0x3: Typkonflikt 0x4: ungültiger Bereich 0x5: Zustandskonflikt (Zugriff auf DB47 temporär nicht möglich wegen interner Prozesszustände) 0x6: Zugriff verweigert 0x7: ungültiger Bereich (Schreibfehler im DB47 Header) 0x8: ungültiger Parameter 0x9: ungültiger Typ 0xA - 0xF: benutzerabhängig
0xC	Resourcefehler	0x0: Lesekonflikt 0x1: Schreibkonflikt 0x2: Resource beschäftigt 0x3: Resource nicht erreichbar 0x4 - 0x7: reserviert 0x8 - 0xF: benutzerabhängig
0xD - 0xF	Benutzerdefinierte Fehler	-



Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	PC oder CX (x86)	Beckhoff FC310x PCI, CX1500-M310, EL6731	Tc2_loFunctions (IO)

3.10 Profinet DPV1 (Sinamics)

3.10.1 F_CreateDpv1ReadReqPkgPNET

```
F_CreateDpv1ReadReqPkgPNET

— pDpv1ReqData POINTER TO ARRAY [1..IMAX_DPV1_SIZE_PNET_REQ] OF BYTE

— iNumOfParams USINT

— iDriveId USINT
— stDpv1Parameter ARRAY [1..IMAX_DPV1_PARAMS] OF ST_Dpv1ParamAddrEx
```

Die Funktion F_CreateDpv1ReadReqPkg erzeugt ein DPV1 Telegramm für einen FB_Dpv1ReadPNET [▶ 90] eines oder mehrerer Parameter eines Antriebs oder der Controller Unit eines Sinamics Profidrives (Profidrive Specification 3.1), das über Profinet angeschlossen ist. Da Profidrives das Motorola-Format und IPCs das Intel-Format nutzen, führt die Funktion automatisch das Vertauschen der Bytereihenfolge im DPV1 Telegramm von Parametern mit Datentypen mit mehr als einem Byte durch.

Die Funktion liefert die aktuelle Länge des DPV1 Telegramms in Bytes (max. 240 Bytes) zurück.

FUNCTION F_CreateDpv1ReadReqPkgPNET: USINT

Eingänge

```
VAR_INPUT
    pDpv1ReqData : POINTER TO ARRAY [1..iMAX_DPV1_SIZE] OF BYTE;(* DPV1 read request *)
    iNumOfParams : USINT; (* 1..39; else: reserved *)
    iDriveId : USINT;
END_VAR
```

Name	Тур	Beschreibung
pDpv1ReqData	POINTER TO ARRAY	Zeiger auf ein Array von 240 Bytes, das das DPV1 Lesetelegramm enthält. Dieses Telegramm muss von der Funktion <u>F CreateDpv1ReadReqPkg</u> [▶ 78] erstellt werden,
		bevor das DPV1 Lesen via bExecute aktiviert wird.
iNumOfParams	USINT	Anzahl der zu lesenden Parameter (1 to 39). Eine weitere Begrenzung ist die Telegrammgröße von 240 Bytes.
iDriveID	USINT	Die ID ist 1 für die ControllerUnit, 2 für das Antriebsobjekt A, 3 für das Antriebsobjekt B eines Doppel/Dreifach-Antriebs. Die Drive ID wird in der Starter Software gesetzt. 116 ist möglich.

🕏 🕒 Ein-/Ausgänge

```
VAR_IN_OUT stDpv1Parameter : ARRAY [1..iMAX_DPV1_PARAMS] OF ST_Dpv1ParamAddrEx;(* list of parameters *) END VAR
```

Name	Тур	Beschreibung
stDpv1Parameter	<u> </u>	Array von 39 Parametern, die zum DPV1 Lesetelegramm zugefügt werden sollen.



Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	PC oder CX (x86)	Beckhoff EL6632	Tc2 IoFunctions (IO)

3.10.2 F_CreateDpv1WriteReqPkgPNET

Die Funktion F_CreateDpv1WriteReqPkgPNET erzeugt ein DPV1 Telegramm für einen FB_Dpv1WritePNET [▶ 93] eines oder mehrerer Parameter eines Antriebs oder der Controller Unit eines Sinamics Profidrives (Profidrive Specification 3.1), das über Profinet angeschlossen ist. Da Profidrives das Motorola-Format und IPCs das Intel-Format nutzen, führt die Funktion automatisch das Vertauschen der Bytereihenfolge im DPV1 Telegramm von Parametern mit Datentypen mit mehr als einem Byte durch.

Die Funktion liefert die aktuelle Länge des DPV1 Telegramms in Bytes (max. 240 Bytes) zurück.

FUNCTION F_CreateDpv1WriteReqPkgPNET: USINT

Eingänge

Name	Тур	Beschreibung
pDpv1ReqData	ARRAY OF BYTE	Zeiger auf ein Array von 240 Bytes, das das DPV1 Schreibtelegramm enthält. Dieses Telegram wird von der Funktion erstellt.
iNumOfParams	USINT	Anzahl der zu schreibenden Parameter (1 bis 39). Eine weitere Begrenzung ist die Telegrammgröße von 240 Bytes.
iDriveID	USINT	Die ID ist 1 für die ControllerUnit, 2 für das Antriebsobjekt A, 3 für das Antriebsobjekt B eines Doppel/Dreifach-Antriebs. Die Drive ID wird in der Starter Software gesetzt. 116 ist möglich.

▼/ ■ Ein-/Ausgänge

```
VAR_IN_OUT
stDpv1Parameter : ARRAY [1..iMAX_DPV1_PARAMS] OF ST_Dpv1ParamAddrEx;(* list of parameters *)
stDpv1ValueHeaderEx : ARRAY [1..iMAX_DPV1_PARAMS] OF ST_Dpv1ValueHeaderEx;(* list of values *)
END_VAR
```

Name	Тур	Beschreibung
stDpv1Parameter		Array von 39 Parametern, die zum DPV1Schreibtelegramm zugefügt werden sollen.
stDpv1ValueHeader Ex		Array von 39 Parameterwerten, die zum DPV1 Schreibtelegramm zugefügt werden sollen.



Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	PC oder CX (x86)	Beckhoff EL6632	Tc2_loFunctions (IO)

3.10.3 F_SplitDpv1ReadResPkgPNET

	F_SplitDpv1ReadResPkgPNET	
_	pDpv1ResData POINTER TO ARRAY [1IMAX_DPV1_SIZE_PNET_RES] OF BYTE	USINT F_SplitDpv1ReadResPkgPNET —
_	stDpv1Parameter ARRAY [1iMAX_DPV1_PARAMS] OF ST_Dpv1ParamAddrEx	
_	stDpv1ValueHeaderEx ARRAY [1iMAX_DPV1_PARAMS] OF ST_Dpv1ValueHeaderEx	

Die Funktion F_SplitDpv1ReadResPkgPNET wertet ein DPV1 Telegramm eines <u>FB_Dpv1ReadPNET [▶ 90]</u> eines oder mehrerer Parameter eines Antriebs oder der Controller Unit eines Sinamics Profidrives (Profidrive Specification 3.1) aus, das über Profinet angeschlossen ist. Da Profidrives das Motorola-Format und IPCs das Intel-Format nutzen, führt die Funktion automatisch das Vertauschen der Bytereihenfolge im DPV1 Telegramm von Parametern mit Datentypen mit mehr als einem Byte durch.

Die Funktion liefert die aktuelle Länge des DPV1 Telegramms in Bytes (max. 240 Bytes) zurück.

FUNCTION F_SplitDpv1ReadResPkgPNET: USINT

Eingänge

```
VAR_INPUT
    pDpv1ResData : POINTER TO ARRAY [1..iMAX_DPV1_SIZE] OF BYTE; (* DPV1 read response *)
END VAR
```

Name	Тур	Beschreibung
pDpv1ResData		Zeiger auf ein Array von 240 Bytes, das das DPV1 Leseantworttelegramm enthält. Dieses Telegramm muss von der Funktion ausgewertet werden.

▼/ ■ Ein-/Ausgänge

```
VAR_IN_OUT
    stpv1Parameter : ARRAY [1..iMAX_DPV1_PARAMS] OF ST_Dpv1ParamAddrEx;(* list of parameters *)
    stDpv1ValueHeaderEx : ARRAY [1..iMAX_DPV1_PARAMS] OF ST_Dpv1ValueHeaderEx;(* list of values *)
END VAR
```

Name	Тур	Beschreibung
stDpv1Parameter		Array von 39 Parametern, die zum DPV1 Lesetelegramm zugefügt werden sollen.
stDpv1ValueHeader Ex		Array von 39 Parameterwerten, die vom Antrieb gelesen werden sollen.

Voraussetzungen

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	PC oder CX (x86)	Beckhoff EL6632	Tc2 IoFunctions (IO)

3.10.4 F_SplitDpv1WriteResPkgPNET

	F_SplitDpv1WriteResPkgPNET	
-	pDpv1ResData POINTER TO ARRAY [1IMAX_DPV1_SIZE_PNET_RES] OF BYTE	USINT F_SplitDpv1WriteResPkgPNET —
-	stDpv1Parameter ARRAY [1iMAX_DPV1_PARAMS] OF ST_Dpv1ParamAddrEx	
\dashv	stDpv1ValueHeaderEx ARRAY [1iMAX_DPV1_PARAMS] OF ST_Dpv1ValueHeaderEx	



Die Funktion F_SplitDpv1WriteResPkgPNET wertet ein DPV1 Telegramm eines <u>FB Dpv1WritePNET [▶93]</u> eines oder mehrerer Parameter eines Antriebs oder der Controller Unit eines Sinamics Profidrives (Profidrive Specification 3.1) aus, das über Profinet angeschlossen ist. Da Profidrives das Motorola-Format und IPCs das Intel-Format nutzen, führt die Funktion automatisch das Vertauschen der Bytereihenfolge im DPV1 Telegramm von Parametern mit Datentypen mit mehr als einem Byte durch.

Die Funktion liefert die aktuelle Länge des DPV1 Telegramms in Bytes (max. 240 Bytes) zurück.

FUNCTION F_SplitDpv1WriteResPkgPNET: USINT

Eingänge

```
VAR_INPUT pDpv1ResData : POINTER TO ARRAY [1..iMAX_DPV1_SIZE] OF BYTE; (* DPV1 read response *) END_VAR
```

Name	Тур	Beschreibung
pDpv1ResData		Zeiger auf ein Array von 240 Bytes, das das DPV1 Schreibantworttelegramm enthält. Dieses Telegramm wird von der Funktion ausgewertet.

▼/ ■ Ein-/Ausgänge

```
VAR_IN_OUT
    stDpv1Parameter : ARRAY [1..iMAX_DPV1_PARAMS] OF ST_Dpv1ParamAddrEx;(* list of parameters *)
    stDpv1ValueHeaderEx : ARRAY [1..iMAX_DPV1_PARAMS] OF ST_Dpv1ValueHeaderEx;(* list of values *)
END_VAR
```

Name	Тур	Beschreibung
stDpv1Parameter		Array von 39 Parametern, die zum DPV1 Schreibtelegramm zugefügt werden sollen.
stDpv1ValueHeader Ex		Array von 39 Parameterwerten, die vom Antrieb gelesen werden sollen.

Voraussetzungen

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	PC oder CX (x86)	Beckhoff EL6632	Tc2 IoFunctions (IO)

3.10.5 FB_Dpv1ReadPNET

```
FB_Dpv1ReadPNET
bExecute BOOL
                                                                                                                                     BOOL bBusy
aNetId T AmsNetId
                                                                                                                                     BOO/ bFrror
iProfinetPort UINT
                                                                                                                               BOOL bErrorValues
iDriveId US
                                                                                                                                     UDINT iErrId
pDpv1ReqData POINTER TO ARRAY [1..IMAX_DPV1_SIZE_PNET_REQ] OF BYTE
                                                                                                                                  BYTE iErrorClass
iDpv1ReqDataLen UDINT
                                                                                                                                 BYTE iErrorCode
pDpv1ResData POINTER TO ARRAY [1..IMAX_DPV1_SIZE_PNET_RES] OF BYTE
iDpv1ResDataLen UDIN7
tTmOut
iRequestRef USINT
```

Der Funktionsbaustein FB_Dpv1ReadPNET liest einen oder mehrere Parameter eines Sinamics Profidrive via DPV1 (Profidrive specification 3.1) via Profinet. Das DPV1 Lesetelegramm muss mit <u>F_CreateDpv1ReadReqPkgPNET [▶ 87]</u> erstellt werden, bevor an bExecute eine steigende Flanke ansteht. Das DPV1 Antworttelegramm muss mit <u>F_SplitDpv1ReadResPkgPNET [▶ 89]</u> ausgewertet werden, nach dem bBusy eine fallende Flanke anzeigt.



Die Ausführung dieses Funktionsbausteins benötigt einige Zeit, abhängig von der Anzahl der Parameter, die gelesen werden sollen. Der Funktionsbaustein sendet das DPV1 Telegramm und pollt nach einem Antworttelegramm.

Eingänge

```
VAR_INPUT

bExecute : BOOL; (* drive access info *)

aNetId : T_AmsNetId; (* NetID of Profibus Master EL6631 *)

iProfinetPort : UINT; (* Port of ProfiDrive *)

iDriveId : USINT; (* 0..255 possible *)

pDpv1ReqData : POINTER TO ARRAY [1..iMAX_DPV1_SIZE_PNET_REQ] OF BYTE;

iDpv1ReqDataLen : UDINT;

pDpv1ResData : POINTER TO ARRAY [1..iMAX_DPV1_SIZE_PNET_RES] OF BYTE;

iDpv1ResDataLen : UDINT;

tTmOut : TIME;

END_VAR
```

Name	Тур	Beschreibung
bExecute	BOOL	Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.
aNetId	T_AmsNetID	Die Netzwerkadresse des Profibus Master Gerätes (siehe ADS-Tab des Profibus Master Gerätes in der I/O-Konfiguration im TwinCAT System)
iProfinetPort	UINT	Die Profinet Port-Nummer des Antriebs. Das ist eine Adresse für mehrere Achsen, spezifiziert im TwinCAT-System in der I/O-Konfiguration.
iDriveID	USINT	Die ID ist 1 für die ControllerUnit, 2 für das Antriebsobjekt A, 3 für das Antriebsobjekt B eines Doppel/Dreifach-Antriebs. Die Drive ID wird in der Starter Software gesetzt. 116 ist möglich.
pDpv1ReqData	POINTER TO ARRAY	Zeiger auf ein Array von 240 Bytes, das das DPV1 Lesetelegramm enthält. Dieses Telegramm muss von der Funktion F CreateDpv1ReadReqPkg [** 78] erstellt werden, bevor das DPV1 Lesen via bExecute aktiviert wird.
iDv1ReqDataLen	UDINT	Maximale Länge des DPV1 Datapuffer (240 bytes).
pDpv1ResData	POINTER TO ARRAY	Zeiger auf ein Array von 240 Bytes, das das DPV1 Leseantworttelegramm enthält. Dieses Telegramm muss von der Funktion <u>F SplitDpv1ReadResPkg [** 80]</u> ausgewertet werden nachdem auf bBusy eine negative Flanke erscheint.
iDpv1ResDataLen	UDINT	Maximale Länge des DPV1 Antwort-Datapuffers (240 bytes).
tTmOut	TIME	Maximale Zeit die bei der Ausführung des Kommandos nicht überschritten werden soll.

▼/ ► Ein-/Ausgänge

```
VAR_OUTPUT
iRequestRef : USINT; (* 1..127; 0: reserved *)
END_VAR
```

Name	Тур	Beschreibung
iRefRequest	USINT	Referenz, die mit jedem Telegramm automatisch hochgezählt wird. Die Referenz wird für die Zuordnung der Antworten auf die Schreib/Lese-Anforderungen benötigt.

Ausgänge

VAR OUTPUT		
bBusy	:	BOOL;
bError	:	BOOL;
bErrorValues	:	BOOL;



iErrId : UDINT;
iErrorClass : BYTE;
iErrorCode : BYTE;
END_VAR

Name	Тур	Beschreibung
bBusy	BOOL	Der Ausgang geht auf TRUE sobald der Funktionsbaustein via bExecute aktiviert wurde und bleibt so lange TRUE, wie der Baustein keine Antwort erhalten hat.
bError	BOOL	Bei Fehlern geht der Ausgang auf TRUE und bBusy auf FALSE.
bErrorValues	BOOL	Ist TRUE wenn der DPV1 Read nicht oder nur teilweise erfolgreich war. Die Fehlerursachen werden über die Fehler-ID geliefert (sowie Class und Code).
iErrld	UDINT	Liefert die <u>ADS Fehlernummer</u> [▶ 163] oder funktionsbausteinspezifische Fehlernummern, wenn bError = TRUE.
iErrClass	BYTE	Profidrive Fehlerklasse
iErrCode	BYTE	Profidrive Fehlercode

Funktionbausteinspezifische Fehlercodes	Beschreibung
0x2	Falsche Antwortreferenz
0x3	DPV1 Read fehlerhaft oder teilweise fehlerhaft
0x4	Falsche Antwort-ID
other error IDs	Siehe ADS Fehlercodes

Fehlerklassen	Beschreibung	Fehlercode
0x0 - 0x9	Reserviert	-
0xA	Anwendungsfehler	0x0: Lesefehler 0x1: Schreibfehler 0x2: Modulfehler 0x3 - 0x7: reserviert 0x8: Versionskonflikt 0x9: nicht unterstützt 0xA - 0xF: benutzerabhängig
0xB	Zugriffsfehler	0x0: ungültiger Index (kein Datenblock DB47, Parameterzugriff wird nicht unterstützt) 0x1: Schreiblängenfehler 0x2: ungültiger Slot 0x3: Typkonflikt 0x4: ungültiger Bereich 0x5: Zustandskonflikt (Zugriff auf DB47 temporär nicht möglich wegen interner Prozesszustände) 0x6: Zugriff verweigert 0x7: ungültiger Bereich (Schreibfehler im DB47 Header) 0x8: ungültiger Parameter 0x9: ungültiger Typ 0xA - 0xF: benutzerabhängig
0xC	Resourcefehler	0x0: Lesekonflikt 0x1: Schreibkonflikt 0x2: Resource beschäftigt 0x3: Resource nicht erreichbar 0x4 - 0x7: reserviert 0x8 - 0xF: benutzerabhängig
0xD - 0xF	Benutzerdefinierte Fehler	-



Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	PC oder CX (x86)	Beckhoff EL6632	Tc2 IoFunctions (IO)

3.10.6 FB_Dpv1WritePNET

```
FB_Dpv1WritePNET

bExecute BOOL
anetId T_AmsNetId
iProfinetPort UINT
iDriveId USINT
pDpv1ReqData POINTER TO ARRAY [1..iMAX_DPV1_SIZE_PNET_REQ] OF BYTE
iDpv1ReqData POINTER TO ARRAY [1..iMAX_DPV1_SIZE_PNET_RES] OF BYTE
iDpv1ResData POINTER TO ARRAY [1..iMAX_DPV1_SIZE_PNET_RES] OF BYTE
iDpv1ResDatalen UDINT
pDpv1ResDatalen UDINT
tTmOut TIME
iRequestRef USINT
```

Der Funktionsbaustein FB_Dpv1WritePNET schreibt einen oder mehrere Parameter eines Sinamics Profidrive via DPV1 (Profidrive Specification 3.1) via Profinet. Das DPV1 Schreibtelegramm muss mit <u>F_CreateDpv1WriteReqPkgPNET [\rightarrow 88]</u> erstellt werden, bevor an bExecute eine steigende Flanke ansteht. Das DPV1 Antworttelegramm muss mit <u>F_SplitDpv1WriteResPkgPNET [\rightarrow 89]</u> ausgewertet werden, nachdem bBusy eine fallende Flanke anzeigt.

Die Ausführung dieses Funktionsbausteins benötigt einige Zeit, abhängig von der Anzahl der Parameter, die gelesen werden sollen. Der Funktionsbaustein sendet das DPV1 Telegramm und pollt nach einem Antworttelegramm.

Eingänge

```
VAR_INPUT

bExecute : BOOL; (* drive access info *)
aNetId : T_AmsNetId; (* NetID of Profibus Master EL6631 *)
iProfinetPort : UINT; (* Port of ProfiDrive *)
iDriveId : USINT; (* 0..255 possible *)
pDpv1ReqData : POINTER TO ARRAY [1..iMAX_DPV1_SIZE_PNET_REQ] OF BYTE;
iDpv1ReqDatalen : UDINT;
pDpv1ResData : POINTER TO ARRAY [1..iMAX_DPV1_SIZE_PNET_RES] OF BYTE;
iDpv1ResDatalen : UDINT;
tTmOut : TIME;
END VAR
```

Name	Тур	Beschreibung
bExecute	BOOL	Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.
aNetId	T_AmsNetID	Die Netzwerkadresse des Profibus Master Gerätes (siehe ADS-Tab des Profibus Master Gerätes in der I/O-Konfiguration im TwinCAT System)
iProfinetPort	UINT	Die Profinet Port-Nummer des Antriebs. Das ist eine Adresse für mehrere Achsen, spezifiziert im TwinCAT-System in der I/O-Konfiguration.
iDriveID	USINT	Die ID ist 1 für die ControllerUnit, 2 für das Antriebsobjekt A, 3 für das Antriebsobjekt B eines Doppel/Dreifach-Antriebs. Die Drive ID wird in der Starter Software gesetzt. 116 ist möglich.
pDpv1ReqData	POINTER TO ARRAY	Zeiger auf ein Array von 240 Bytes, das das DPV1 Schreibtelegramm enthält. Dieses Telegramm muss von der Funktion <u>F CreateDpv1WriteReqPkg [* 79]</u> erstellt werden, bevor das DPV1 Lesen via bExecute aktiviert wird.
iDv1ReqDataLen	UDINT	Maximale Länge des DPV1 Datapuffer (240 bytes).



Name	Тур	Beschreibung
pDpv1ResData	POINTER TO ARRAY	Zeiger auf ein Array von 240 Bytes, das das DPV1 Schreibantworttelegramm enthält. Dieses Telegramm muss von der Funktion F SplitDpv1WriteResPkg [> 81] ausgewertet werden nachdem auf bBusy eine negative Flanke erscheint.
iDpv1ResDataLen	UDINT	Maximale Länge des DPV1 Antwort-Datapuffers (240 bytes).
tTmOut	TIME	Maximale Zeit die bei der Ausführung des Kommandos nicht überschritten werden soll.

VAR_OUTPUT iRequestRef : USINT; (* 1..127; 0: reserved *) END_VAR

Name	Тур	Beschreibung
iRefRequest	USINT	Referenz, die mit jedem Telegramm automatisch hochgezählt wird. Die Referenz wird für die Zuordnung der Antworten auf die Schreib/Lese-Anforderungen benötigt.

Ausgänge

VAR_OUTPUT
bBusy : BOOL;
bError : BOOL;
bErrorValues : BOOL;
iErrId : UDINT;
iErrorClass : BYTE;
iErrorCode : BYTE;

Name	Тур	Beschreibung
bBusy	BOOL	Der Ausgang geht auf TRUE sobald der Funktionsbaustein via bExecute aktiviert wurde und bleibt so lange TRUE, wie der Baustein keine Antwort erhalten hat.
bError	BOOL	Bei Fehlern geht der Ausgang auf TRUE und bBusy auf FALSE.
bErrorValues	BOOL	Ist TRUE wenn der DPV1 Write nicht oder nur teilweise erfolgreich war. Die Fehlerursachen werden über die Fehler-ID geliefert (sowie Class und Code).
iErrld	UDINT	Liefert die <u>ADS Fehlernummer [▶ 163]</u> oder funktionsbausteinspezifische Fehlernummern, wenn bError = TRUE.
iErrClass	BYTE	Profidrive Fehlerklasse
iErrCode	BYTE	Profidrive Fehlercode

Funktionbausteinspezifische Fehlercodes	Beschreibung
0x2	Falsche Antwortreferenz
0x3	DPV1 Read fehlerhaft oder teilweise fehlerhaft
0x4	Falsche Antwort-ID
other error IDs	Siehe ADS Fehlercodes

Fehlerklassen	Beschreibung	Fehlercode
0x0 - 0x9	Reserviert	-
0xA	Anwendungsfehler	0x0: Lesefehler 0x1: Schreibfehler 0x2: Modulfehler 0x3 - 0x7: reserviert



Fehlerklassen	Beschreibung	Fehlercode
		0x8: Versionskonflikt 0x9: nicht unterstützt 0xA - 0xF: benutzerabhängig
0xB	Zugriffsfehler	0x0: ungültiger Index (kein Datenblock DB47, Parameterzugriff wird nicht unterstützt) 0x1: Schreiblängenfehler 0x2: ungültiger Slot 0x3: Typkonflikt 0x4: ungültiger Bereich 0x5: Zustandskonflikt (Zugriff auf DB47 temporär nicht möglich wegen interner Prozesszustände) 0x6: Zugriff verweigert 0x7: ungültiger Bereich (Schreibfehler im DB47 Header) 0x8: ungültiger Parameter 0x9: ungültiger Typ 0xA - 0xF: benutzerabhängig
0xC	Resourcefehler	0x0: Lesekonflikt 0x1: Schreibkonflikt 0x2: Resource beschäftigt 0x3: Resource nicht erreichbar 0x4 - 0x7: reserviert 0x8 - 0xF: benutzerabhängig
0xD - 0xF	Benutzerdefinierte Fehler	-

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	PC oder CX (x86)	Beckhoff EL6632	Tc2_loFunctions (IO)

3.11 RAID Controller

3.11.1 FB_RAIDFindCntlr



Diese Funktion gibt den Zählerstand des RAID-Controllers und die entsprechenden Controller-IDs zurück.

HINWEIS

Der Funktionsbaustein sollte nur einmal in einem SPS-Programm aufgerufen werden!

Die Systemperformance kann durch zyklisches Aufrufen dieses Funktionsblocks dramatisch beeinflusst werden.

Eingänge

```
VAR_INPUT
sNETID : T_AmsNetId;
bWrtRd : BOOL;
tTimeOut : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```



Name	Тур	Beschreibung
sNETID		lst ein String, der die AMS-Netzwerk-ID des Zielgerätes enthält, zu dem das ADS-Kommando geschickt wird.
bWrtRd		Das ADS-Kommando wird von der steigenden Flanke dieses Inputs getriggert.
tTimeOut		Gibt die Timeout-Zeit an, die bei der Ausführung des Kommandos nicht überschritten werden darf.

Ausgänge

VAR_OUTPUT

stRAIDCntlrFound : ST_RAIDCntlrFound;

nBytesRead : UDINT;
bBusy : BOOL;
bError : BOOL;
nErrorID : UDINT;

END_VAR

Name	Тур	Beschreibung
stRAIDCntlrFound	ST RAIDCntlrFound [▶ 146]	Enthält die Anzahl der gefundenen RAID-Controller und ihre RAID-Controller-Ids.
nBytesRead	UDINT	Anzahl der erfolgreich zurückgegebenen Datenbytes.
bBusy	BOOL	Dieser Ausgang bleibt so lange TRUE, bis der Block ein Kommando ausgeführt hat, aber längstens für die Zeit, die am Timeout'-Input ansteht. Solange Busy = TRUE, wird kein neues Kommando an den Inputs angenommen werden. Bitte beachten Sie, dass es nicht die Ausführung des Dienstes ist, sondern die Zeit in der er ausgeführt werden darf, die hier gemonitort wird.
bError	BOOL	Dieser Ausgang wird auf TRUE gesetzt, wenn ein Fehler während der Ausführung eines Kommandos auftritt. Der kommandospezifische Error-Code steht in 'nErrorld'. Wenn der Block einen Timeout-Error hat, wird 'bError' auf TRUE gesetzt und 'nErrorld' ist 1861 (hexadezimal 0x745). Es wird auf FALSE gesetzt, wenn ein Kommando an den Eingängen ausgeführt wird.
nErrorID	UDINT	Liefert bei einem gesetzten ERR-Ausgang die ADS-Fehlernummer [108].

Voraussetzungen

Entwicklungsumgebung	•	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_loFunctions (IO)

3.11.2 FB_RAIDGetInfo



Dieser Funktionsblock gibt ein RAID-Info zurück, das die Anzahl der RAID-Controller-Sets und die maximale Anzahl von Antrieben pro Set enthält.



HINWEIS

Der Funktionsbaustein sollte nur einmal in einem SPS-Programm aufgerufen werden!

Die Systemperformance kann durch zyklisches Aufrufen dieses Funktionsblocks dramatisch beeinflusst werden.

Eingänge

VAR_INPUT

sNETID : T_AmsNetId; bWrtRd : BOOL; nRAIDCntlrID : UDINT;

tTimeOut : TIME := DEFAULT_ADS_TIMEOUT;

END_VAR

Name	Тур	Beschreibung
sNETID	T_AmsNetID	Ist ein String, der die AMS-Netzwerk-ID des Zielgerätes enthält, zu dem das ADS-Kommando geschickt wird.
bWrtRd	BOOL	Das ADS-Kommando wird von der steigenden Flanke dieses Inputs getriggert.
nRAIDCntlrID	UDINT	Die RAID-Controller-ID (kann mit FB_RAIDCntlrFind gelesen werden).
tTimeOut	TIME	Gibt die Timeout-Zeit an, die bei der Ausführung des Kommandos nicht überschritten werden darf.

Ausgänge

VAR_OUTPUT

stRAIDInfo : ST_RAIDInfo;
nBytesRead : UDINT;
bBusy : BOOL;
bError : BOOL;
nErrorID : UDINT;

END VAR

Name	Тур	Beschreibung
stRAIDInfo	ST_RAIDInfo [▶ 147]	Gibt ein RAID-Info zurück, das die Anzahl von RAID- Controller-Sets und die maximale Anzahl von Antrieben pro Set enthält.
nBytesRead	UDINT	Anzahl der erfolgreich zurückgegebenen Datenbytes.
bBusy	BOOL	Dieser Ausgang bleibt so lange TRUE, bis der Block ein Kommando ausgeführt hat, aber längstens für die Zeit, die am Timeout'-Input ansteht. Solange Busy = TRUE, wird kein neues Kommando an den Inputs angenommen werden. Bitte beachten Sie, dass es nicht die Ausführung des Dienstes ist, sondern die Zeit in der er ausgeführt werden darf, die hier gemonitort wird.
bError	BOOL	Dieser Ausgang wird auf TRUE gesetzt, wenn ein Fehler während der Ausführung eines Kommandos auftritt. Der kommandospezifische Error-Code steht in 'nErrorld'. Wenn der Block einen Timeout-Error hat, wird 'bError' auf TRUE gesetzt und 'nErrorld' ist 1861 (hexadezimal 0x745). Es wird auf FALSE gesetzt, wenn ein Kommando an den Eingängen ausgeführt wird.
nErrorID	UDINT	Liefert bei einem gesetzten ERR-Ausgang die <u>ADS-</u> <u>Fehlernummer [▶ 163]</u> .



Entwicklungsumgebung	•	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_loFunctions (IO)

3.11.3 FB_RAIDGetStatus

```
FB_RAIDGetStatus
sNETID T_AmsNetId
                                     ST_RAIDStatusRes stRAIDStatusRes
bWrtRd BOOL
                                                    UDINT nBytesRead
stRAIDConfigReq ST_RAIDConfigReq
                                                         BOOL bBusy
                                                         BOOL bError
tTimeOut TIME
                                                      UDINT nErrorID
```

Dieser Funktionsblock gibt den RAID-Set-Index, den RAID-Typ, den RAID-Status, die Anzahl der RAID-Antriebe und den Status der RAID-Antriebe zurück.

HINWEIS

Höchstens 1mal pro Sekunde aufrufen!

Die Systemperformance kann durch zyklisches Aufrufen dieses Funktionsblocks dramatisch beeinflusst werden.

Eingänge

VAR_INPUT sNETID : T_AmsNetId; bWrtRd : BOOL;

stRAIDConfigReq : ST_RAIDConfigReq;

tTimeOut : TIME := DEFAULT_ADS_TIMEOUT;

END VAR

Name	Тур	Beschreibung
sNETID	T_AmsNetID	lst ein String, der die AMS-Netzwerk-ID des Zielgerätes enthält, zu dem das ADS-Kommando geschickt wird.
bWrtRd		Das ADS-Kommando wird von der steigenden Flanke dieses Inputs getriggert.
stRAIDConfigReq	ST_RAIDConfigReq [▶ 147]	RAID-Konfiguration Request-Parameters werden in dieser Struktur festgelegt. Sie enthält die Controller ID und den Index des RAID-Sets.
tTimeOut	TIME	Gibt die Timeout-Zeit an, die bei der Ausführung des Kommandos nicht überschritten werden darf.

Ausgänge

VAR OUTPUT

stRAIDStatusRes : ST_RAIDStatusRes;
nBytesRead : UDINT;
bBusy : BOOL; nErrorID : BOOL; : UDINT; END_VAR

Name	Тур	Beschreibung
stRAIDStatusRes	<u> </u>	Diese Struktur enthält die RAID-Status-Antwort, den RAID-Set-Index, den RAID-Typ, den RAID-Status, die Anzahl der RAID-Antriebe, den Status der RAID-Antriebe.
nBytesRead	UDINT	Anzahl der erfolgreich zurückgegebenen Datenbytes.



Name	Тур	Beschreibung
bBusy	BOOL	Dieser Ausgang bleibt so lange TRUE, bis der Block ein Kommando ausgeführt hat, aber längstens für die Zeit, die am Timeout'-Input ansteht. Solange Busy = TRUE, wird kein neues Kommando an den Inputs angenommen werden. Bitte beachten Sie, dass es nicht die Ausführung des Dienstes ist, sondern die Zeit in der er ausgeführt werden darf, die hier gemonitort wird.
bError	BOOL	Dieser Ausgang wird auf TRUE gesetzt, wenn ein Fehler während der Ausführung eines Kommandos auftritt. Der kommandospezifische Error-Code steht in 'nErrorld'. Wenn der Block einen Timeout-Error hat, wird 'bError' auf TRUE gesetzt und 'nErrorld' ist 1861 (hexadezimal 0x745). Es wird auf FALSE gesetzt, wenn ein Kommando an den Eingängen ausgeführt wird.
nErrorID	UDINT	Liefert bei einem gesetzten ERR-Ausgang die <u>ADS-</u> <u>Fehlernummer [▶ 163]</u> .

Entwicklungsumgebung	•	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_loFunctions (IO)

3.12 SERCOS

3.12.1 IOF_SER_GetPhase



Der Funktionsbaustein IOF_SER_GetPhase ermittelt die aktuelle Kommunikationsphase auf dem SERCOS-Ring. Die Kommunikationsphasen können die Werte von 0 bis 4 annehmen.

Eingänge

```
VAR_INPUT

NETID : T_AmsNetId;

DEVICEID : UDINT;

GET : BOOL;

TMOUT : TIME := DEFAULT_ADS_TIMEOUT;

END VAR
```

Name	Тур	Beschreibung
NETID	T_AmsNetId	Hier kann die Netzwerkadresse des TwinCAT-Rechners angegeben werden, auf dem das ADS-Kommando ausgeführt werden soll. Für den lokalen Rechner kann auch ein Leerstring angegeben werden.
DEVICEID	UDINT	Über die Deviceld (Geräte-Id) wird der SERCOS-Master spezifiziert, dessen Kommunikationsphase ermittelt werden soll. Die Geräte-Ids werden während der Hardware-Konfiguration von TwinCAT-System festgelegt.
GET	BOOL	Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.



Name	Тур	Beschreibung
TMOUT	TIME	Gibt die Timeout-Zeit an, die bei der Ausführung des
		Kommandos nicht überschritten werden darf.

Ausgänge

VAR_OUTPUT

BUSY : BOOL;

ERR : BOOL;

ERRID : UDINT;

PHASE : BYTE;

END VAR

Name	Тур	Beschreibung
BUSY	BOOL	Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
ERR	BOOL	Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der BUSY-Ausgang zurückgesetzt wurde.
ERRID	UDINT	Liefert bei einem gesetzten ERR-Ausgang die <u>ADS-</u> <u>Fehlernummer [▶ 163]</u> .
PHASE	BYTE	Die aktuelle Kommunikationsphase im SERCOS-Ring.

Voraussetzungen

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	- (/	Sercans SCS-P ISA; Sercans SCS-P PCI; Beckhoff FC750x PCI	Tc2_loFunctions (IO)

3.12.2 IOF_SER_SaveFlash



Der Funktionsbaustein IOF_SER_SaveFlash überprüft die im DPRAM-Speicher stehenden Systemparameter. Wenn kein Fehler vorliegt aktiviert und speichert er sie ins EEPROM. Der Funktionsbaustein kann Systemparameter im EEPROM von der Steuerung passend zur Applikation einstellen.

HINWEIS

Das EEPROM besitzt eine maximale Wiederbeschreibbarkeit von 100 000 Schreibzyklen.

Die SPS sollte diesen Funktionsbaustein nicht automatisch, sondern nur durch den Anwender gezielt aktivieren.

Eingänge

VAR_INPUT NETID

NETID : T_AmsNetId;
DEVICEID : UDINT;
SAVE : BOOL:

SAVE : BOOL; TMOUT : TIME := DEFAULT_ADS_TIMEOUT;

END_VAR



Name	Тур	Beschreibung
NETID	T_AmsNetId	Hier kann die Netzwerkadresse des TwinCAT-Rechners angegeben werden, auf dem das ADS-Kommando ausgeführt werden soll. Für den lokalen Rechner kann auch ein Leerstring angegeben werden.
DEVICEID	UDINT	Über die Deviceld (Geräte-Id) wird der SERCOS-Master spezifiziert, dessen Systemparameter gespeichert werden sollen. Die Geräte-Ids werden während der Hardware-Konfiguration vom TwinCAT-System festgelegt.
SAVE	BOOL	Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.
TMOUT	TIME	Gibt die Timeout-Zeit an, die bei der Ausführung des Kommandos nicht überschritten werden darf.

Ausgänge

VAR OUTPUT

BUSY : BOOL;
ERR : BOOL;
ERRID : UDINT;
END_VAR

Name	Тур	Beschreibung
BUSY	BOOL	Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
ERR	BOOL	Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der BUSY-Ausgang zurückgesetzt wurde.
ERRID	UDINT	Liefert bei einem gesetzten ERR-Ausgang die ADS-Fehlernummer [• 163].

Voraussetzungen

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0		Sercans SCS-P ISA; Sercans SCS-P PCI; Beckhoff FC750x PCI	Tc2_loFunctions (IO)

3.12.3 IOF_SER_ResetErr



Der Funktionsbaustein IOF_SER_ResetErr löscht folgende Fehler eines SERCOS-Masters:

- Die Fehler in den vorhandenen Antrieben;
- Den Diagnosestatus im Diagnosekanal der vorhandenen Antriebe;
- · Den Systemfehler;



Eingänge

VAR INPUT : T_AmsNetId; NETID DEVICEID : UDINT;

RESET : BOOL;
TMOUT : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR

Name	Тур	Beschreibung
NETID	T_AmsNetId	Hier kann die Netzwerkadresse des TwinCAT-Rechners angegeben werden, auf dem das ADS-Kommando ausgeführt werden soll. Für den lokalen Rechner kann auch ein Leerstring angegeben werden.
DEVICEID	UDINT	Über die Deviceld (Geräte-Id) wird der SERCOS-Master spezifiziert, dessen Fehler gelöscht werden sollen. Die Geräte-Ids werden während der Hardware-Konfiguration von TwinCAT-System festgelegt.
RESET	BOOL	Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.
TMOUT	TIME	Gibt die Timeout-Zeit an, die bei der Ausführung des Kommandos nicht überschritten werden darf.

Ausgänge

VAR_OUTPUT

BUSY : BOOL; ERR : BOOL; ERRID : UDINT; END_VAR

Name	Тур	Beschreibung
BUSY	BOOL	Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
ERR	BOOL	Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der BUSY-Ausgang zurückgesetzt wurde.
ERRID	UDINT	Liefert bei einem gesetzten ERR-Ausgang die ADS-Fehlernummer [10].

Voraussetzungen

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0		Sercans SCS-P ISA; Sercans SCS-P PCI; Beckhoff FC750x PCI	Tc2_loFunctions (IO)

IOF_SER_SetPhase 3.12.4

```
IOF_SER_SetPhase
NETID T_AmsNetId BOOL BUSY
DEVICEID UDINT BOOL ERR
PHASE BYTE UDINT ERRID
SET BOOL
TMOUT TIME
```

Der Funktionsbaustein IOF_SER_SetPhase führt den Phasenhochlauf im SERCOS-Ring auf einen bestimmten Wert durch.



Eingänge

VAR_INPUT VAR_INPUT

NETID : T_AmsNetId;

DEVICEID : UDINT;

PHASE : BYTE;

SET : BOOL;

TMOUT : TIME := DEFAULT_ADS_TIMEOUT;

END_VAR

Name	Тур	Beschreibung
NETID	T_AmsNetId	Hier kann die Netzwerkadresse des TwinCAT-Rechners angegeben werden, auf dem das ADS-Kommando ausgeführt werden soll. Für den lokalen Rechner kann auch ein Leerstring angegeben werden.
DEVICEID	UDINT	Über die Deviceld (Geräte-Id) wird der SERCOS-Master spezifiziert, dessen Kommunikationsphase gesetzt werden soll. Die Geräte-IDs werden während der Hardware-Konfiguration von TwinCAT-System festgelegt.
PHASE	BOOL	Die zu setzende Kommunikationsphase im SERCOS-Ring.
SET	BOOL	Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.
TMOUT	TIME	Gibt die Timeout-Zeit an, die bei der Ausführung des Kommandos nicht überschritten werden darf.

Ausgänge

VAR_OUTPUT

BUSY : BOOL; ERR : BOOL; ERRID : UDINT;

END_VAR

Name	Тур	Beschreibung
BUSY	BOOL	Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
ERR	BOOL	Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der BUSY-Ausgang zurückgesetzt wurde.
ERRID	UDINT	Liefert bei einem gesetzten ERR-Ausgang die ADS-Fehlernummer [• 163].

Voraussetzungen

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	,	Sercans SCS-P ISA; Sercans SCS-P PCI; Beckhoff FC750x PCI	Tc2_loFunctions (IO)

TE1000 103 Version: 1.5.3



3.12.5 IOF_SER_IDN_Read

```
IOF_SER_IDN_Read
sNetId T_AmsNetId
                                          UDINT cbRead
nIDN UINT
                                       DWORD nAttribRd
bExecute BOOL
                              ST_SercosParamAttrib sAttrib
nPort UINT
                                           BOOL bBusy
nMode DINT
                                           BOOL bError
                                           UDINT nErrId
nAttrib DWORD
cbLen UDINT
dwDestAddr PVOID
tTimeout TIME
```

Der Funktionsbaustein IOF_SER_IDN_Read erlaubt das Lesen eines Wertes aus einem S- oder P-Parameter eines Sercos-Antriebes. Datentyp und Größe werden automatisch anhand des Attributes bestimmt.

Eingänge

```
VAR_INPUT
sNetId : T_AmsNetId;
nIDN : UINT;
bExecute : BOOL;
nPort : UINT;
nMode : DINT;
nAttrib : DWORD;
cbLen : UUINT;
dwDestAddr : PVOID;
tTimeout : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

Name	Тур	Beschreibung
sNetId	T_AmsNetID	Hier kann die Netzwerkadresse des TwinCAT-Rechners angegeben werden, auf dem das ADS-Kommando ausgeführt werden soll. Für den lokalen Rechner kann auch ein Leerstring angegeben werden.
nIDN	UINT	Beinhaltet die Sercos-Parameter-Nummer, auf die lesend zugegriffen werden soll. Für S-Parameters muss nIDN zwischen 0 und 32767 liegen, für P-Parameters zwischen 32768 und 65535.
bExecute	BOOL	Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.
nPort	UINT	Die Port-Number nPort wird vom TwinCAT-System während der Hardware-Konfiguration vergeben.
nMode	DINT	Der Lesemodus bestimmt, welcher Teil des Parameters gelesen werden soll. nMode = 0: Wert nMode = 2: Name nMode = 3: Attribut (wird immer gelesen, um Datentyp und Größe zu bestimmen, es sei denn, nAttrib ist <> 0) nMode = 4: Einheit (nicht für jeden Parameter verfügbar) nMode = 5: Minimum (nicht für jeden Parameter verfügbar) nMode = 6: Maximum (nicht für jeden Parameter verfügbar)
nAttrib	DWORD	Attribut des Parameter, wenn es bekannt ist. Wenn nAttrib = 0 dann liest IOF_SER_IDN_Write erst das Parameter-Attribut vom Antrieb, bevor der Wert in den Parameter des Antriebs geschrieben wird.
cbLen	UDINT	Maximale Länge des Datapuffers, der den Wert aufnehmen soll.
dwDestAddr	PVOID	Adresse des Datapuffers, der den Wert aufnehmen soll.



Name	Тур	Beschreibung
tTimeout	TIME	Gibt die Timeout-Zeit an, die bei der Ausführung des
		Kommandos nicht überschritten werden darf.

Ausgänge

VAR_OUTPUT

: UDINT; cbRead

cokead : UDINT;
nAttribRd : DWORD;
sAttrib : ST_SercosParamAttrib;
bBusy : BOOL;
bError : BOOL;
nErrId : UDINT;
VAR

END_VAR

Name	Тур	Beschreibung
cbRead	UDINT	Anzahl der gelesenen und nach dwDestAddr kopierten Bytes.
nAttribRd	DWORD	Attribut des Parameter, kann für den nächsten Zugriff (nAttrib) auf den Parameter gespeichert werden.
sAttrib	ST SercosParamAttrib [▶ 148]	Beinhaltet das Attribut nAttribRd des Sercos-Parameters in einzelne Variablen zerlegt.
bBusy	BOOL	Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
bError	BOOL	Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde.
nErrld	UDINT	Liefert bei einem gesetzten ERR-Ausgang die <u>ADS-Fehlernummer [• 163]</u> bzw. die spezifische Funktionsbaustein-Fehlernummer.

spezifische Funktionsbaustein- Fehlernummer	Beschreibung
0x1003	Falscher Parameter-Mode
0x1004	Falsche Parameterdatengröße

Voraussetzungen

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	\ /	Sercans SCS-P ISA; Sercans SCS-P PCI; Beckhoff FC750x PCI	Tc2_loFunctions (IO)

3.12.6 IOF_SER_IDN_Write



TE1000 105 Version: 1.5.3



Der Funktionsbaustein IOF_SER_IDN_Write erlaubt das Schreiben eines Wertes in einen S- oder P-Parameter eines Sercos-Antriebes. Datentyp und Größe werden automatisch anhand des Attributes bestimmt.

Eingänge

```
VAR_INPUT

sNetId : T_AmsNetId;

nIDN : UINT; (* S: 0****** ********, P: 1****** *************************

bExecute : BOOL;

nPort : UINT;

nAttrib : DWORD;

cbLen : UDINT;

dwSrcAddr : PVOID;

tTimeout : TIME := DEFAULT_ADS_TIMEOUT;

END VAR
```

Name	Тур	Beschreibung
sNetId	T_AmsNetID	Hier kann die Netzwerkadresse des TwinCAT-Rechners angegeben werden, auf dem das ADS-Kommando ausgeführt werden soll. Für den lokalen Rechner kann auch ein Leerstring angegeben werden.
nIDN	UINT	Beinhaltet die Sercos-Parameter-Nummer, auf die schreibend zugegriffen werden soll. Für S-Parameters muss nIDN zwischen 0 und 32767 liegen, für P-Parameters zwischen 32768 und 65535.
bExecute	BOOL	Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.
nPort	UINT	Die Port-Number nPort wird vom TwinCAT System während der Hardware-Konfiguration vergeben.
nAttrib	DWORD	Attribut des Parameter, wenn es bekannt ist. Wenn nAttrib = 0 dann liest IOF_SER_IDN_Write erst das Parameter-Attribut vom Antrieb, bevor der Wert in den Parameter des Antriebs geschrieben wird.
cbLen	UDINT	Länge des Datapuffers, der den Wert enthält.
dwDestAddr	PVOID	Adresse des Datapuffers, der den Wert enthält.
tTimeout	TIME	Gibt die Timeout-Zeit an, die bei der Ausführung des Kommandos nicht überschritten werden darf.

Ausgänge

```
VAR_OUTPUT

nAttribRd : DWORD;
sAttrib : ST_SercosParamAttrib;
bBusy : BOOL;
bError : BOOL;
nErrId : UDINT;
END_VAR
```

Name	Тур	Beschreibung
nAttribRd	DWORD	Attribut des Parameter, kann für den nächsten Zugriff (nAttrib) auf den Parameter gespeichert werden.
sAttrib	ST_SercosParamAttrib [• 148]	Beinhaltet das Attribut nAttribRd des Sercos-Parameters in einzelne Variablen zerlegt.
bBusy	BOOL	Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
bError	BOOL	Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde.



Name	Тур	Beschreibung
nErrld	UDINT	Liefert bei einem gesetzten ERR-Ausgang die ADS-
		Fehlernummer [163] bzw. die spezifische
		Funktionsbaustein-Fehlernummer.

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	,	1	Tc2_loFunctions (IO)
		Sercans SCS-P PCI;	
		Beckhoff FC750x PCI	

3.12.7 IOF_SER_DRIVE_Backup

```
IOF_SER_DRIVE_Backup
bCheck BOOL
                                    BOOL bBusy
                                   BOOL bError
bBackup BOOL
                                  UDINT nErrId
bRestore BOOL
bCRCEnable BOOL
                                BOOL bCheckOK
bStdBackupList BOOL
sNetId T_AmsNetId
nPort UINT
sComment T_MaxString
ePath E_OpenPath
sPathName T_MaxString
tTimeout TIME
```

Der Funktionsbaustein IOF_SER_DRIVE_Backup erlaubt das Backup und Restore der Antriebsdaten (Sund P-Parameter) von der SPS in eine Binärdatei. Die Liste der zu sichernden S- und P-Parameterdaten wird standardmäßig dem Sercos-Parameter IDN192 entnommen. Backup und Restore erfordern den SERCOS-Parameter-Mode (Phase 2).

Wenn bStdBackupList = TRUE (Standard) dann wird der Parameter IDN192 als Liste der zu sichernden Daten genommen, anderenfalls IDN17, die Liste aller Sercos-Parameter. Restore erfordert eine Backup-Datei, die mit Parameter IDN192 erstellt wurde, da einige Parameter der Liste IDN17 schreibgeschützt sind.

Backup und Restore erzeugen eine CRC16-CCITT und eine 16 bit Check-Summe und speichern diese in Parameter IDN142, wenn verfügbar.

Das Dateiformat der Backup-Datei ist in <u>Backup-Dateiformat</u> [<u>158</u>] beschrieben.

Eingänge

```
VAR_INPUT

bCheck : BOOL;

bBackup : BOOL;

bRestore : BOOL;

bCRCEnable : BOOL := TRUE;

bStdBackupList: BOOL := TRUE;

sNetId : T_AmsNetId;

nPort : UINT;

sComment : T_MaxString;

ePath : E_OpenPath := PATH_BOOTPATH;

sPathName : T_MaxString := 'DRIVEPAR.BIN';

tTimeout : TIME := DEFAULT_ADS_TIMEOUT;

END_VAR
```

Name	Тур	Beschreibung
bCheck	BOOL	Über eine positive Flanke an diesem Eingang wird die Überprüfung per CRC und Checksumme aktiviert. CRC und Checksumme werden persistent und im Parameter



Name	Тур	Beschreibung	
		IDN142 nach einem Backup oder Restore gespeichert. Wenn der Wert aus Parameter IDN142 und die persistenten Daten übereinstimmen wird bCheckOK auf TRUE gesetzt, anderenfalls wird bCheckOK auf FALSE gesetzt.	
bBackup	BOOL	Über eine positive Flanke an diesem Eingang wird das Backup aktiviert.	
bRestore	BOOL	Über eine positive Flanke an diesem Eingang wird das Restore aktiviert.	
bCRCEnable	BOOL	Der CRC16-CCITT und die 16 bit Checksumme werden über bCRCEnable = TRUE aktiviert. Die CRC und die Checksumme werden in Parameter IDN142 gespeichert, wenn bCRCEnable = TRUE.	
bStdBackupList	BOOL	Bestimmt, welche Parameterliste für das Backup benutzt wird. Standardmäßig wird IDN192 (bStdBackupList = TRUE) für das Backup benutzt, wenn bStdBackupList = FALSE dann wird die Liste aller Parameter IDN017 benutzt. Restore benötigt eine Backup-Datei, die mit der Liste IDN192 erzeugt wurde.	
sNetId	T_AmsNetID	Hier kann die Netzwerkadresse des TwinCAT-Rechners angegeben werden, auf dem das ADS-Kommando durchgeführt werden soll. Für den lokalen Rechner kann auch ein Leerstring angegeben werden.	
nPort	UINT	Die Port-Number nPort wird vom TwinCAT System während der Hardware-Konfiguration vergeben.	
sComment	T_MaxString	Kommentar, der in den Datei-Header der Backup-Datei geschrieben wird.	
ePath	E_OpenPath	Bestimmt den Pfad der Backup-Datei. Wenn ePath = PATH_BOOTPATH dann wird der TwinCAT BOOT-Pfad genommen, bei ePath = PATH_GENERIC wird der in sPathName spezifizierte Pfad genommen.	
sPathName	T_MaxString	Beinhaltet den Dateinamen (bei Verwendung des Boot- Pfades) bzw. den kompletten Pfad und Dateinamen bei Verwendung des generischen Pfades.	
tTimeout	TIME	Gibt die Timeout-Zeit an, die bei der Ausführung des Kommandos nicht überschritten werden darf.	

Ausgänge

VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
nErrId : UDINT;
bCheckOK : BOOL;
END_VAR

Name	Тур	Beschreibung
bBusy	BOOL	Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
bError	BOOL	Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde.
nErrld	UDINT	Liefert bei einem gesetzten ERR-Ausgang die <u>ADS-Fehlernummer [▶ 163]</u> bzw. die spezifische Funktionsbaustein-Fehlernummer.

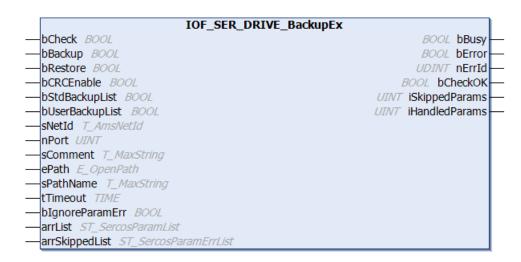


spezifische Funktionsbaustein- Fehlernummer	Beschreibung
0x1003	Falscher Parameter-Mode
0x1004	Falsche Parameterdatengröße
0x1005	Falscher Backup Parameter Typ
0x1006	Backup Parameterliste war nicht IDN 192

Name	Тур	Beschreibung
bCheckOk	BOOL	lst WAHR, wenn der Checksummen-Test erfolgreich war.

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	, ,	,	Tc2_loFunctions (IO)
		Sercans SCS-P PCI;	
		Beckhoff FC750x PCI	

3.12.8 IOF_SER_DRIVE_BackupEx



Der Funktionsbaustein IOF_SER_DRIVE_BackupEx erlaubt das Backup (Sichern) und Restore (Wiederherstellen) der Antriebsdaten (S- und P-Parameter) über die SPS in eine Binärdatei bzw. zurück in den Antrieb. Die Liste der zu sichernden S- und P-Parameterdaten wird standardmäßig dem Sercos-Parameter IDN192 entnommen. Backup und Restore erfordern den SERCOS-Parameter-Mode (Phase 2).

Wenn bStdBackupList = TRUE (Standard) ist, dann wird der Parameter IDN192 als Liste der zu sichernden Daten genommen.

Wenn bUserBackupList = TRUE ist, dann wird der die Parameterliste arrList als Liste der zu sichernden Daten genommen.

Anderenfalls wird IDN17 verwendet, die Liste aller Sercos-Parameter.

Restore erfordert eine Backup-Datei, die mit Parameter IDN192 oder mit einer Userparameterliste erstellt wurde. Einige Parameter der Liste IDN17 sind schreibgeschützt.

Backup und Restore können eine CRC16-CCITT und eine 16 bit Check-Summe erstellen und speichern diese in Parameter IDN142, wenn verfügbar. Die Option bCRCEnable ist standardmäßig deaktiviert (FALSE).

Das Dateiformat der Backup-Datei ist in <u>Backup-Dateiformat [▶ 158]</u> beschrieben.



Eingänge

```
VAR_INPUT

bCheck : BOOL;
bBackup : BOOL;
bRestore : BOOL;
bCRCEnable : BOOL;
bStdBackupList : BOOL := TRUE;
bUserBackupList : BOOL;
sNetId : T_AmsNetId;
nPort : UINT;
sComment : T_MaxString;
ePath : E_OpenPath := PATH_BOOTPATH;
sPathName : T_MaxString := 'DRIVEPAR.BIN';
tTimeout : TIME := DEFAULT_ADS_TIMEOUT;
bIgnoreParamErr : BOOL;
END_VAR
```

Name	Тур	Beschreibung
bCheck	BOOL	Über eine positive Flanke an diesem Eingang wird die Überprüfung per CRC und Checksumme aktiviert. CRC und Checksumme werden persistent und im Parameter IDN142 nach einem Backup oder Restore gespeichert. Wenn der Wert aus Parameter IDN142 und die persistenten Daten übereinstimmen wird bCheckOK auf TRUE gesetzt, anderenfalls wird bCheckOK auf FALSE gesetzt.
bBackup	BOOL	Über eine positive Flanke an diesem Eingang wird das Backup aktiviert.
bRestore	BOOL	Über eine positive Flanke an diesem Eingang wird das Restore aktiviert.
bCRCEnable	BOOL	Der CRC16-CCITT und die 16 bit Checksumme werden über bCRCEnable = TRUE aktiviert. Die CRC und die Checksumme werden in Parameter IDN142 gespeichert, wenn bCRCEnable = TRUE.
bStdBackupList	BOOL	Bestimmt, welche Parameterliste für das Backup benutzt wird. Standardmäßig wird IDN192 (bStdBackupList = TRUE) für das Backup benutzt, wenn bStdBackupList = FALSE dann wird die Liste aller Parameter IDN017 benutzt. Restore benötigt eine Backup-Datei, die mit der Liste IDN192 erzeugt wurde.
bUserBackupList	BOOL	Bestimmt, ob eine benutzerdefinerte Parameterliste arrList für das Backup benutzt wird. Standardmäßig wird IDN192 (bStdBackupList = TRUE) für das Backup benutzt, wenn bStdBackupList = FALSE und bUserBackupList = TRUE ist, dann wird die Liste arrList benutzt. Restore benötigt eine Backup-Datei, die mit der Liste IDN192 oder einer benutzerdefinierten Parameterliste erzeugt wurde.
sNetId	T_AmsNetID	Hier kann die Netzwerkadresse des TwinCAT-Rechners angegeben werden, auf dem das ADS-Kommando durchgeführt werden soll. Für den lokalen Rechner kann auch ein Leerstring angegeben werden.
nPort	UINT	Die Port-Number nPort wird vom TwinCAT System während der Hardware-Konfiguration vergeben.
sComment	T_MaxString	Kommentar, der in den Datei-Header der Backup-Datei geschrieben wird.
ePath	E_OpenPath	Bestimmt den Pfad der Backup-Datei. Wenn ePath = PATH_BOOTPATH dann wird der TwinCAT BOOT-Pfad genommen, bei ePath = PATH_GENERIC wird der in sPathName spezifizierte Pfad genommen.
sPathName	T_MaxString	Beinhaltet den Dateinamen (bei Verwendung des Boot- Pfades) bzw. den kompletten Pfad und Dateinamen bei Verwendung des generischen Pfades.



Name	Тур	Beschreibung
tTimeout	TIME	Gibt die Timeout-Zeit an, die bei der Ausführung des Kommandos nicht überschritten werden darf.
blgnoreParamErr	BOOL	Bestimmt, ob bei Parameterlese- oder Parameterschreibfehlern das Backup/Restore fortgeführt oder abgebrochen werden soll. Standardmäßig wird bei Fehlern abgebrochen (blgnoreParamErr = FALSE). Ist das Ignorieren von Fehlern aktiviert (blgnoreParamErr = TRUE), dann werden in der Liste der übersprungenen Parameter arrSkippedList die Parameternummer und die Fehlernummern hinterlegt.

▼/ ■ Ein-/Ausgänge

VAR_IN_OUT

_IN_OUT arrList : ST_SercosParamList; arrSkippedList : ST_SercosParamErrList; arrList

END_VAR

Name	Тур	Beschreibung
arrList	ST_SercosParamList [▶ 150]	Bei Standardbackup über IDN192 (bStdBackupList = TRUE) stehen in dieser Liste nach dem Backup die Backup-Parameter aus IDN192. Bei benutzerdefiniertem Backup (bUserBackupList = TRUE und bStdBackupList = FALSE) müssen in dieser Liste vor dem Backup die Liste der zu sichernden Parameter stehen. Bei Backup über IDN17 (bUserBackupList = FALSE und bStdBackupList = FALSE) stehen in dieser Liste nach dem Backup die Liste vorhandenen Parameter aus IDN17.
arrSkippedList	ST_SercosParamErrList [• 149]	Enthält eine Liste der übersprungenen Parameter (die Parameternummer und die Fehlernummern).

Ausgänge

VAR_OUTPUT

bBusy : BOOL;

bError : BOOL;

nErrId : UDINT;

bCheckOK : BOOL; iSkippedParams : UINT; iHandledParams : UINT;

END_VAR

Name	Тур	Beschreibung
bBusy	BOOL	Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
bError	BOOL	Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde.
nErrld	UDINT	Liefert bei einem gesetzten ERR-Ausgang die <u>ADS-Fehlernummer [* 163]</u> bzw. die spezifische Funktionsbaustein-Fehlernummer.

spezifische Funktionsbaustein- Fehlernummer	Beschreibung
0x1003	Falscher Parameter-Mode
0x1004	Falsche Parameterdatengröße
0x1005	Falscher Backup Parameter Typ
0x1006	Backup Parameterliste war nicht IDN 192

TE1000 111 Version: 1.5.3



Name	Тур	Beschreibung
bCheckOk	BOOL	lst WAHR, wenn der Checksummen-Test erfolgreich war.
iSkippedParams	UINT	Enthält die Anzahl der übersprungenen Parameter (siehe arrSkippedList), falls das Ignorieren der Parameterfehler aktiv war (blgnoreParamErr = TRUE).
iHandledParams	UINT	Enthält die Anzahl der erfolgreich gesicherten/ wiederhergestellten Parameter.

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0		Sercans SCS-P ISA; Sercans SCS-P PCI; Beckhoff FC750x PCI	Tc2_loFunctions (IO)

3.12.9 IOF_SER_DRIVE_Reset

Der Funktionsbaustein IOF_SER_DRIVE_Reset führt einen Antriebs-Reset eines Sercos-Antriebes durch. Antriebsfehler werden gelöscht.

Eingänge

```
VAR_INPUT

sNetId : T_AmsNetId;

nPort : UINT;

bReset : BOOL;

tTimeout : TIME := DEFAULT_ADS_TIMEOUT;

END VAR
```

Name	Тур	Beschreibung
sNetId	T_AmsNetID	Hier kann die Netzwerkadresse des TwinCAT-Rechners angegeben werden, auf dem das ADS-Kommando ausgeführt werden soll. Für den lokalen Rechner kann auch ein Leerstring angegeben werden.
nPort	UINT	Die Port-Number nPort wird vom TwinCAT-System während der Hardware-Konfiguration vergeben.
bReset	BOOL	Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.
tTimeout	TIME	Gibt die Timeout-Zeit an, die bei der Ausführung des Kommandos nicht überschritten werden darf.

Ausgänge

```
VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
nErrId : UDINT;
END_VAR
```



Name	Тур	Beschreibung
bBusy	BOOL	Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
bError	BOOL	Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde.
nErrId	UDINT	Liefert bei einem gesetzten ERR-Ausgang die ADS- Fehlernummer [▶ 163] bzw. die spezifische Funktionsbaustein-Fehlernummer.

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	- (/	Sercans SCS-P ISA; Sercans SCS-P PCI; Beckhoff FC750x PCI	Tc2_loFunctions (IO)

3.13 TcTouchLock

TcTouchLock ist eine Funktion der Beckhoff IPC Panel.

Weitere Informationen dazu finden Sie in der Dokumentation Touch Lock.

Die Hardware-Voraussetzungen finden Sie ebenfalls dort, in dem Kapitel Voraussetzungen.

3.13.1 FB_TcTouchLock_AcquireFocus



Der Funktionsbaustein FB_TcTouchLock_AcquireFocus dient der Vermeidung von parallelen, sich störenden Eingaben über mehrere Multitouch-Control-Panel, die an einen IPC angeschlossen sind. Zu diesem Zweck wird ein Fokus auf eines der angeschlossenen Control-Panel gelegt und damit die Eingabe über alle anderen angeschlossenen Control-Panel gesperrt. Mit dem Funktionsbaustein FB_TcTouchLock_AcquireFocus kann dieser Fokus angefordert und freigegeben werden.

Wird der Fokus an einem Multitouch-Control-Panel angefordert, wenn ein anderes Multitouch-Control-Panel ihn derzeit besitzt, so muss der Fokus von diesem zunächst frei gegeben werden. Sobald die Freigabe erfolgt ist, wird der Fokus automatisch auf das im Wartezustand befindliche Gerät gesetzt.

Die Multitouch-Control-Panel, auf die durch den Funktionsblock zugegriffen werden, müssen zuvor durch die Kommandozeilen-Applikation TcTouchLockService.exe konfiguriert werden. Dabei muss jedem Gerät eine spezifische Identifikationsnummer zugewiesen werden.

Eingänge

```
VAR_INPUT
bEnable : BOOL;
sSetID : STRING(32);
tLEDTime : TIME := 200;
END_VAR
```



Name	Тур	Beschreibung
bEnable	BOOL	TRUE = Fokus anfordern, FALSE = Fokus abgeben
sSetID	STRING	ID des Gerätes
tLEDTime	TIME	Die Ausgabe-LED blinkt im festgelegten Intervall (100ms – 1s), während der Fokus angefordert wird.

Ausgänge

```
VAR_OUTPUT
    bAcquired : BOOL := FALSE;(* Focus status information *)
    bLED : BOOL := FALSE;(* LED control output *)
    bBusy : BOOL;(* TRUE => function in progress *)
    bError: : BOOL;(* Error flag *)
    nErrID : UDINT;(* Error code *)
END_VAR
```

Name	Тур	Beschreibung
bAcquired		TRUE, wenn der Client den Fokus besitzt und FALSE, wenn er ihn verliert.
bLED	BOOL	Dieser Ausgang hat je nach Modus folgende Bedeutung:

Modus	Bedeutung
Konstant TRUE	Das Panel besitzt den Fokus
Konstant FALSE	Das Panel besitzt den Fokus nicht
Toggelt	Das Panel wartet darauf den Fokus zu erhalten

Name	Тур	Beschreibung
bBusy	BOOL	Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
bError	BOOL	Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde.
nErrld	UDINT	Liefert bei einem gesetzten ERR-Ausgang die <u>ADS-</u> <u>Fehlernummer [</u> ▶ 163] bzw. die spezifische Funktionsbaustein-Fehlernummer.

Fehlercodes	Fehlerbeschreibung
0x0000	Kein Fehler
0x0006	Ziel-Port nicht gefunden

Beispiel: Touch-Focus über Sondertaste steuern

Das manuelle Setzen des Fokus kann z.B. über eine Sondertaste des Panels gesteuert werden. Da der Fokus auch bei gesperrter Eingabe über den Touchscreen angefordert werden soll, ist eine Eingabemöglichkeit außerhalb des sperrbaren Touchscreens zu berücksichtigen. Über den TwinCAT System Manager wird die Sondertaste mit der entsprechenden Input-Variable des PLC Programms verknüpft. Pro Panel wird eine FB_TcTouchLock_AquireFocus Instanz erzeugt und mit der ID des Panels konfiguriert. Nach Drücken der Sondertaste an einem Panel, wobei der Baustein R_TRIG die steigende Flanke detektiert, versucht das PLC Programm über die entsprechende FB_TcTouchLock_AquireFocus Instanz den Touch-Fokus zu setzen. Der Funktionsbaustein kann außerdem einen Ausgang ansteuern (z.B. eine LED), der signalisiert, ob der Touch-Fokus erfolgreich gesetzt wurde oder ob noch versucht wird den Fokus zu holen. Erneutes Drücken der Sondertaste setzt den Touch-Fokus wieder zurück und ermöglicht damit ein Setzen des Touch-Fokus an anderen Panels.





Das PLC-Programm sieht für zwei Panels folgendermaßen aus:

```
PROGRAM MAIN
button1 AT%IX0.0 : BOOL;
button2 AT%IX0.1 : BOOL;
led1 AT%QX0.0 : BOOL;
led2 AT%QX0.1 : BOOL;
fbPanel1 : FB TcTouchLock AcquireFocus := ( sSetID := 'A' );
fbPanel2 : FB_TcTouchLock_AcquireFocus := ( sSetID := 'B' );
trigger1 : R TRIG;
trigger2 : R_TRIG;
END_VAR
(* Panel 1 *)
trigger1 ( CLK := button1 );
IF trigger1.Q THEN
fbPanel1.bEnable := NOT fbPanel1.bEnable;
END IF
fbPanel1(bLED=>LED1);
(* Panel 2 *)
trigger2 ( CLK := button2 );
IF trigger2.Q THEN
fbPanel2.bEnable := NOT fbPanel2.bEnable;
END IF
fbPanel2(bLED=>LED2);
```

Voraussetzungen

Entwicklungsumgebung	•	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1. >= 4022.31	PC oder CX (x86, x64)	Tc2_loFunctions (IO)

3.14 Drittherstellergeräte

3.14.1 Phoenix IBS SC/I-T

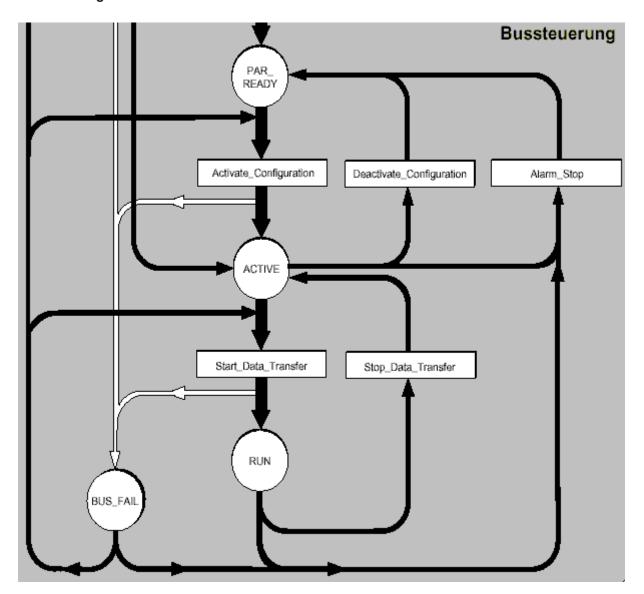
3.14.1.1 Übersicht

Die Bibliothek bietet eine komfortable Möglichkeit die wichtigsten Firmwaredienste der Phoenix Interbuskarte IBS SC/I-T zur Bussteuerung von der TwinCAT PLC auszuführen. Im folgenden Bild sind die Zustände und Übergangsbedingungen der Bussteuerung dargestellt.



TE1000

Bussteuerung



<u>SCIT_ActivateConfiguration [▶ 117]</u>: Führt den Befehl **Activate_Configuration** aus.

<u>SCIT_DeactivateConfiguration</u> [**\rightarrow** 118]: Führt den Befehl **Deactivate_Configuration** aus.

<u>SCIT_StartDataTransfer [▶ 119]</u>: Führt den Befehl **Start_Data_Transfer** aus.

<u>SCIT_StopDataTransfer</u> [▶ 120]: Führt den Befehl **Stop_Data_Transfer** aus.

SCIT_AlarmStop [▶ 121]: Führt den Befehl Alarm_Stop aus.

Konfiguration

<u>SCIT ControlActiveConfiguration</u> [▶ 123]: Dient zur Beeinflussung der aktiven Konfiguration der Interbus-Teilnehmer. Dieses Kommando kann sowohl im Zustand *PAR_READY* als auch im Zustand *ACTIVE* und *RUN* ausgeführt werden. Hierüber können einzelne, abhängige und gruppierte Teilnehmer aktiviert und deaktiviert werden.

Fehlerdiagnose

<u>SCIT_GetErrorInfo</u> [▶ 124]: Liefert Fehlerart und Fehlerort eines Interbus-Teilnehmers nach einem Busfehler.

<u>SCIT_ConfDevErrAll [</u>▶ <u>125]</u>: Peripheriestörungen aller Geräte quittieren.



3.14.1.2 SCIT_ActivateConfiguration

```
SCIT_ActivateConfiguration

NETID T_AmsNetId BOOL BUSY

PORT UINT BOOL ERR

WRTRD BOOL UDINT ERRID

TMOUT TIME WORD ADDERRORINFO
```

Der Funktionsbaustein SCIT_ActivateConfiguration dient als Hilfsbaustein um einen **Activate_Configuration** auf der Interbuskarte durchzuführen, die mit der NETID und dem PORT adressiert wird. Durch einen **Activate_Configuration** wird die Karte in den Zustand <u>ACTIVE [• 115]</u> versetzt.

Eingänge

```
VAR_INPUT

NETID : T_AmsNetId;

PORT : T_AmsPort;

WRTRD : BOOL;

TMOUT : TIME := DEFAULT_ADS_TIMEOUT;

END_VAR
```

Name	Тур	Beschreibung
NETID	_	Hier kann die Netzwerkadresse des Rechners angegeben werden, in dem die Karte eingebaut ist. Befindet sich die Karte auf demselben System kann auch ein Leerstring angegeben werden.
PORT	T_AmsPort	Beinhaltet die ADS-Portnummer der Karte, die vom TwinCAT-System vergeben wurde.
WRTRD	BOOL	Über eine positive Flanke an diesem Wert wird der Baustein aktiviert.
TMOUT	TIME	Maximale Zeit die bei der Ausführung des Kommandos nicht überschritten werden soll.

Ausgänge

```
VAR_INPUT

BUSY : BOOL;
ERR : BOOL;
ERRID : UDINT;
RESULT : WORD;
ADDERRINFO : WORD;
```

Name	Тур	Beschreibung
BUSY	BOOL	Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
ERR	BOOL	Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der BUSY-Ausgang zurückgesetzt wurde.
ERRID	UDINT	Liefert bei einem gesetzten ERR-Ausgang die <u>ADS-</u> <u>Fehlernummer [▶ 163]</u> .
RESULT	WORD	Liefert das Ergebnis von der Karte zurück (Voraussetzung ist ein fehlerfreier ADS-Transport (ERR = FALSE)). RESULT = 0 kennzeichnet eine erfolgreiche Ausführung des Befehls. Ein Wert ungleich 0 beinhaltet die Fehlernummer der Phoenixkarte.
ADDERRORINFO	WORD	Enthält bei negativer Rückmeldung der Karte zusätzliche Fehlerinformationen (vgl. Befehlsbeschreibung der Phoenixkarte).



Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	Keine, diese Funktionalität wird zur Zeit von TwinCAT 3 nicht unterstützt!	Tc2_loFunctions (IO)

3.14.1.3 SCIT_DeactivateConfiguration

```
SCIT_DeactivateConfiguration

NETID T_AmsNetId BOOL BUSY

PORT UINT BOOL ERR

WRTRD BOOL UDINT ERRID

TMOUT TIME WORD ADDERRORINFO
```

Der Funktionsbaustein SCIT_DeactivateConfiguration dient als Hilfsbaustein um einen **Deactivate_Configuration** auf der Interbuskarte durchzuführen, die mit der NETID und dem PORT adressiert wird. Durch einen **Deactivate_Configuration** wird die Karte in den Zustand <u>PAR READY [* 115]</u> versetzt und alle Ausgänge zurückgenommen.

Eingänge

```
VAR_INPUT

NETID : T_AmsNetId;

PORT : T_AmsPort;

WRTRD : BOOL;

TMOUT : TIME := DEFAULT_ADS_TIMEOUT;

END_VAR
```

Name	Тур	Beschreibung
NETID	T_AmsNetId	Hier kann die Netzwerkadresse des Rechners angegeben werden, in dem die Karte eingebaut ist. Befindet sich die Karte auf demselben System kann auch ein Leerstring angegeben werden.
PORT	T_AmsPort	Beinhaltet die ADS-Portnummer der Karte, die vom TwinCAT-System vergeben wurde.
WRTRD	BOOL	Über eine positive Flanke an diesem Wert wird der Baustein aktiviert.
TMOUT	TIME	Maximale Zeit die bei der Ausführung des Kommandos nicht überschritten werden soll.

Ausgänge

```
VAR_INPUT

BUSY : BOOL;
ERR : BOOL;
ERRID : UDINT;
RESULT : WORD;
ADDERRINFO : WORD;
END_VAR
```

Name	Тур	Beschreibung
BUSY	BOOL	Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
ERR	BOOL	Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der BUSY-Ausgang zurückgesetzt wurde.



Name	Тур	Beschreibung
ERRID	UDINT	Liefert bei einem gesetzten ERR-Ausgang die <u>ADS-</u> <u>Fehlernummer [▶ 163]</u> .
RESULT	WORD	Liefert das Ergebnis von der Karte zurück (Voraussetzung ist ein fehlerfreier ADS-Transport (ERR = FALSE)). RESULT = 0 kennzeichnet eine erfolgreiche Ausführung des Befehls. Ein Wert ungleich 0 beinhaltet die Fehlernummer der Phoenixkarte.
ADDERRORINFO	WORD	Enthält bei negativer Rückmeldung der Karte zusätzliche Fehlerinformationen (vgl. Befehlsbeschreibung der Phoenixkarte).

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
	Keine, diese Funktionalität wird zur Zeit von TwinCAT 3 nicht unterstützt!	Tc2_loFunctions (IO)

3.14.1.4 SCIT_StartDataTransfer

```
SCIT_StartDataTransfer

NETID T_AmsNetId BOOL BUSY

PORT UINT BOOL ERR

WRTRD BOOL UDINT ERRID

TMOUT TIME WORD ADDERRORINFO
```

Der Funktionsbaustein SCIT_StartDataTransfer dient als Hilfsbaustein um einen **Start_Data_Transfer** auf der Interbuskarte durchzuführen, die mit der NETID und dem PORT adressiert wird. Durch einen **Start_Data_Transfer** wird die Karte in den Zustand <u>RUN [*15]</u> versetzt.

Eingänge

VAR_INPUT

NETID : T_AmsNetId;

PORT : T_AmsPort;

WRTRD : BOOL;

TMOUT : TIME := DEFAULT_ADS_TIMEOUT;

END_VAR

Name	Тур	Beschreibung
NETID		Hier kann die Netzwerkadresse des Rechners angegeben werden, in dem die Karte eingebaut ist. Befindet sich die Karte auf demselben System kann auch ein Leerstring angegeben werden.
PORT	T_AmsPort	Beinhaltet die ADS-Portnummer der Karte, die vom TwinCAT-System vergeben wurde.
WRTRD	BOOL	Über eine positive Flanke an diesem Wert wird der Baustein aktiviert.
TMOUT	TIME	Maximale Zeit die bei der Ausführung des Kommandos nicht überschritten werden soll.

Ausgänge

AR_INPUT	
BUSY : BOOL;	
ERR : BOOL;	



ERRID : UDINT;
RESULT : WORD;
ADDERRINFO : WORD;
END_VAR

Name	Тур	Beschreibung	
BUSY	BOOL	Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.	
ERR	BOOL	Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem de BUSY-Ausgang zurückgesetzt wurde.	
ERRID	UDINT	Liefert bei einem gesetzten ERR-Ausgang die ADS-Fehlernummer [108].	
RESULT	WORD	Liefert das Ergebnis von der Karte zurück (Voraussetzung ist ein fehlerfreier ADS-Transport (ERR = FALSE)). RESULT = 0 kennzeichnet eine erfolgreiche Ausführung des Befehls. Ein Wert ungleich 0 beinhaltet die Fehlernummer der Phoenixkarte.	
ADDERRORINFO	WORD	Enthält bei negativer Rückmeldung der Karte zusätzliche Fehlerinformationen (vgl. Befehlsbeschreibung der Phoenixkarte).	

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	Keine, diese Funktionalität wird zur Zeit von TwinCAT 3 nicht unterstützt!	Tc2_loFunctions (IO)

3.14.1.5 SCIT_StopDataTransfer

```
SCIT_StopDataTransfer

NETID T_AmsNetId BOOL BUSY

PORT UINT BOOL ERR

WRTRD BOOL UDINT ERRID

TMOUT TIME WORD ADDERRORINFO
```

Der Funktionsbaustein SCIT_StopDataTransfer dient als Hilfsbaustein um einen **Stop_Data_Transfer** auf der Interbuskarte durchzuführen, die mit der NETID und dem PORT adressiert wird. Durch einen **Stop_Data_Transfer** wird die Karte in den Zustand <u>ACTIVE [* 115]</u> versetzt, die Ausgänge werden *nicht* zurückgenommen.

Eingänge

```
VAR_INPUT

NETID : T_AmsNetId;

PORT : T_AmsPort;

WRTRD : BOOL;

TMOUT : TIME := DEFAULT_ADS_TIMEOUT;

END_VAR
```

Name	Тур	Beschreibung
NETID		Hier kann die Netzwerkadresse des Rechners angegeben werden, in dem die Karte eingebaut ist. Befindet sich die Karte auf demselben System kann auch ein Leerstring angegeben werden.



Name	Тур	Beschreibung
PORT	T_AmsPort	Beinhaltet die ADS-Portnummer der Karte, die vom TwinCAT-System vergeben wurde.
WRTRD	BOOL	Über eine positive Flanke an diesem Wert wird der Baustein aktiviert.
TMOUT	TIME	Maximale Zeit die bei der Ausführung des Kommandos nicht überschritten werden soll.

Ausgänge

VAR_INPUT

BUSY : BOOL;
ERR : BOOL;
ERRID : UDINT;
RESULT : WORD;
ADDERRINFO : WORD;

END_VAR

Name	Тур	Beschreibung	
BUSY	BOOL	Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.	
ERR	BOOL	Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem d BUSY-Ausgang zurückgesetzt wurde.	
ERRID	UDINT	Liefert bei einem gesetzten ERR-Ausgang die ADS-Fehlernummer [• 163].	
RESULT	WORD	Liefert das Ergebnis von der Karte zurück (Voraussetzung ist ein fehlerfreier ADS-Transport (ERR = FALSE)). RESULT = 0 kennzeichnet eine erfolgreiche Ausführung des Befehls. Ein Wert ungleich 0 beinhaltet die Fehlernummer der Phoenixkarte.	
ADDERRORINFO	WORD	Enthält bei negativer Rückmeldung der Karte zusätzliche Fehlerinformationen (vgl. Befehlsbeschreibung der Phoenixkarte).	

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
	Keine, diese Funktionalität wird zur Zeit von TwinCAT 3 nicht unterstützt!	Tc2_loFunctions (IO)

3.14.1.6 SCIT_AlarmStop



Der Funktionsbaustein SCIT_AlarmStop dient als Hilfsbaustein um einen **Alarm_Stop** auf der Interbuskarte durchzuführen, die mit der NETID und dem PORT adressiert wird. Durch einen **Alarm_Stop** wird die Karte in den Zustand <u>PAR_READY</u> [115] versetzt und alle Ausgänge zurückgenommen.



Eingänge

```
VAR_INPUT

NETID : T_AmsNetId;

PORT : T_AmsPort;

WRTRD : BOOL;

TMOUT : TIME := DEFAULT_ADS_TIMEOUT;

END VAR
```

Name	Тур	Beschreibung
NETID	T_AmsNetId	Hier kann die Netzwerkadresse des Rechners angegeben werden, in dem die Karte eingebaut ist. Befindet sich die Karte auf demselben System kann auch ein Leerstring angegeben werden.
PORT	T_AmsPort	Beinhaltet die ADS-Portnummer der Karte, die vom TwinCAT-System vergeben wurde.
WRTRD	BOOL	Über eine positive Flanke an diesem Wert wird der Baustein aktiviert.
TMOUT	TIME	Maximale Zeit die bei der Ausführung des Kommandos nicht überschritten werden soll.

Ausgänge

VAR INPUT

BUSY : BOOL;
ERR : BOOL;
ERRID : UDINT;
RESULT : WORD;
ADDERRINFO : WORD;

Name Beschreibung Тур **BUSY BOOL** Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung **ERR** Sollte ein Fehler bei der Übertragung des Kommandos **BOOL** erfolgen, dann wird dieser Ausgang gesetzt, nachdem der BUSY-Ausgang zurückgesetzt wurde. **ERRID UDINT** Liefert bei einem gesetzten ERR-Ausgang die ADS-<u>Fehlernummer [▶ 163]</u>. **RESULT** WORD Liefert das Ergebnis von der Karte zurück (Voraussetzung ist ein fehlerfreier ADS-Transport (ERR = FALSE)). RESULT = 0 kennzeichnet eine erfolgreiche Ausführung des Befehls. Ein Wert ungleich 0 beinhaltet die Fehlernummer der Phoenixkarte. **ADDERRORINFO** WORD Enthält bei negativer Rückmeldung der Karte zusätzliche Fehlerinformationen (vgl. Befehlsbeschreibung der Phoenixkarte).

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	Keine, diese Funktionalität wird zur Zeit von TwinCAT 3 nicht unterstützt!	Tc2_loFunctions (IO)



3.14.1.7 SCIT_ControlActiveConfiguration

```
SCIT_ControlActiveConfiguration

NETID T_AmsNetId BOOL BUSY

PORT UINT BOOL ERR

WRTRD BOOL UDINT ERRID

TMOUT TIME WORD RESULT

SWITCH_CODE WORD WORD ADDERRORINFO

DEVICE_NO WORD
```

Der Funktionsbaustein SCIT_ControlActiveConfiguration dient als Hilfsbaustein um einen **Control_Active_Configuration** auf der Interbuskarte durchzuführen, die mit der NETID und dem PORT adressiert wird. Durch einen **Control_Active_Configuration** kann der Zustand eines Teilnehmers (oder mehrerer, wenn der angegebene Teilnehmer Teil einer Gruppe ist) verändert werden.

Eingänge

```
VAR_INPUT

NETID : T_AmsNetId;

PORT : T_AmsPort;

WRTRD : BOOL;

TMOUT : TIME := DEFAULT_ADS_TIMEOUT;

SWITCH_CODE : WORD;

DEVICE_NO : WORD;

END_VAR
```

Name	Тур	Beschreibung
NETID	T_AmsNetId	Hier kann die Netzwerkadresse des Rechners angegeben werden, in dem die Karte eingebaut ist. Befindet sich die Karte auf demselben System kann auch ein Leerstring angegeben werden.
PORT	T_AmsPort	Beinhaltet die ADS-Portnummer der Karte, die vom TwinCAT-System vergeben wurde.
WRTRD	BOOL	Über eine positive Flanke an diesem Wert wird der Baustein aktiviert.
TMOUT	TIME	Maximale Zeit die bei der Ausführung des Kommandos nicht überschritten werden soll.
SWITCH_CODE	WORD	Gibt an welche Aktion mit dem Teilnehmer ausgeführt werden soll:
		0 = Segment Off 1 = Segment On 2 = Device_Off 3 = Device_On 4 = Device_Disable 5 = Device_Enable
DEVICE_NO	WORD	Gibt die Gerätenummer des angesprochenen Teilnehmers an. Für den Teilnehmer 3.1 muss z. B. ein Wert von 16#0301 angegeben werden.

Ausgänge

```
VAR_INPUT

BUSY : BOOL;
ERR : BOOL;
ERRID : UDINT;
RESULT : WORD;
ADDERRINFO : WORD;
END VAR
```



Name	Тур	Beschreibung
BUSY	BOOL	Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
ERR	BOOL	Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der BUSY-Ausgang zurückgesetzt wurde.
ERRID	UDINT	Liefert bei einem gesetzten ERR-Ausgang die <u>ADS-</u> <u>Fehlernummer [▶ 163]</u> .
RESULT	WORD	Liefert das Ergebnis von der Karte zurück (Voraussetzung ist ein fehlerfreier ADS-Transport (ERR = FALSE)). RESULT = 0 kennzeichnet eine erfolgreiche Ausführung des Befehls. Ein Wert ungleich 0 beinhaltet die Fehlernummer der Phoenixkarte.
ADDERRORINFO	WORD	Enthält bei negativer Rückmeldung der Karte zusätzliche Fehlerinformationen (vgl. Befehlsbeschreibung der Phoenixkarte).

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	Keine, diese Funktionalität wird zur Zeit von TwinCAT 3 nicht unterstützt!	Tc2_loFunctions (IO)

3.14.1.8 SCIT_GetErrorInfo

```
SCIT_GetErrorInfo

NETID T_AmsNetId BOOL BUSY
PORT UINT BOOL ERR
WRTRD BOOL UDINT ERRID
TMOUT TIME WORD RESULT
WORD ERRORCODE
WORD ADDERRORINFO
```

Der Funktionsbaustein SCIT_GetErrorInfo liest die genaue Fehlerursache und den genauen Fehlerort eines zuvor aufgetretenen Busfehlers aus der Interbuskarte, die mit der NETID und dem PORT adressiert wird.

Eingänge

VAR_INPUT

NETID : T_AmsNetId;

PORT : T_AmsPort;

WRTRD : BOOL;

TMOUT : TIME := DEFAULT_ADS_TIMEOUT;

END_VAR

Name	Тур	Beschreibung
NETID	T_AmsNetId	Hier kann die Netzwerkadresse des Rechners angegeben werden, in dem die Karte eingebaut ist. Befindet sich die Karte auf demselben System kann auch ein Leerstring angegeben werden.
PORT	T_AmsPort	Beinhaltet die ADS-Portnummer der Karte, die vom TwinCAT-System vergeben wurde.
WRTRD	BOOL	Über eine positive Flanke an diesem Wert wird der Baustein aktiviert.



Name	Тур	Beschreibung
TMOUT	TIME	Maximale Zeit die bei der Ausführung des Kommandos
		nicht überschritten werden soll.

Ausgänge

VAR_INPUT
BUSY : BOOL;
ERR : BOOL;
ERRID : UDINT;
RESULT : WORD;
ERRORCODE : WORD;
ADDERRINFO : WORD;

Name	Тур	Beschreibung
BUSY	BOOL	Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
ERR	BOOL	Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der BUSY-Ausgang zurückgesetzt wurde.
ERRID	UDINT	Liefert bei einem gesetzten ERR-Ausgang die <u>ADS-</u> <u>Fehlernummer [▶ 163]</u> .
RESULT	WORD	Liefert das Ergebnis von der Karte zurück (Voraussetzung ist ein fehlerfreier ADS-Transport (ERR = FALSE)). RESULT = 0 kennzeichnet eine erfolgreiche Ausführung des Befehls. Ein Wert ungleich 0 beinhaltet die Fehlernummer der Phoenixkarte.
ERRORCODE	WORD	Liefert Informationen zur Fehlerart (vgl. Fehlerbeschreibung der Phoenix-Karte).
ADDERRINFO	WORD	Enthält bei negativer Rückmeldung der Karte zusätzliche Fehlerinformationen (vgl. Befehlsbeschreibung der Phoenixkarte).

Voraussetzungen

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
	Keine, diese Funktionalität		Tc2_loFunctions (IO)
	wird zur Zeit von TwinCAT 3 nicht unterstützt!	SC/RI/RT-LK	

3.14.1.9 SCIT_ConfDevErrAll



Der Funktionsbaustein SCIT_ConfDevErrAll quittiert Peripheriestörungen aller vorhandenen Geräte gleichzeitig. Intern wird die **Control_Device_Function** der Interbuskarte aufgerufen. Die Interbuskarte wird mit der NETID und dem PORT adressiert.



Eingänge

```
VAR_INPUT

NETID : T_AmsNetId;

PORT : T_AmsPort;

WRTRD : BOOL;

TMOUT : TIME := DEFAULT_ADS_TIMEOUT;

END VAR
```

Name	Тур	Beschreibung
NETID	T_AmsNetId	Hier kann die Netzwerkadresse des Rechners angegeben werden, in dem die Karte eingebaut ist. Befindet sich die Karte auf demselben System kann auch ein Leerstring angegeben werden.
PORT	T_AmsPort	Beinhaltet die ADS-Portnummer der Karte, die vom TwinCAT-System vergeben wurde.
WRTRD	BOOL	Über eine positive Flanke an diesem Wert wird der Baustein aktiviert.
TMOUT	TIME	Maximale Zeit die bei der Ausführung des Kommandos nicht überschritten werden soll.

Ausgänge

VAR INPUT

BUSY : BOOL;
ERR : BOOL;
ERRID : UDINT;
RESULT : WORD;
ADDERRINFO : WORD;

Name Beschreibung Тур **BUSY BOOL** Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung **ERR** Sollte ein Fehler bei der Übertragung des Kommandos **BOOL** erfolgen, dann wird dieser Ausgang gesetzt, nachdem der BUSY-Ausgang zurückgesetzt wurde. **ERRID UDINT** Liefert bei einem gesetzten ERR-Ausgang die ADS-<u>Fehlernummer [▶ 163]</u>. **RESULT** WORD Liefert das Ergebnis von der Karte zurück (Voraussetzung ist ein fehlerfreier ADS-Transport (ERR = FALSE)). RESULT = 0 kennzeichnet eine erfolgreiche Ausführung des Befehls. Ein Wert ungleich 0 beinhaltet die Fehlernummer der Phoenixkarte. **ADDERRORINFO** WORD Enthält bei negativer Rückmeldung der Karte zusätzliche Fehlerinformationen (vgl. Befehlsbeschreibung der Phoenixkarte).

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
	Keine, diese Funktionalität wird zur Zeit von TwinCAT	Tc2_loFunctions (IO)
	3 nicht unterstützt!	



3.14.2 ads-tec

3.14.2.1 FB_ReadAdsTecSysData

```
FB_ReadAdsTecSysData

— sNetId T_AmsNetId BOOL bValid —
bEnable BOOL BError —
tCycleTime TIME UDINT nErrId —
ST_AdsTecSysData stSysData
```

Der Funktionsbaustein FB_ReadAdsTecSysData liest die Systemdaten/Diagnosedaten eines ads-tec Industrie-PCs aus. Der Baustein wird Levelgetriggert, d.h. nur beim gesetzten bEnable -Eingang werden die Systemdaten zyklisch gelesen. Um dabei die Systemauslastung niedrig zu halten wird der Lesezyklus automatisch alle ~100ms (Defaultwert) wiederholt. Bei einem gesetzten bValid-Ausgang sind die zuletzt gelesenen Daten gültig (d.h. der letzte Lesezyklus wurde fehlerfrei durchgeführt). Beim Auftreten eines Fehlers wird der bError-Ausgang gesetzt und das zyklische Lesen gestoppt. Mit einer erneuten steigenden Flanke am bEnable-Eingang können vorhandene Fehler gelöscht und das zyklische Lesen neu gestartet werden.

Eingänge

VAR_INPUT
 sNetId : T_AmsNetId;
 bEnable : BOOL;
 tCycleTime : TIME := T#100ms;
END VAR

Name	Тур	Beschreibung
sNetId	T_AmsNetID	Hier kann ein String mit der Netzwerkadresse des TwinCAT-Rechners angegeben werden, dessen Systemdaten gelesen werden sollen. Für den lokalen Rechner kann auch ein Leerstring angegeben werden.
bEnable	BOOL	Mit einer steigenden Flanke wird der Baustein zurückgesetzt (vorherige Fehler am Ausgang <i>bError</i> und <i>nErrld</i> gelöscht). Bei einem gesetzten Eingang werden die Systemdaten zyklisch gelesen.
tCycleTime	TIME	Das zyklische Leseintervall.

Ausgänge

VAR_OUTPUT
bValid : BOOL;
bError : BOOL;
nErrId : UDINT;
stSysData : ST_AdsTecSysData;
END VAR

Name	Тур	Beschreibung
bValid	BOOL	Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
bError	BOOL	Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der BUSY-Ausgang zurückgesetzt wurde.
nErrId	UDINT	Liefert bei einem gesetzten <i>bError</i> -Ausgang die <u>ADS-</u> <u>Fehlernummer [▶ 163]</u> .
stSysData	ST_AdsTecSysDat	a [▶ 139] Struktur mit den Systemdaten/Diagnosedaten.



Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	Keine, diese Funktionalität wird zur Zeit von TwinCAT 3 nicht unterstützt!	Tc2_loFunctions (IO)



4 Funktionen

4.1 [veraltete Funktionen]

4.1.1 F_GetVersionTcloFunctions

Diese Funktion ist veraltet und sollte nicht verwendet werden. Verwenden Sie bitte die globale Konstante: <u>stLibVersion Tc2 loFunctions [▶ 157]</u> um Versionsinformationen der SPS-Bibliothek auszulesen.

Diese Funktion liest Versionsinformationen der SPS-Bibliothek aus.

FUNCTION F_GetVersionTcloFunctions: UINT

Eingänge

```
VAR_INPUT
nVersionElement : INT;
END_VAR
```

Name	Тур	Beschreibung
nVersionElement	INT	Versionselement, das gelesen werden soll. Mögliche Parameter:
		• 1 : major number;
		• 2 : minor number;
		3 : revision number;

Voraussetzungen

Entwicklungsumge- bung	Zielplattform		Einzubindende SPS-Bibliotheken (Kategorie-gruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Alle IO-Geräte	Tc2_loFunctions (IO)

4.1.2 F_GetVersionRAIDController

Diese Funktion ist veraltet und sollte nicht verwendet werden. Verwenden Sie bitte die globale Konstante: <u>stLibVersion_Tc2_loFunctions [▶ 157]</u> um Versionsinformationen der SPS-Bibliothek auszulesen.

Diese Funktion liest Versionsinformationen der SPS-Bibliothek aus.

FUNCTION F_GetVersionRAIDController : UINT

Eingänge

```
VAR_INPUT nVersionElement : INT; END_VAR
```

Name	Тур	Beschreibung
nVersionElement	INT	Versionselement, das gelesen werden soll. Mögliche
		Parameter:



Name	Тур	Beschreibung
		• 1 : major number;
		• 2 : minor number;
		• 3 : revision number;

Entwicklungsumge- bung	Zielplattform		Einzubindende SPS-Bibliotheken (Kategorie-gruppe)
	PC oder CX (x86, x64, ARM)	Alle IO-Geräte	Tc2_loFunctions (IO)



5 Datentypen

5.1 E_PD_Dpv1Error

E_PD_Dpv1Error lists the DPV1-Error IDs:

```
TYPE E PD Dpv1Error :
    ePD Err ParamNumber
                                          := 0, (* Unzulässige Parameternummer *)
                                          := 1, (* Parameterwert nicht änderbar *)
    ePD Err ParamReadOnly
    ePD_Err_ValueOutOfRange
                                          := 2, (* Untere oder obere Wertgrenze überschritten *)
                                          := 3, (* Fehlerhafter Subindex *)
    ePD Err InvalidSubIndex
                                          := 4, (* Kein Array *)
    ePD Err NoArray
                                          := 5, (* Falscher Datentyp *)
:= 6, (* Kein Setzen errlaubt (nur Rücksetzen) *)
    ePD Err WrongDataType
    ePD Err OnlyResetPermitted
    ePD_Err_DescNotChangable
                                          := 7, (* Beschreibungselement nicht änderbar *)
                                          := 9, (* Beschreibungselement nicht vorhanden *)
    ePD Err DescNotFound
    ePD_Err_NoPermissionToChange := 11, (* Keine Bedienhoheit *)
                                          := 15, (* Kein Textarray vorhanden *)
    ePD_Err_NoTextArray
    ePD_Err_JobNotExecutable
                                         := 17, (* Auftrag wegen Betriebszustand nicht ausführbar *)
    ePD_Err_ValueInvalid
                                          := 20, (* Wert unzulässig *)
    ePD_Err_ResponseToLong
                                          := 21, (* Antwort zu lang *)
    ePD_Err_ParamAddrInvalid
                                        := 22, (* Parameteradresse unzulässig *)
                                         := 23, (* Format unzulässig *)
:= 24, (* Anzahl Werte nicht konsistent *)
    ePD_Err_FormatInvalid
    ePD Err ParamWrIfInitDrv
                                          := 111,
(* Schreibzugriff nur in Inbetriebnahmezustand Leistungsteil *)
    ePD_Err_ParamWrIfFastInit := 112, (* Schreibzugriff nur in Schnellinbetriebnahme *)
ePD_Err_ParamWrIfReady := 113, (* Schreibzugriff nur in Bereit *)
    ePD_Err_ParamWrIfInitParamReset := 114,
(* Schreibzugriff nur in Inbetriebnahmezustand Parameterreset *)
    ePD Err ParamWrIfInitSafety
                                         := 115,
(* Schreibzugriff nur in Inbetriebnahmezustand Safety *)
    ePD Err ParamWrIfInitTechApp := 116,
(* Schreibzugriff nur in Inbetriebnahmezustand Tech.Appl./Einheiten *)
    eru_Err_ParamWrIfInit := 117,(* Schreibzugriff nur in Inbetriebnahmezustand *)
ePD_Err_ParamWrIfInitDwnLd := 118.
(* Schreibzugriff nur in Inbetriebnahmezustand Download *)
    ePD_Err_ParamNoWrtIfDwnLd := 119,(* Darf im Download nicht geschrieben werden *)
ePD_Err_ParamWrIfInitDrvCfg := 120,
    ePD Err ParamWrIfInitDrvCfg
(* Schreibzugriff nur in Inbetriebnahmezustand Antriebskonfiguration *)
    ePD Err ParamWrIfInitSetDrvType := 121,
(* Schreibzugriff nur in Inbetriebnahmezustand Festlegung Antriebstyp *)
    ePD Err ParamWrIfInitDatasetCfg := 122,
(* Schreibzugriff nur in Inbetriebnahmezustand Datensatz-Basiskonfiguration *)
    ePD Err ParamWrIfInitDevCfg
                                      := 123,
(* Schreibzugriff nur in Inbetriebnahmezustand Gerätekonfiguration *)
    ePD Err ParamWrIfInitDevDwnLd := 124,
(* Schreibzugriff nur in Inbetriebnahmezustand Gerätedownload *)
    ePD Err ParamWrIfInitDevPrmReset := 125,
(* Schreibzugriff nur in Inbetriebnahmezustand Geräteparameterreset *)
    ePD Err ParamWrIfInitDevReady
                                      := 126,
(* Schreibzugriff nur in Inbetriebnahmezustand Gerät bereit *)
    ePD_Err_ParamWrIfInitDevice := 127,(* Schreibzugriff nur in Inbetriebnahmezustand Gerät *)
    ePD Err ParamNoWriteIfDwnLd
                                          := 129, (* darf im Download nicht geschrieben werden *)
                                      := 129,(* daff im Downtoad Hight geschifteden worden /
:= 130,(* Übernahme der Steuerungshoheit über BICO gesperrt *)
:= 131,(* gewünschte BICO-Verschaltung unmöglich *)
:= 132,(* Parameteränderung gesperrt *)
    ePD Err CtrlTakeOverBlocked
    ePD_Err_ParamBicoSetInvalid
ePD_Err_ParamChangeBlocked
    ePD_Err_BaramNoAccessDefined := 133, (* Keine Zugriffsmethode definiert *)
ePD_Err_BelowDefinedMinimum := 200, (* Unterhalb aktuell gültiger Grenze *)
ePD_Err_AboveDefinedMaximum := 201, (* Oberhalb aktuell gültiger Grenze *)
                                          := 200, (* Unterhalb aktuell gültiger Grenze *)
    ePD Err WriteNotPermitted
                                          := 204(* Schreiben nicht erlaubt *)
END TYPE
```



Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Beckhoff FC310x PCI, CX1500-M310, EL6731, EL6632	Tc2_loFunctions (IO)

5.2 E_BatteryStatus

Batteriestatus.

```
TYPE E_BatteryStatus :
    (
        BatteryUnknownStatus,
        BatteryOk,
        BatteryReplace
);
END_TYPE
```

Name	Wert	Bedeutung
BatteryUnknownStatus	0	Der Akkustatus ist unbekannt.
BatteryOk	1	Der Akkustatus ist OK.
BatteryReplace	2	Der Akku soll ausgewechselt werden.

Voraussetzungen

Entwicklungsum- gebung	Zielplattform	US	SV Hardware	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	•	Beckhoff CU81x0-0xx0	Tc2_loFunctions
		•	Beckhoff BAPI v1;	(IO)
		•	Beckhoff P24Vxxxx;	
		•	Beckhoff CP903x-Karte (PCI/ISA);	
			Beckhoff CX2100-09x4 Modelle (z.B. CX2100-0904 oder CX2100-0914 + "Smart Battery" CX2900-0192);	
			Die mit Beckhoff Industrie-PC ausgelieferten APC-Geräte die das Smartprotokoll unterstützen und mit dem Windows USV- Dienst konfiguriert werden können;	

5.3 E_PD_Datatype

E_PD_Datatype enthält die möglichen Datentypen eines Profidrive-Parameters.



```
ePD_FLOAT := 8, (* IEEE 754 *)

ePD_VSTRING := 9, (* ISO/IEC 646, variable length *)

ePD_OCTSTRING := 10, (* bytearray, variable length *)

ePD_TIMEOFDAY_WDI := 12, (* 6 Bytes: 4 bytes ms + 2 bytes day since 1.1.1984 *)

ePD_TIMEDIFF := 13, (* 4 | 6 Bytes: 4 bytes ms + optional 2 bytes days*)

ePD_N2_16BIT := 33,

ePD_N4_32BIT := 34,

ePD_V2_BITSEQ := 35,

ePD_L2_NIBBLE := 36,

ePD_L2_NIBBLE := 36,

ePD_R2_PECIP_TC := 37
       ePD_R2_RECIP_TC := 37,
ePD_T2_TC_16BIT := 38,
ePD_T2_TC_32BIT := 39,
ePD_D2_TC := 40,
ePD_E2_FIXPT_16 := 41,
       ePD_C2_FIXPT_32 := 42,
       ePD_X2_NV_16 := 43,
ePD_X4_NV_32 := 44,
ePD_DATE := 50,
(* 7 Bytes: 2 bytes ms + 2 bits (res.), 6 bits(minutes) + 1 bit (0: StdTime/
1:DaylightSavingTime), 2 bits (res.), 5 bits(hours) + 3 bits (DayOfWeek), 5 bits(DayOfMonth) + 2 bit
ePD_TIMEDIFF_WDI := 53,(* 6 Bytes: 4 bytes ms + 2 bytes days *)
       ePD TIMEDIFF NODI:= 54, (* 0 .. 4294967295 ms *)
                              := 64,
       ePD ZERO
                                     := 65,
:= 66,
       ePD_BYTE
       ePD_DWORD
                                    := 67,
                                      := 68
);
END_TYPE
```

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Beckhoff FC310x PCI, CX1500-M310, EL6731, EL6632	Tc2_loFunctions (IO)

5.4 E_RAIDDriveStatus

Name	Wert	Bedeutung
eRAID_DRIVE_STATUS_OK	0	Zeigt an, dass der physische Antrieb betriebsbereit ist.
eRAID_DRIVE_STATUS_DEGRADED	1	Zeigt an, dass der physische Antrieb eine SMART- Meldung an den Controller geschickt hat.
eRAID_DRIVE_STATUS_REBUILDING	2	Zeigt an, dass der physische Antrieb der Zielantrieb eines RAID-Set-Rebuilds ist. Wenn der Rebuild erfolgreich absolviert ist, ändert sich der Status zu eRAID_DRIVE_STATUS_OK . Wenn der Rebuild fehlschlägt, wird der Status entsprechend upgedated.
eRAID_DRIVE_STATUS_FAILED	3	Zeigt an, dass der physische Antrieb nicht behebbare Fehler an den Controller gemeldet hat oder der Antrieb hat eine Anbieter-spezifische Aktion gestartet, um den Antrieb aus dem RAID-Set zu entfernen. Es gibt keine Garantie für die Betriebsbereitschaft des Antriebs und Datenverlust ist aufgetreten oder droht.



Entwicklungsumgebung	•	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_loFunctions (IO)

5.5 E_RAIDDriveUsage

```
TYPE E_RAIDDriveUsage :
(
    eRAID_DRIVE_CONFIG_NOT_USED := 0,
    eRAID_DRIVE_CONFIG_MEMBER := 1,
    eRAID_DRIVE_CONFIG_SPARE := 2
);
END_TYPE
```

Name	Wert	Bedeutung
eRAID_DRIVE_CONFIG_NOT_USED	0	Zeigt an, dass der physische Antrieb nicht Teil eines RAID-Sets ist.
eRAID_DRIVE_CONFIG_MEMBER	1	Zeigt an, dass der physische Antrieb Teil eines RAID- Sets ist.
eRAID_DRIVE_CONFIG_SPARE	2	Zeigt an, dass der physische Antrieb als "hot swap spare" Teil dieses RAID-Sets ist.

"hot swap spare" -> Fällt ein Laufwerk innerhalb des RAID-Verbundes aus, wird es im laufenden Betrieb (hot swap) durch das Reservelaufwerk (hot spare) ersetzt. Dadurch ist die Redundanz schnellstmöglich wiederhergestellt.

Voraussetzungen

Entwicklungsumgebung	•	Einzubindende SPS-Bibliotheken (Kategoriegruppe)	
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_loFunctions (IO)	

5.6 E_RAIDStatus

Name	Wert	Bedeutung
eRAID_SET_STATUS_OK	0	Zeigt an, dass das RAID-Set betriebsbereit ist.
eRAID_SET_STATUS_DEGRADED	1	Zeigt an, dass das RAID-Set nicht mehr in einem fehlertoleranten Modus arbeitet.
eRAID_SET_STATUS_REBUILDING	2	Zeigt an, dass das RAID-Set im Rebuild ist. Dies bedeutet einen eingeschränkten Betrieb. Wenn der Rebuild erfolgreich abgeschlossen ist, ändert sich der Status zu eRAID_SET_STATUS_OK . Wenn der Rebuild fehlschlägt, wird der Status entsprechend upgedated.
eRAID_SET_STATUS_FAILED	3	Zeigt an, dass das RAID-Set fehlgeschlagen ist. Es gibt keine Garantie für die Betriebsbereitschaft des RAID-Sets und Datenverlust ist aufgetreten oder droht.



Entwicklungsumgebung	•	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_loFunctions (IO)

5.7 E_RAIDType

```
TYPE E_RAIDType :
(
    eRAID_TYPE_NONE := 0,
    eRAID_TYPE_0 := 1,
    eRAID_TYPE_1 := 2,
    eRAID_TYPE_10 := 3,
    eRAID_TYPE_5 := 4,
    eRAID_TYPE_5 := 5,
    eRAID_TYPE_OTHER := 255
);
END_TYPE
```

Name	Wert	Bedeutung
eRAID_TYPE_NONE	0	Zeigt an, dass das RAID-Set aus einem einzigen Antrieb besteht. Kein Set mit der angegebenen Nummer existiert.
eRAID_TYPE_0	1	Zeigt an, dass das RAID-Set ein gestriptes Set ist ohne Fehlertoleranz.
eRAID_TYPE_1	2	Zeigt an, dass das RAID-Set ein gemirrortes Set ist.
eRAID_TYPE_10	3	Zeigt an, dass das RAID-Set ein gestriptes und gemirrortes Set ist.
eRAID_TYPE_5	4	Zeigt an, dass das RAID-Set ein Parity-Set ist.
eRAID_TYPE_15	5	Zeigt an, dass das RAID-Set ein Advanced-Parity-Set ist.
eRAID_TYPE_OTHER	255	Zeigt an, dass die Konfiguration des RAID-Sets nicht den Standardtypen entspricht.

Voraussetzungen

Entwicklungsumgebung	•	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_loFunctions (IO)

5.8 E_SercosAttribLen

```
TYPE E_SercosAttribLen : (
   eLEN_2BYTE := 1, (* 2 bytes, fixed length *)
   eLEN_4BYTE := 2, (* 4 bytes, fixed length *)
   eLEN_8BYTE := 3, (* 8 bytes, fixed length *)
   eLEN_VIBYTE := 4, (* 1 bytes, variable length *)
   eLEN_V2BYTE := 5, (* 2 bytes, variable length *)
   eLEN_V4BYTE := 6, (* 4 bytes, variable length *)
   eLEN_V8BYTE := 7 (* 8 bytes, variable length *)
);
END_TYPE
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	,	Sercans SCS-P ISA; Sercans SCS-P PCI; Beckhoff FC750x PCI	Tc2_loFunctions (IO)



5.9 E_SercosAttribType

Voraussetzungen

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0		Sercans SCS-P ISA; Sercans SCS-P PCI; Beckhoff FC750x PCI	Tc2_loFunctions (IO)

5.10 E_UpsCommStatus

Kommunikationsstatus zur USV-Hardware.

```
TYPE E_UpsCommStatus :
   (
     UpsCommUnknownStatus,
     UpsCommOk,
     UpsCommFailed
);
END_TYPE
```

Name	Wert	Bedeutung
UpsCommUnknownStatus	0	Der Kommunikationsstatus ist unbekannt.
UpsCommOk	1	Die Kommunikation zur USV ist hergestellt.
UpsCommFailed	2	Die Kommunikation zur USV wurde unterbrochen.

Voraussetzungen

Entwicklungsum- gebung	Zielplattform	USV Hardware	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Beckhoff CU81x0-0xx0	Tc2_loFunctions
		Beckhoff BAPI v1;	(IO)
		Beckhoff P24Vxxxx;	
		Beckhoff CP903x-Karte (PCI/ ISA);	
		 Beckhoff CX2100-09x4 Modelle (z.B. CX2100-0904 oder CX2100-0914 + "Smart Battery" CX2900-0192); 	
		 Die mit Beckhoff Industrie-PC ausgelieferten APC-Geräte die das Smartprotokoll unterstützen und mit dem Windows USV- Dienst konfiguriert werden können; 	



5.11 E_UpsPowerStatus

Status der Spannungsversorung.

```
TYPE E_UpsPowerStatus :
    (
        PowerUnknownStatus,
        PowerOnLine,
        PowerOnBattery
);
END_TYPE
```

Name	Wert	Bedeutung
PowerUnknownStatus	0	Der Status der Spannungsversorgung ist unbekannt
PowerOnLine	1	Netzspannungsversorgung.
PowerOnBattery	2	Akkuspannungsversorgung.

Voraussetzungen

Entwicklungsum- gebung	Zielplattform	USV Hardware	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Beckhoff CU81x0-0xx0	Tc2_loFunctions
		Beckhoff BAPI v1;	(IO)
		Beckhoff P24Vxxxx;	
		Beckhoff CP903x-Karte (PCI/ ISA);	
		Beckhoff CX2100-09x4 Modelle (z.B. CX2100-0904 oder CX2100-0914 + "Smart Battery" CX2900-0192);	
		Die mit Beckhoff Industrie-PC ausgelieferten APC-Geräte die das Smartprotokoll unterstützen und mit dem Windows USV- Dienst konfiguriert werden können;	

5.12 IODEVICETYPES

```
TYPE IODEVICETYPES:
(
IODEVICETYPE_UNKNOWN
CORVICETYPE_C1220
                                                     := 0,(* Unknown device *)
                                                        := 1, (* Beckhoff Lightbus-Master *)
IODEVICETYPE C1200
                                                       := 2, (* Beckhoff Lightbus-Master *)
IODEVICETYPE_SPC3
IODEVICETYPE CIF30DPM
                                                      := 3, (* ProfiBus Slave (Siemens) *)
IODEVICETYPE_CIF30DPM := 4,(* ISA ProfiBus-Master 2 kByte (Hilscher) *)
IODEVICETYPE_CIF40IBSM := 5,(* ISA Interbus-S-Master 2 kByte (Hilscher) *)
IODEVICETYPE_BKHFPC := 6,(* Beckhoff PC C2001*)
IODEVICETYPE_CP5412A2 := 7,(* ProfiBus-Master (Siemens)*)
IODEVICETYPE_SERCANSISA := 8,(* Sercos Master (Indramat)*)
IODEVICETYPE_LPTPORT := 9 (* Int Port*)
                                                       := 4, (* ISA ProfiBus-Master 2 kByte (Hilscher) *)
                                                     := 8, (* Sercos Master (Indramat)*)
:= 9, (* Lpt Port*)
IODEVICETYPE_LPTPORT
                                                      := 10, (* Generic DPRAM*)
IODEVICETYPE DPRAM
IODEVICETYPE_Drags.

IODEVICETYPE_COMPORT := 11,(* COM role,
IODEVICETYPE_CIF30CAN := 12,(* ISA CANopen-Master (Hilscher)*)
IODEVICETYPE_CIF30PB := 13,(* ISA ProfiBus-Master 8 kByte (Hilscher)*)
IODEVICETYPE_BKHFCP2030 := 14,(* Beckhoff CP2030 (Pannel-Link)*)
IODEVICETYPE_IBSSCIT := 15,(* Interbus-S-Master (Phoenix)*)

- 16.(* ISA Interbus-S-Master (Hilscher)*)
                                                     := 16,(* ISA Interbus-S-Master (Hilscher)*)
:= 17,(* ISA DeviceNet-Master (Hilscher)*)
IODEVICETYPE_CIF30DNM
IODEVICETYPE_FCXXXX
IODEVICETYPE_CIF50PB
                                                      := 18, (* Beckhoff-Fieldbus card *)
                                                      := 19,(* PCI ProfiBus-Master 8 kByte (Hilscher)*)
:= 20,(* PCI Interbus-S-Master (Hilscher)*)
IODEVICETYPE CIF50IBM
                                                      := 21, (* PCI DeviceNet-Master (Hilscher)*)
IODEVICETYPE_CIF50DNM
IODEVICETYPE CIF50CAN
                                                        := 22, (* PCI CANopen-Master (Hilscher)*)
                                               := 23, (* PCMCIA ProfiBus-Master (Hilscher)*)
IODEVICETYPE CIF60PB
```



```
IODEVICETYPE_CIF60DNM := 24,(* PCMCIA DeviceNet-Master (Hilscher)*)
IODEVICETYPE_CIF60CAN := 25,(* PCMCIA CANopen-Master (Hilscher)*)
IODEVICETYPE_CIF104DP := 26,(* PC104 ProfiBus-Master 2 kByte (Hilscher)*)
                                                                            := 26, (* PC104 ProfiBus-Master 2 kByte (Hilscher)*)
:= 27, (* PC104 ProfiBus-Master 8 kByte (Hilscher)*)
 IODEVICETYPE_CIF104DP
 IODEVICETYPE C104PB
                                                                       := 27, (* PC104 Profibus-Master o Abyte (Hilscher), 
:= 28, (* PC104 Interbus-S-Master 2 kByte (Hilscher)*)
:= 29, (* PC104 CANopen-Master (Hilscher)*)
:= 30, (* PC104 DeviceNet-Master (Hilscher)*)
 IODEVICETYPE_C104IBM
 IODEVICETYPE C104CAN
                                                                  := 30,(* PC104 DeviceNet-Master (Hilscher)*)
:= 31,(* Beckhoff CP9030 (Pannel-Link with UPS)*)
:= 32,(* Motherboard System Management Bus*)
 IODEVICETYPE C104DNM
 IODEVICETYPE BKHFCP9030
 IODEVICETYPE SMB
                                                                             := 33, (* Beckhoff-PROFIBUS-Monitor*)
 IODEVICETYPE PBMON
                                                                 := 33,(    Becknoff-PROFIBUS-Monitor*)
:= 34,(* PCI ProfiBus-Master (Siemens)*)
:= 35,(* PCMCIA Interbus-S-Master (Hilscher)*)
:= 36,(* Beckhoff-Lightbus-I/II-PCI-Karte*)
:= 37,(* obsolete: dont use*)
:= 38,(* Beckhoff-Profibus-PCI*)
:= 39,(* Beckhoff-CanOpen-PCI*)
:= 41.(* Beckhoff-DeviceNet-PCI*)
 IODEVICETYPE_CP5613
IODEVICETYPE_CIF60IBM
 IODEVICETYPE_FC200X
 IODEVICETYPE_FC3100_OLD
 IODEVICETYPE_FC3100
                                                                            := 39, (* Beckhoff-CanOpen-PCI*)
:= 41, (* Beckhoff-DeviceNet-PCI*)
 IODEVICETYPE FC5100
IODEVICETYPE FC5200
                                                                := 57,(* Virtual USB Device *)
:= 58,(* Beckhoff-CANopen-Monitor-PCI *)
:= 59,(* Beckhoff-DeviceNet-Monitor-PCI *)
:= 60,(* Beckhoff-Profibus-PCI Slave *)
  IODEVICETYPE USB
 IODEVICETYPE_FC5100MON
 IODEVICETYPE_FC5200MON
IODEVICETYPE_FC3100SLV
                                                                             := 61, (* Beckhoff-CanOpen-PCI Slave *)
 IODEVICETYPE_FC5100SLV
IODEVICETYPE FC5200SLV
 IODEVICETYPE IBSSCITPCI
                                                                             := 62,(* Beckhoff-DeviceNet-PCI Slave *)
:= 63,(* PCI Interbus-S-Master (Phoenix) *)
 IODEVICETYPE_IBSSCRIRTLKPCI
                                                                              := 64, (* PCI Interbus-S-Master with Slave-
Module LWL Basis (Phoenix) *)

IODEVICETYPE_CX1100_BK := 65,(* Beckhoff-CX1100 terminal bus possible to the process of the pro
 Module LWL Basis (Phoenix) *)
                                                                             := 65, (* Beckhoff-CX1100 terminal bus power supply *)
                                                                            := 74, (* PC104 DeviceNet-Slave *)
:= 75, (* PC104 Sercos-Master *)
 IODEVICETYPE_CX1500_M750
                                                                            := 76,(* PC104 Sercos-Slave *)
:= 77,(* BX terminal bus interface *)
 IODEVICETYPE_CX1500_B750
 IODEVICETYPE BX BK
                                                                              := 78, (* BX SSB-Master *)
 IODEVICETYPE_BX_M510
 IODEVICETYPE BX_B310 := 79,(* BX ProfiBus-Slave *)
IODEVICETYPE_IBSSCRIRTPCI := 80,(* PCI Interbus-S-
 Master with slave module copper basis (Phoenix) *)
 IODEVICETYPE BX B510 := 81,(* BX CANopen Slave *)
IODEVICETYPE BX B520 := 82,(* BX DeviceNet Slave *)
                                                                            := 83, (* BCxx50 ProfiBus Slave *)
:= 84, (* BCxx50 CANopen Slave *)
 IODEVICETYPE BC3150
 IODEVICETYPE BC5150
                                                                            := 84,( BCXX50 CANOPEN STAVE *)
:= 85,(* BCXX50 DeviceNet Slave *)
:= 86,(* Beckhoff Profibus-EtherCAT Terminal *)
:= 87,(* Beckhoff CanOpen-EtherCAT Terminal *)
 IODEVICETYPE BC5250
  IODEVICETYPE EL6731
 IODEVICETYPE EL6751
                                                                            := 88, (* Beckhoff DeviceNet-EtherCAT Terminal *)
:= 89, (* COM ProfiBus Master 8 kByte (Hilscher) *)
 IODEVICETYPE_EL6752
 IODEVICETYPE COMPB
                                                                            := 90, (* COM Interbus-S Master (Hilscher) *)
  IODEVICETYPE COMIBM
                                                                             := 91, (* COM DeviceNet Master (Hilscher)
:= 92, (* COM CANopen Master (Hilscher) *)
 IODEVICETYPE_COMDNM
IODEVICETYPE COMCAN
                                                                           := 93, (* COM CANopen Slave (Hilscher) *)
 IODEVICETYPE_COMIBS
 IODEVICETYPE_ETHERCAT
                                                                              := 94, (* EtherCAT in direct mode *)
 IODEVICETYPE PROFINETIOCONTROLLER:= 95, (* PROFINET Master *)
 IODEVICETYPE_PROFINETIOCONTROLLER:= 95, (* PROFINET Master *)

IODEVICETYPE_PROFINETIODEVICE := 96, (* PROFINET Slave *)

IODEVICETYPE_EL6731SLV := 97, (* Beckhoff Profibus Slave EtherCAT Terminal *)

IODEVICETYPE_EL6751SLV := 98, (* Beckhoff CanOpen Slave EtherCAT Terminal *)

IODEVICETYPE_EL6752SLV := 99, (* Beckhoff DeviceNet Slave EtherCAT Terminal *)

IODEVICETYPE_C104PPB := 100, (* PC104+ Profibus Master 8 kByte (Hilscher) *)
                                                                             := 101,(* PC104+ CANopen Master (Hilscher) *)
:= 102,(* PC104+ DeviceNet Master (Hilscher) *)
 IODEVICETYPE_C104PCAN
IODEVICETYPE_C104PDNM
                                                                   := 103, (* BCxx50 Serial Slave *)
 IODEVICETYPE BC8150
```



```
:= 104, (* BX9000 Ethernet Slave *)
IODEVICETYPE BX9000
IODEVICETYPE CX9000 BK
                                        := 105,(* Beckhoff-CX9000 K-Bus Power Supply *)
                                          := 106, (* Beckhoff-RT-Ethernet-EtherCAT-Terminal *)
:= 107, (* BC9050 Ethernet Slave *)
IODEVICETYPE EL6601
IODEVICETYPE BC9050
                                          := 108, (* BC9120 Ethernet Slave *)
:= 109, (* Ethernet Miniport Adapter *)
IODEVICETYPE BC9120
IODEVICETYPE ENETADAPTER
                                          := 110, (* BC9020 Ethernet Slave *)
IODEVICETYPE BC9020
                                          := 111, (* EtherCAT Protocol in direct mode *)
:= 112, (* *)
IODEVICETYPE_ETHERCATPROT
IODEVICETYPE_ETHERNETNVPROT
IODEVICETYPE ETHERNETPNMPROT := 113,(* Profinet Controller *)
IODEVICETYPE_EL6720 := 114,(* Beckhoff-Lightbus-EtherCAT-Terminal *)
IODEVICETYPE_ETHERNETPNSPROT := 115,(* Profinet Device*)
IODEVICETYPE_BKHFCP6608 := 116, (* Beckhoff CP6608(IXP PC) *)
IODEVICETYPE_PTP_IEEE1588 := 117, (* *)
                                          := 118, (* EL6631-0010 Profinet Slave terminal *)
IODEVICETYPE EL6631SLV
                                          := 119,(* EL6631 Profinet Master terminal *)
:= 120,(* Beckhoff-CX5100 K-Bus power supply *)
IODEVICETYPE EL6631
IODEVICETYPE_CX5000_BK
IODEVICETYPE PCIDEVICE := 121, (* Generic PCI DPRAM (TCOM) *)
IODEVICETYPE ETHERNETUPDPROT := 122, (* UDP Protocol *)
IODEVICETYPE ETHERNETAUTOPROT := 123, (* Automation Protocol *)
                                        := 124,(* CCAT *)
:= 125,(* Virtuell USB Device (remote via CPLINK3) *)
IODEVICETYPE_CCAT
IODEVICETYPE_CPLINK3
                                          := 126, (* EL6632 *)
IODEVICETYPE EL6632
                                          := 127, (* CCAT Profibus Master *)
:= 128, (* CCAT Profibus Slave *)
IODEVICETYPE CCAT PBM
IODEVICETYPE CCAT PBS
                                         := 129, (* CCAT CANopen Master *)
:= 130, (* EtherCAT Slave *)
:= 131, (* BACnet device *)
IODEVICETYPE_CCAT_CNM
IODEVICETYPE ETHERCATSLAVE
IODEVICETYPE_BACNET
                                          := 132, (* CCAT CANopen Slave *)
:= 133, (* ETHERNET IP Master *)
:= 134, (* ETHERNET IP Slave (OLD) *)
IODEVICETYPE_CCAT_CNS
IODEVICETYPE_ETHIP_SCANNER
IODEVICETYPE_ETHIP_ADAPTER
IODEVICETYPE CX8000 BK
                                           := 135, (* Beckhoff-CX8100 K-Bus Power Supply -
 LEGACY use IODEVICETYPE CX BK *)
IODEVICETYPE_ETHERNETUDPPROT := 136,(* Upd Protocol *)
IODEVICETYPE_BC9191 := 137,(* BC9191 Ethernet Slave *)
                                          := 138,(* Real-Time Ethernet Protocol (BK90xx, AX2000-B900) *)
IODEVICETYPE ENETPROTOCOL
                                          := 139, (* ETHERNET IP Slave (NEW) *)
:= 140, (* Profinet Controller CCAT RT *)
IODEVICETYPE ETHIP ADAPTEREX
IODEVICETYPE_PNCONTR_CCAT RT
IODEVICETYPE PNCONTR CCAT IRT := 141, (* Profinet Controller CCAT RT + IRT *)
IODEVICETYPE PNDEV_CCAT RT := 142, (* Profinet Device CCAT RT *)
IODEVICETYPE_PNDEV_CCAT_IRT := 143, (* Profinet Device CCAT RT + IRT *)
IODEVICETYPE_ETHERCATSIMULATION := 144,(* EtherCAT-Simulation *)
IODEVICETYPE_EL6652SLV := 145,(* EL6652-0010 *)
                                          := 146, (* PTP CLock via CCAT *)
IODEVICETYPE_PTP_VIA_CCAT
                                           := 147, (* BACnet Rev9 device *)
IODEVICETYPE BACNETR9
IODEVICETYPE ETHERCATXFC
                                           := 148, (* EtherCAT in xfc mode *)
IODEVICETYPE_CX2500_0030
IODEVICETYPE_CX2500_0031
                                          := 149, (* CX2500-0030 RS232 Serial Communication Port *)
                                          := 150, (* CX2500-0031 RS422/RS485 Serial Communication Port *)
IODEVICETYPE EL6652MST
                                          := 151, (* EL6652 *)
(* Reserved for new devices*)
IODEVICETYPE MAX
END TYPE
```

Entwicklungsumge- bung	Zielplattform		Einzubindende SPS-Bibliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Alle IO-Geräte	Tc2_loFunctions (IO)

5.13 ST_AdsTecSysData

```
TYPE ST_AdsTecSysData

STRUCT

bShiftKey : BOOL; (* TRUE == Shift key pressed*)

bRMouseKey : BOOL; (* TRUE == Right mouse key pressed *)

bHotKey : BOOL; (* TRUE == Hotkey pressed *)

bTaskChaKey : BOOL; (* TRUE == Task change key pressed *)

bABCKey : BOOL; (* TRUE == ABC soft keyboard key pressed*)

bRsrv1 : BOOL;

bRsrv2 : BOOL;
```



```
bRsrv3 : BOOL;
bMainFanErr : BOOL; (* TRUE == Main fan error*)
bCpuFanErr : BOOL; (* TRUE == CPU fan error*)
bTempErr : BOOL; (* TRUE == Internal temperature error ( temp > 50°C)*)
bBatteryErr : BOOL; (* TRUE == Battery error *)
bRsrv4 : BOOL;
bRsrv5 : BOOL;
bRsrv6 : BOOL;
bRsrv7 : BOOL;
nMainNtcTemp : SINT; (* Main NTC temperature (-127°C .. + 127°C) *)
nExtNtcTemp : SINT; (* External NTC temperature (-127°C .. + 127°C) *)
nRsrv8 : ARRAY[1..12] OF BYTE;
END_STRUCT
END_TYPE
```

Name	Тур	Beschreibung
bShiftKey	BOOL	"Shift"-Taste gedrückt (Taste ganz rechts in der Front)
bRMouseKey	BOOL	"Rechte Maus"-Taste gedrückt
bHotKey	BOOL	"Hotkey"-Taste gedrückt
bTaskChaKey	BOOL	"Taskwechsel"-Taste gedrückt
bABCKey	BOOL	"ABC Softkeyboard"-Taste gedrückt
bMainFanErr	BOOL	Fehler Hauptlüfter
bCpuFanErr	BOOL	Fehler CPU-Lüfter
bTempErr	BOOL	Temperaturfehler (Innentemperatur > 50°C)
bBatteryErr	BOOL	Batteriefehler (derzeit reserviert)
nMainNtcTemp	SINT	Temperaturwert 1 (eingelöteter NTC-127°C + 127°C)
nExtNtcTemp	SINT	Temperaturwert 2 (anschließbarer NTC, nicht bei jedem Gerät vorhanden)
bRsrv1 - bRsrv7	ARRAY OF BYTE	Reserviert
nRsrv8	ARRAY OF BYTE	Reserviert

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
	Keine, diese Funktionalität wird zur Zeit von TwinCAT 3 nicht unterstützt!	Tc2_loFunctions (IO)

5.14 ST_Dpv1ParamAddrEx

ST_Dpv1ParamAddrEx enthält die Daten eines Profidrive-Parameters.

```
TYPE ST_DpvlParamAddrEx:

STRUCT

iAttribute : USINT;
iNumOfElements : USINT;
iParameterNumber : UINT;
iSubIndex : UINT;
iDataAddr : PVOID;
iDataSize : UDINT;
eFormat : E_PD_Datatype;
iNumOfValues : UINT;
iErrorValue : UDINT;
stError : ST_PD_DpvlError;
END_STRUCT
END_TYPE
```

Name	Тур	Beschreibung
iAttribute	USINT	0x10: Wert; 0x20: Beschreibung; 0x30: Text; 0x80F0: herstellerspezifisch; andere Werte sind reserviert.
iNumOfElements	USINT	1234: Anzahl der Elemente; 0: Spezialfunktionen; andere Werte sind reserviert



Name	Тур	Beschreibung
iParameterNumber	UINT	165535: Parameternummer; 0: reserviert
iSubIndex	UINT	065535: Unterindex
iDataAddr	PVOID	Adresse des Puffers/Adresse der SPS-Variablen.
iDataSize	UDINT	Größe des Puffers/Größe der SPS-Variablen.
eFormat	E_PD_Datatype	0x010x36: Datentyp; 0x40: ZERO; 0x41: BYTE; 0x42: WORD; 0x43: DWORD; 0x44: Fehler; andere Werte sind reserviert.
iNumOfValues	UINT	0234: Anzahl der Werte; andere Werte sind reserviert
iErrorValue	UDINT	DPV1 Fehlerwert
stError	ST_PD_Dpv1Error	DPV1 Fehlerflag, DPV1 Fehleraufzählungstyp

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0		Beckhoff FC310x PCI, CX1500-M310, EL6731, EL6632	Tc2_loFunctions (IO)

5.15 ST_Dpv1ValueHeaderEx

ST_Dpv1ValueHeaderEx enthält die Daten eines Parameters im DPV1 Telegramm und seine String-Repräsentation.

```
TYPE ST_Dpv1ValueHeaderEx :

STRUCT

eFormat : E_PD_Datatype;
iNumOfValues : USINT;
iOffset : USINT;
iDataLen : UINT;
strData : STRING;

END_STRUCT
END_TYPE
```

Name	Тур	Beschreibung
eFormat	E_PD_Datatype	0x010x36: Datentyp; 0x40: ZERO; 0x41: BYTE; 0x42: WORD; 0x43: DWORD; 0x44: Fehler; andere Werte sind reserviert.
iNumOfValues	USINT	0234: Anzahl der Werte; andere Werte sind reserviert.
iOffset	USINT	Offset im DPV1 Antworttelegramm
iDataLen	UINT	Datenlänge
strData	STRING	Daten als STRING

Voraussetzungen

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Beckhoff FC310x PCI, CX1500-M310, EL6731, EL6632	Tc2_loFunctions (IO)



5.16 ST_NovRamAddrInfo

```
TYPE ST_NovRamAddrInfo:
STRUCT

pCardAddress : POINTER TO DWORD;
iCardMemSize : UDINT;
END_STRUCT
END TYPE
```

Name	Тур	Beschreibung
pCardAddress	POINTER TO DWORD	Der Address pointer vom NOV/DP-RAM
iCardMemSize	UDINT	Die Konfigurierte NOV/DP-RAM-Größe in Bytes.

Voraussetzungen

Entwicklungsumge- bung	Zielplattform		Einzubindende SPS-Bibliotheken (Kategorie-gruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Alle IO-Geräte	Tc2_loFunctions (IO)

5.17 ST_NovRamAddrInfoEx

```
TYPE ST_NovRamAddrInfoEx:

STRUCT

pCardAddress : POINTER TO DWORD;

iCardMemSize : UDINT;

eAccessType : E_IOACCESSTYPE

END_STRUCT

END_TYPE
```

Name	Тур	Beschreibung
pCardAddress	POINTER TO DWORD	Der Address pointer vom NOV/DP-RAM
iCardMemSize	UDINT	Die Konfigurierte NOV/DP-RAM-Größe in Bytes.
eAccessType	E_IOACCESSTYPE	Die Zugriffsart auf NOV/DP-RAM.
		elOAccess_Default: normaler Zugriff via MEMCPY- Funktion möglich
		elOAccess_Byte: nur BYTE-Zugriff via FOR-Schleife möglich
		elOAccess_WordSwap: nur WORT-Zugriff + High/Low- Byte-Swapping via FOR-Schleife möglich

Voraussetzungen

Entwicklungsumge- bung	Zielplattform		Einzubindende SPS-Bibliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Alle IO-Geräte	Tc2_loFunctions (IO)

5.18 ST_Parameter_IN

Eingangsdaten von der ASI-Klemme.

```
TYPE ST_ParameterBuffer :
STRUCT
    iParameterStatus : WORD;
    iParameterReadValue : DWORD;
END_STRUCT
END_TYPE
```



Byte-Offset	Bit-Offset	Beschreibung
0	0-5	Status-Bits (wie bei bisherigen Klemmen)
0	6	0: keine Diagnose, 1: Diagnose (wie bei bisherigen Klemmen)
0	7	immer 0: keine Registerkommunikation
1	0-3	0-3 Reserviert für Erweiterungen
1	4	Toggle-Bit, um Auftrag zu quittieren (bei Cyclic wird das Bit 6 aus Byte 0 kopiert)
1	5	Quittung (0: NoError, 1: Error)
1	6	0: Cyclic, 1: Acyclic
1	7	0: Parameterzugriff, 1: ADS
2		Input-Daten (Cyclic), Parameterwert (Acyclic) oder Fehlernummer Bit 0-7
3		Input-Daten (Cyclic), Parameterwert (Acyclic) oder Fehlernummer Bit 8-15
4		Input-Daten (Cyclic), Parameterwert (Acyclic) oder Fehlernummer Bit 16-23
5		Input-Daten (Cyclic), Parameterwert (Acyclic) oder Fehlernummer Bit 24-31

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
		g. a.bba)

5.19 ST_Parameter_OUT

Ausgangsdaten zur ASI-Klemme.

```
TYPE ST_ParameterBuffer :
STRUCT
ParameterControl : WORD;
iParametervalue : DWORD;
END_STRUCT
END_TYPE
```

Byte-Offset	Bit-Offset	Beschreibung
0	0-5	Parameternummer Bit 0-5 (oder Parameter-Offset)
0	6	Bei Acyclic: 0: Read, 1:Write, bei Cyclic (immer Read/Write) wird das Bit in die Inputdaten kopiert, um eine direkte Zuordnung zu haben (dann könnten die Cyclic-Daten auch geändert werden)
0	7	0: Parameterzugriff, 1: Registerkommunikation
1	0-5	Parameternummer Bit 6-11 (oder Parameter-Page)
1	6	0: Cyclic, 1: Acyclic
1	7	0: Parameterzugriff, 1: ADS
2		Output-Daten (Cyclic) oder Parameterwert (Acyclic) Bit 0-7
3		Output-Daten (Cyclic) oder Parameterwert (Acyclic) Bit 8-15
4		Output-Daten (Cyclic) oder Parameterwert (Acyclic) Bit 16-23
5		Output-Daten (Cyclic) oder Parameterwert (Acyclic) Bit 24-32



Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	PC oder CX (x86)	ASI Masterklemme	Tc2 IoFunctions (IO)

5.20 ST_ParameterBuffer

Datenpuffer für die E/A-Daten der ASI-Klemme

```
TYPE ST_ParameterBuffer :

STRUCT

ParameterControl : ARRAY[0..50] OF WORD;
iParametervalue : ARRAY[0..50] OF DWORD;
iParameterStatus : ARRAY[0..50] OF WORD;
iParameterReadValue : ARRAY[0..50] OF DWORD;
icounterState : INT;
icounterControl : INT;
END_STRUCT
END_TYPE
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	PC oder CX (x86)	ASI Masterklemme	Tc2_loFunctions (IO)

5.21 ST_PD_Dpv1Error

```
TYPE ST_PD_DpvlError:
STRUCT
bError: BOOL;
eErrorId: E_PD_DpvlError;
END_STRUCT
END_TYPE
```

Name	Тур	Beschreibung
bError	BOOL	Fehlerflag (TRUE => Fehler, FALSE => kein Fehler)
eErrorID	E_PD_Dpv1Error [▶ 131]	Fehlercode

Voraussetzungen

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Beckhoff FC310x PCI, CX1500-M310, EL6731, EL6632	Tc2_loFunctions (IO)

5.22 ST_PNET_CCDSTS

```
TYPE ST_PNET_CCDSTS:
STRUCT

iCycleCounter: UINT;
iDataState: USINT;
iTransferState: USINT;
END_STRUCT
END_TYPE
```



Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	PC oder CX (x86)	Beckhoff EL6632	Tc2 IoFunctions (IO)

5.23 ST_PNIOConfigRecord

```
TYPE ST_PNIOConfigRecord:

STRUCT

iRW : UINT;
iNumOfAR : UINT;
iAPI : UDINT;
iSlot : UINT;
iSubSlot : UINT;
stPNIORecord: ST_PNIORecord;
END_STRUCT
END_TYPE
```

Name	Тур	Beschreibung
iRW	UINT	0: Read, 1: Write.
iNumOfAR	UINT	Anzahl der Argumente.
iAPI	UDINT	API-Nummer.
iSlot	UINT	Slot-Nummer.
iSubSlot.	UINT	SubSlot-Nummer.
stPNIORecord	ST PNIORecord [145]	Typ: <u>ST_PNIORecord [▶ 145]</u>

Voraussetzungen

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	PC oder CX (x86)	Beckhoff EL6632	Tc2 IoFunctions (IO)

5.24 ST_PNIORecord

```
TYPE ST_PNIORecord :
STRUCT
   iIndex   : UINT;
   iLength   : UINT; (* 0 for READ *)
   iTransfSeq : UINT;
   iAligned   : UINT;
END_STRUCT
END_TYPE
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	PC oder CX (x86)	Beckhoff EL6632	Tc2_loFunctions (IO)

5.25 ST_PNIOState

```
TYPE ST_PNIOState :
STRUCT
    bInDataExchange : BOOL; (* bit 0 *)
    bApplRunning : BOOL; (* bit 2 *)
    bDiagIndicator : BOOL; (* bit 3 *)
END_STRUCT
END_TYPE
```



Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	PC oder CX (x86)	Beckhoff EL6632	Tc2 IoFunctions (IO)

5.26 ST_PZD_IN

Datenwörter vom Antrieb zur PLC.

```
TYPE ST_PZD_IN:
STRUCT

WSTW:WORD;
WHIW:WORD;
PZD3:WORD;
PZD4:WORD;
PZD5:WORD;
PZD5:WORD;
PZD6:WORD;
PZD6:WORD;
PZD7:WORD;
PZD8:WORD;
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	PC oder CX (x86)	AX2000 Profibus box	Tc2_loFunctions (IO)

5.27 ST_PZD_OUT

Datenwörter von der PLC zum Antrieb.

```
TYPE ST_PZD_OUT:
STRUCT

WCtrlW:WORD;
PZD2:WORD;
PZD3:wORD;
PZD4:WORD;
PZD5:WORD;
PZD5:WORD;
PZD6:WORD;
PZD6:WORD;
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	PC oder CX (x86)	AX2000 Profibus box	Tc2_loFunctions (IO)

5.28 ST_RAIDCntIrFound

```
TYPE ST_RAIDCntlrFound:

STRUCT

nRAIDCntlrCount: UDINT;

nRAIDCntlrIds: ARRAY [1..g_nMAX_NUMBER_OF_RAID_CNTLRS] OF UDINT;

END_STRUCT
END_TYPE
```

Name	Тур	Beschreibung
nRAIDCntlrCount	UDINT	Anzahl von RAID-Controllern
nRAIDCntlrlds		ID von jedem RAID-Controller (Default-Wert ist 4294967295 und daher ungültig).



g_nMAX_NUMBER_OF_RAID_CNTLRS ist die maximale Anzahl von RAID-Controllern und ist definiert als globale Konstante = 10.

Voraussetzungen

Entwicklungsumgebung	•	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_loFunctions (IO)

5.29 ST_RAIDConfigReq

```
TYPE ST_RAIDConfigReq:
STRUCT

nRAIDCntlrID: UDINT;
nRAIDSetIndex: UDINT;
END_STRUCT
END_TYPE
```

Name	Тур	Beschreibung
nRAIDCntlrID	UDINT	ID des RAID-Controllers
nRAIDSetIndex		Enthält die Nummer des RAID-Sets für das Information angefordert wird. Bitte beachten Sie, dass im Fall von Beckhoff Boards CBx051 das Sets mit Index 0 beginnt.

Voraussetzungen

Entwicklungsumgebung	•	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_loFunctions (IO)

5.30 ST_RAIDDriveStatus

```
TYPE ST_RAIDDriveStatus :

STRUCT

eRAIDDriveStatus : E_RAIDDriveStatus;

eRAIDDriveUsage : E_RAIDDriveUsage;

nSATAPort : UINT;

sRAIDDriveSerial : STRING [39];

END_STRUCT
END_TYPE
```

Name	Тур	Beschreibung
eRAIDDriveStatus	E RAIDDriveStatus [▶ 133]	Enthält den Status des physischen Antriebs.
eRAIDDriveUsage	E_RAIDDriveUsage [▶ 134]	Besagt, ob der physische Antrieb Teil des RAID-Sets ist.
nSATAPort	UINT	Enthält die SAS-Adresse des physischen Antriebs. Wenn der Antrieb keine SAS-Adresse hat, wie z. B. mit einem direkt angehängten SATA Antrieb, dann sollte dieses Feld mit 0 gefüllt werden.
sRAIDDriveSerial	STRING	Seriennummer des RAID-Antriebs (40 Buchstaben).

Voraussetzungen

Entwicklungsumgebung	•	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_loFunctions (IO)

5.31 ST_RAIDInfo

```
TYPE ST_RAIDInfo:
STRUCT
nNumRAIDSets: UDINT;
```



nMaxDrivesPerSet : UDINT; bReserved : ARRAY [1..92] OF BYTE; END STRUCT END_TYPE

Name	Тур	Beschreibung
nNumRAIDSets	UDINT	Anzahl der aktuell definierten RAID-Sets. Falls bislang keine Sets definiert worden sind, wird der Wert 0 zurückgegeben.
nMaxDriverPerSet	UDINT	Maximale Anzahl der physischen Antriebe in einem logischen RAID-Set. Dies kann ein absolutes Maximum sein oder das aktuell für alle Sets definierte Maximum.
bReserved	UDINT	Reserviert für interne Zwecke.

Voraussetzungen

TYPE ST RAIDStatusRes :

Entwicklungsumgebung	•	Einzubindende SPS-Bibliotheken (Kategoriegruppe)		
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_loFunctions (IO)		

ST_RAIDStatusRes 5.32

STRUCT nRAIDSetIndex : UDINT; eRAIDType : E_RAIDType;
eRAIDStatus : E_RAIDStatus;
nRAIDDriveCount : UINT:

: UINT; nReserved

stRAIDDriveStatus : ARRAY [1..g nMAX NUMBER OF RAID DRIVES] OF ST RAIDDriveStatus;

END TYPE

Name	Тур	Beschreibung
nRAIDSetIndex	UDINT	RAID-Set-ID, wie beim Input.
eRAIDType	E_RAIDType [▶_135]	Enthält den Basis-RAID-Typ des RAID-Sets eRAID_TYPE_NONE zeigt an, dass das RAID-Set aus einem einzigen Antrieb besteht, also kein Set mit der angegebenen Nummer existiert.
eRAIDStatus	E_RAIDStatus [▶ 134]	Enthält den Status des RAID-Sets
nRAIDDriveCount	UINT	Enthält die Anzahl von Antrieben im RAID-Set
nReserved	UINT	Reserviert
stRAIDDriveStatus	ST_RAIDDriveStatus [▶ 147]	Enthält den Status des physischen Antriebs und die Information, ob der physische Antrieb Teil des RAID-Sets ist.
nRAIDSetIndex	ARRAY OF ST_RAIDDriveStatus	RAID-Set-ID, wie beim Input.

g_nMAX_NUMBER_OF_RAID_DRIVES ist die Anzahl der RAID-Antriebs-Status die gelesen werden können und ist definiert als globale Konstante = 10.

Voraussetzungen

Entwicklungsumgebung	•	Einzubindende SPS-Bibliotheken (Kategoriegruppe)		
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_loFunctions (IO)		

ST_SercosParamAttrib 5.33

ST_SercosParamAttrib beinhaltet das Attribut nAttrib des Sercos-Parameters in einzelne Variablen zerlegt.



```
TYPE ST_SercosParamAttrib:

STRUCT

nFactor : UINT;
eLength : E_SercosAttribLen;
bCommand : BOOL;
eType : E_SercosAttribType;
bReserved1 : BOOL;
nComma : USINT;
bWriteProtCP2 : BOOL;
bWriteProtCP3 : BOOL;
bWriteProtCP4 : BOOL;
bReserved2 : BOOL;
END_STRUCT
END_TYP
```

Name	Тур	Beschreibung
nFactor	UINT	Bits 015.
eLength	E SercosAttribLen [▶ 135]	Bits 1618.
bCommand	BOOL	Bit 19.
еТуре	E_SercosAttribType [▶ 136]	Bits 2022.
bReserved1	BOOL	Bit 23.
nComma	USINT	Bits 2427.
bWriteProtCP2	BOOL	Bit 28.
bWriteProtCP3	BOOL	Bit 29.
bWriteProtCP4	BOOL	Bit 30.
bReserved2	BOOL	Bit 31.

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	- (/	Sercans SCS-P ISA; Sercans SCS-P PCI; Beckhoff FC750x PCI	Tc2_loFunctions (IO)

5.34 ST_SercosParamErrList

```
TYPE ST_SercosParamErrList:

STRUCT

iActCount : UINT;
iMaxCount : UINT;
iList : ARRAY [0..2047] OF UINT;
iError : ARRAY [0..2047] OF UDINT;
END_STRUCT
END_TYPE
```

Name	Тур	Beschreibung
iActCount	UINT	Anzahl der übersprungenen Parameter (hier bedeutet 3 = 3 Parameterfehler).
iMaxCount	UINT	Anzahl der übersprungenen Parameter (hier bedeutet 3 = 3 Parameterfehler).
iList	ARRAY OF UINT	Feld von bis zu 2048 Parameternummern, bei denen Zugriffsfehler auftraten.
iError	ARRAY OF UDINT	Feld von bis zu 2048 Zugriffsfehlernummern.



Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	- (/	Sercans SCS-P ISA; Sercans SCS-P PCI; Beckhoff FC750x PCI	Tc2_loFunctions (IO)

5.35 ST_SercosParamList

```
TYPE ST_SercosParamList:

STRUCT

iActCount : UINT;

iMaxCount : UINT;

iList : ARRAY [0..2047] OF UINT;

END_STRUCT

END_TYPE
```

Name	Тур	Beschreibung
iActCount	UNIT	Aktuelle Anzahl der Parameter einer Liste * 2. Sercos speichert hier die Anzahl der Bytes, eine Parameternummer besteht aus zwei Bytes, z.B. 6 bedeutet 3 Parameter.
iMaxCount	UINT	Maximale Anzahl der Parameter einer Liste * 2. Sercos speichert hier die Anzahl der Bytes, eine Parameternummer besteht aus zwei Bytes, z.B. 6 bedeutet 3 Parameter.
iList	ARRAY OF UINT	Feld von bis zu 2048 Parameternummern.

Voraussetzungen

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	,	Sercans SCS-P ISA; Sercans SCS-P PCI; Beckhoff FC750x PCI	Tc2_loFunctions (IO)

5.36 ST_UPSStatus

```
TYPE ST UPSStatus
STRUCT
                          : STRING; (* UPS vendor name *)
    Vendor
                          : STRING; (* UPS model name *)
    Model
    FirmwareRev : STRING; (* UPS firmware revision *)
SerialNumber : STRING; (* UPS serial number *)
    BatteryLifePercent: DWORD; (* The percent of battery capacity remaining in the UPS (0..100%) *)
BatteryLifeTime: DWORD; (* Remaining UPS run time, in minutes *)
eBatteryStatus: E_BatteryStatus; (* UPS battery state *)
    eCommStatus
                          : E_UpsCommStatus; (* Status of the communication path to the UPS *)
: E_UpsPowerStatus; (* Status of utility-supplied power into the UPS *)
    ePowerStatus
    nPowerFailCnt
                         : DWORD; (* Power Fail counter. Increments every time the UPS service detects
 power fail *)
    dwChargeFlags
                          : DWORD; (* Battery charge status flags. This member can be one or more of th
e following values.
    Bits0..7 := General battery status flags (if all bits are set to 0 => unknown status)
         Bit0 := High (bit set => high battery charge)
         Bit1 := Low (bit set => low battery charge)
         Bit2 := Critical (bit set => battery is near empty)
         Bit3 := Charging (bit set => battery is charging)
         Bits4..6 := reserved (all bits are 0)
         Bit7 := No Battery (bit set => battery not found or not connected, bit not set => battery is
 existing or unknown state)
   Bits8..15 := Special status information (if all bits are set to 0 => state ok or unknown state)
         Bit8 := UPS Fan Error (bit set => fan hardware reports an error, bit not set => fan is ok)
         Bit9 := Over Temperature (bit set => over temperature detected, bit not set => temperature i
```



```
s ok)
     Bit10 := Service Interval Notify (bit set => service interval time expired, bit not set => service interval time not expired )
     Bit11 := Under Temperature (bit set => under temperature detected , bit not set => temperature is ok )
     Bit12 := Fuse Not Ok (bit set => fuse broken or missed, bit not set => fuse ok)
     Bit13 := Alarm1 (reserved for later use, bit is 0)
     Bit14 := Alarm2 (reserved for later use, bit is 0)
     Bit15 := Alarm3 (reserved for later use, bit is 0)
     Bits16..31 := (reserved for later use, all bits are 0)
*)
END_STRUCT
END_TYPE
```

Nicht alle USV-Modelle können alle Statusinformationen liefern.

X: Die Statusinformation ist bei diesem Modell vorhanden.

*) Nur vorhanden, wenn das Model "Smart Signaling to any APC UPS & TwinCAT" konfiguriert wurde.

Statusin- formation	CU81x0-0xx0	Beckhoff BAPI v1	Beckhoff P24Vxxxx	Beckhoff CP903x ISA/ PCI-Karte	Beckhoff CX2100-09x4	APC Back- UPS Pro 280	APC Smart- UPS 420	Beschreibung
Vendor	Х	Х	X	Х	X	Х	Х	Herstellername.
Model	Х	Х	X	Х	X	Х	Х	Modelstring. Leerstring, wenn keine USV konfiguriert wurde.
Firmware Rev	Х	Х	X	X	X	Х	Х	Versionsinformat ionen zur USV- Firmware. Leerstring, wenn die USV diesen Parameter nicht unterstützt.
SerialNum ber	Х	X	Х	Keine	Х	Х	Х	Seriennummer der USV. Leerstring, wenn die USV diesen Parameter nicht unterstützt.
BatteryLif ePercent	X	X	X	Keine	X	Х	Х	Verbliebene Akkulaufzeit in Prozent. Der Wert ist immer Null wenn die USV diesen Parameter nicht liefern kann.
BatteryLif eTime	Х	Х	Х	Keine	X	Х	Х	Verbliebene Akkulaufzeit in Minuten. Der Wert ist immer Null wenn die USV diesen Parameter nicht liefern kann.
eBatteryS tatus	BatteryUnkno wnStatus wenn kein Akku vorhanden ist (gilt nur für das Model mit Batterie).	BatteryOk	BatteryUnkno wnStatus wenn kein Akku vorhanden ist, ab USV- Softwareversio n >=2.0.0.6 und USV-Firmware >= 25.1.1	BatteryUnkno wnStatus wenn kein Akku vorhanden ist. BatteryOk	BatteryUnkno wnStatus wenn kein Akku vorhanden ist (gilt nur für das Model mit "Smart Battery" und nicht mit Kondensatoren).	X	X	Akku-Status (Typ: E_BatteryStatus).
			BatteryOk					



Statusin- formation	CU81x0-0xx0	Beckhoff BAPI v1	Beckhoff P24Vxxxx	Beckhoff CP903x ISA/ PCI-Karte	Beckhoff CX2100-09x4	APC Back- UPS Pro 280	APC Smart- UPS 420	Beschreibung
eCommSt atus	X	X	X	X	X	Х	X	Status der Kommunikation zur USV (Typ: E_UpsCommSta tus).
ePowerSt atus	Х	Х	Х	Х	Х	Х	X	Status der externen Spannungsverso rgung (Typ: E_UpsPowerSta tus).
nPowerFa ilCnt	Х	X	X	X	X	*X	*X	Power-Fail- Zähler. Der Zähler wird inkrementiert wenn ein Spannungsausfa II vom USV- Service erkannt wurde.
dwCharge Flags	No Battery (Bit 7 gesetzt). Keine Kommunikation zum Akku (gilt nur für das Model mit Batterie). Service Interval Notify (Bit 10 gesetzt) Die konfigurierte Akku-Service-Intervallzeit ist abgelaufen.	7 gesetzt) ab USV-Firmware	No Battery (Bit 7 gesetzt) ab USV- Softwareversio n >= 2.0.0.6 und Firmware >= 25.1.I Die Existenz des Akkus wird jede Minute überprüft. UPS Fan Error (Bit 8 gesetzt) ab USV- Softwareversio n >= 2.0.0.7 und Firmware >= 40.1.I Der USV Lüfterstatus wird jede Minute überprüft. Erfordert eine neuere (zweite) Hardwarerevis ion! Service Interval Notify (Bit 10 gesetzt). Der konfigurierte Akkuwechsel Intervall Service ist abgelaufen. Implementiert in der USV-Softwareversio n >= 3.0.0.8;	High (Bit 0 gesetzt) wenn Akku voll geladen. Charging (Bit 3 gesetzt) No Battery (Bit 7 gesetzt) wenn kein Akku gefunden wurde.	No Battery (Bit 7 gesetzt). Keine Kommunikation zum Akku (gilt nur für das Model mit "Smart Battery" und nicht mit Kondensatoren). Over Temperature (Bit 9 gesetzt) wenn Übertemperatur detektiert wurde und das Laden des Akkus unterbrochen wurde. Erfordert eine neuere (zweite) Hardwarerevis ion. Implementiert in der USV-Softwareversio n >= 3.0.0.18. Service Interval Notify (Bit 10 gesetzt) Die konfigurierte Akku-Service-Intervallzeit ist abgelaufen. Under Temperature (Bit 11 gesetzt) wenn Untertemperatur detektiert wurde und das Laden des	Keine	Keine	Akku- Ladestatus- Flags und spezielle Statusinformatio nen.



Statusin- formation	CU81x0-0xx0	Beckhoff BAPI v1	Beckhoff P24Vxxxx	Beckhoff CP903x ISA/ PCI-Karte	Beckhoff CX2100-09x4	APC Back- UPS Pro 280	APC Smart- UPS 420	Beschreibung
					Akkus unterbrochen wurde. Erfordert eine neuere (zweite) Hardwarerevis ion. Implementiert in der USV- Softwareversio n >= 3.0.0.18.			
					Fuse Not Ok (Bit 12 gesetzt) Die "Smart Battery"- Sicherung ist defekt oder nicht vorhanden. Erfordert eine neuere (zweite) Hardwarerevis ion. Implementiert in der USV- Softwareversio n >= 3.0.0.18.			

Entwicklungsum- gebung	Zielplattform	USV Hardware	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Beckhoff CU81x0-0xx0	Tc2_loFunctions
		Beckhoff BAPI v1;	(IO)
		Beckhoff P24Vxxxx;	
		Beckhoff CP903x-Karte (PCI/ ISA);	
		Beckhoff CX2100-09x4 Modelle (z.B. CX2100-0904 oder CX2100-0914 + "Smart Battery" CX2900-0192);	
		Die mit Beckhoff Industrie-PC ausgelieferten APC-Geräte die das Smartprotokoll unterstützen und mit dem Windows USV- Dienst konfiguriert werden können;	

5.37 ST_KL1501InData

Struktur zur Verknüpfung im Eingangs-Prozessabbild.

```
TYPE ST_KL1501InData:
STRUCT
iStatus: USINT;
arrDataIn: ARRAY[0..1] OF UINT;
END_STRUCT
END_TYPE
```



Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bi- bliotheken
TwinCAT v3.1.4018.26	PC/CX	Tc2_loFunctions ab v3.3.5.0

5.38 ST_KL1501OutData

Struktur zur Verknüpfung im Ausgangs-Prozessabbild.

```
TYPE ST_KL1501OutData:

STRUCT

iCtrl : USINT;

arrDataOut : ARRAY[0..1] OF UINT;

END_STRUCT

END_TYPE
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bi- bliotheken
TwinCAT v3.1.4018.26	PC/CX	Tc2_loFunctions ab v3.3.5.0

5.39 ST_KL27x1InData

Struktur zur Verknüpfung im Eingangs-Prozessabbild.

```
TYPE ST_KL27x1InData:
STRUCT

iStatus: USINT;
iDataIn: INT;
END_STRUCT
END_TYPE
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken
TwinCAT v3.1.4018.26	PC/CX	, ,	Tc2_loFunctions ab v3.3.5.0

5.40 ST_KL27x1OutData

Struktur zur Verknüpfung im Ausgangs-Prozessabbild.

```
TYPE ST_KL27x1OutData:

STRUCT

iCtrl : USINT;

iDataOut : INT;

END_STRUCT

END_TYPE
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken
TwinCAT v3.1.4018.26	PC/CX	,	Tc2_loFunctions ab v3.3.5.0



5.41 ST_KL320xInData

Struktur zur Verknüpfung im Eingangs-Prozessabbild.

```
TYPE ST_KL320xInData:
STRUCT
iStatus: USINT;
iDataIn: INT;
END_STRUCT
END_TYPE
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken
TwinCAT v3.1.4018.26	PC/CX	KL3201, KL3202, KL3204	Tc2_loFunctions ab v3.3.5.0

5.42 ST_KL320xOutData

Struktur zur Verknüpfung im Ausgangs-Prozessabbild.

```
TYPE ST_KL320xOutData:
STRUCT
iCtrl : USINT;
iDataOut : INT;
END_STRUCT
END_TYPE
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken
TwinCAT v3.1.4018.26	PC/CX	KL3201, KL3202, KL3204	Tc2_loFunctions ab v3.3.5.0

5.43 ST_KL3208InData

Struktur zur Verknüpfung im Eingangs-Prozessabbild.

```
TYPE ST_KL3208InData:
STRUCT
iStatus: USINT;
iDataIn: INT;
END_STRUCT
END_TYPE
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bi- bliotheken
TwinCAT v3.1.4018.26	PC/CX	Tc2_loFunctions ab v3.3.5.0

5.44 ST_KL3208OutData

Struktur zur Verknüpfung im Ausgangs-Prozessabbild.

```
TYPE ST_KL3208OutData:

STRUCT

iCtrl : USINT;

iDataOut : INT;

END_STRUCT

END_TYPE
```



Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bi- bliotheken
TwinCAT v3.1.4018.26	PC/CX	Tc2_loFunctions ab v3.3.5.0

5.45 ST_KL3228InData

Struktur zur Verknüpfung im Eingangs-Prozessabbild.

```
TYPE ST_KL3228InData:
STRUCT
iStatus: USINT;
iDataIn: INT;
END_STRUCT
END_TYPE
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bi- bliotheken
TwinCAT v3.1.4018.26	PC/CX	Tc2_loFunctions ab v3.3.5.0

5.46 ST_KL3228OutData

Struktur zur Verknüpfung im Ausgangs-Prozessabbild.

```
TYPE ST_KL3228OutData:

STRUCT

iCtrl : USINT;

iDataOut: INT;

END_STRUCT

END_TYPE
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bi- bliotheken
TwinCAT v3.1.4018.26	PC/CX	Tc2_loFunctions ab v3.3.5.0



6 Globale Konstanten

6.1 Bibliotheksversion

Alle Bibliotheken haben eine bestimmte Version. Diese Version wird im Repository der SPS-Bibliothek angezeigt.

Die Versionsnummer der Bibliothek ist in einer globalen Konstante gespeichert.

Global_Version

```
VAR_GLOBAL CONSTANT
stLibVersion_Tc2_IoFunctions : ST_LibVersion;
END VAR
```

Name	Тур	Beschreibung
stLibVersion_Tc2_lo	ST_LibVersion	Versionsnummer der Tc2_loFunctions-Bibliothek
Functions		

Zum Vergleich zwischen vorhandener und erforderlicher Version dient die Funktion F_CmpLibVersion (in der Tc2_System-Bibliothek)



Kompatibilität zu TwinCAT 2

Abfragemöglichkeiten von TwinCAT 2 Bibliotheken sind nicht mehr verfügbar.

Voraussetzungen

Entwicklungsumge- bung	Zielplattform		Einzubindende SPS-Bibliotheken (Kategorie-gruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Alle IO-Geräte	Tc2_loFunctions (IO)



7 Anhang

7.1 SERCOS Dateiformat der Backup-Datei

- 1. Datei-Header vom Typ ST_SercosFileHeader
- 2. n * Daten
- a) Parameter-Header vom Type ST ParamHeader
- b) Parameter-Daten als Bytes

Beispiel für n Parameter

```
1 * ST SercosFileHeader (268
Bytes)
nVersion
           ( 4 Bytes)
nListType (4 Bytes)
cbCommentLen ( 4 Bytes)
sComment (256 Bytes)
n * (ST SercosParamHeader + Data)
nIDN
            ( 2 Bytes)
cbSize
            ( 2 Bytes)
            ( 4 Bytes)
nAttrib
            (cbSize Bytes),
arrData
kann für jeden Parameter verschieden groß sein, je nach Typ oder
Listenlänge
```

Beispiel für 3 Parameter

TYPE ST_SercosFileHeader (268 Bytes)

Der Datei-Header der Sercos-Backup-Datei basiert auf der Struktur ST SercosFileHeader.

```
TYPE ST_SercosFileHeader :
STRUCT
   nVersion    : UDINT;(* 4 Bytes *)
   nListType    : UDINT;(* 4 Bytes *)
```



```
cbCommentLen : UDINT;(* 4 Bytes *)
sComment : T_MaxString;(* 256 Bytes *)
END_STRUCT
END_TYPE
```

Name	Тур	Beschreibung
nVersion	UDINT	Beinhaltet die Dateiversion, momentan 1.
nListType	UDINT	Beinhaltet die IDN-Parameterliste, die für das Backup benutzt wurde. Der Standard-Wert ist 192 (Liste aller Backup-Parameter), bei benutzerdefinierter Backupliste steht hier 0. Alternativ kann die Liste aller Sercos-Parameter (IDN 17) verwendet werden. Das Restore erfordert allerdings die Liste aus Parameter 192 oder über die benutzerdefinierte Liste (0) erfolgen.
cbCommentLen	UDINT	Länge des Kommentars der Backup-Datei.
sComment	T_MaxString	Kommentar der Backup-Datei. Der String wird mit allen 256 Zeichen geschrieben.

TYPE ST_SercosParamHeader (8 Bytes)

Im Anschluss an den Datei-Header folgt in der Backup-Datei je Parameter ein Parameter Header vom Typ ST_SercosParamHeader.

```
TYPE ST_SercosParamHeader:

STRUCT

nIDN : UINT; (* 2 Bytes *)

cbSize : UINT; (* 2 Bytes *)

nAttrib : DWORD; (* 4 Bytes *)

END_STRUCT

END_TYPE
```

Name	Тур	Beschreibung
nIDN		Sercos-Parameter-Nummer.
cbSize		Länge der Daten in Bytes, die diesem Header folgen. Kann für jeden Parameter verschieden sein, ja nach Parameter Typ oder Listenlänge.
nAttrib	ST_SercosParamData [▶ 148]	Attribut des Sercos-Parameters, wird zur Bestimmung von Länge und Datentyp benötigt.

Parameter Daten (cbSize Bytes)

Auf jeden Sercos-Parameter-Header in der Backup-Datei folgen unmittelbar die Daten. Die Anzahl der Daten-Bytes ist im Parameter-Header in cbSize gespeichert.

Voraussetzungen

Entwicklungsumgebung	Zielplattform		Einzubindende SPS-Bi- bliotheken (Kategorie- gruppe)
TwinCAT v3.1.0	- (/	Sercans SCS-P ISA; Sercans SCS-P PCI; Beckhoff FC750x PCI	Tc2_loFunctions (IO)

7.2 AX200x Profibus Parameternummer

PNU	Datentyp	Zugriff		SERVOSTAR™AS- CII-Befehl
Profilparamete	r			
904	UINT32		Nummer des unterstützen PPO-Write, immer 2	-



PNU	Datentyp	Zugriff	Kurzbeschreibung	SERVOSTAR™AS- CII-Befehl
911	UINT32	ro	Nummer des unterstützen PPO-Read, immer 2	-
918	UINT32	ro	Teilnehmeradresse am PROFIBUS	ADDR
930	UINT32	r/w	Auswahlschalter für Betriebsart	-
963	UINT32	ro	PROFIBUS-Baudrate	-
965	Octet-String2	ro	Nummer des PROFIDRIVE-Profils (0302H)	-
970	UINT32	wo	Defaultparametersatz laden	RSTVAR
971	UINT32	wo	Parameter nichtflüchtig speichern	SAVE
Herstellers	pezifische Parame	ter SERV	OSTAR™	1
Allgemeine	Parameter			
1000	Visible String4	ro	Gerätekennung	-
1001	UINT32	ro	Herstellerspezifisches Fehlerregister	ERRCODE
1002	UINT32	ro	Herstellerspezifisches Statusregister	-
Drehzahlre	glerparameter	1		I
1200	UINT32	r/w	Kp – Verstärkungsfaktor des Drehzahlreglers	GV
1201	UINT32	r/w	Tn – Nachstellzeit des Drehzahlreglers	GVTN
1202	UINT32	r/w	PID – T2 – Zeitkonstante des Drehzahlreglers	GVT2
1203	UINT32	r/w	Sollwertrampe+, Drehzahlregler	ACC
1204	UINT32	r/w	Sollwertrampe-, Drehzahlregler	DEC
1205	UINT32	r/w	Not-Rampe, Drehzahlregler	DECSTOP
1206	UINT32	r/w	Maximale Drehzahl	VLIM
1207	UINT32	r/w	Überdrehzahl	VOSPD
1208	UINT32	r/w	Zählrichtung	DIR
Lageregler	parameter			
1250	UINT32	r/w	Multiplikator für Geschwindigkeiten Tippen/Ref.	VMUL
1251	UINT32	r/w	Achstyp	POSCNFG
1251	INTEGER32	r/w	In-Position-Fenster	PEINPOS
1253	INTEGER32	r/w	Schleppfehlerfenster	PEMAX
1254	INTEGER32	r/w	Positionsregister 1	SWE1
1255	INTEGER32	r/w	Positionsregister 2	SWE2
1256	INTEGER32	r/w	Positionsregister 3	SWE3
1257	INTEGER32	r/w	Positionsregister 4	SWE4
1258	UINT32	r/w	Auflösung Nenner	PGEARO
1259	UINT32	r/w	Auflösung Zähler	PGEARI
1260	UINT32	r/w	Minimale Beschleunigungs-, Bremszeit	PTMIN
1261	UINT32	r/w	FeedForward-Faktor Lageregler	GPFFV
1262	UINT32	r/w	KV - Faktor Lageregler	GP
1263	UINT32	r/w	KP - Faktor Lageregler	GPV
1264	UINT32	r/w	Tn - Nachstellzeit Lageregler	GPTN
1265	UINT32	r/w	Maximale Geschwindigkeit für Positionierbetrieb	PVMAX
1266	UINT32	r/w	Konfigurationsvariable für Softwareschalter	SWCNFG
1267	UINT32	r/w	Konfigurationsvariable 2 für Softwareschalter	SWCNFG2



PNU	Datentyp	Zugriff	Kurzbeschreibung	SERVOSTAR™AS- CII-Befehl
Positionier	daten für den Lag	ereglermo	dus	
1300	INTEGER32	r/w	Position	O_P
1301	INTEGER16	r/w	Geschwindigkeit	O_V
1302	UINT32	r/w	Fahrauftragsart	O_C
1304	UINT32	r/w	Anfahrzeit (Beschleunigung)	O_ACC1
1305	UINT32	r/w	Bremszeit (Verzögerung)	O_DEC1
1306	UINT32	r/w	Ruckbegrenzung (Beschleunigung)	O_ACC2
1307	UINT32	r/w	Ruckbegrenzung (Verzögerung)	O_DEC2
1308	UINT32	r/w	Nummer des Folgefahrauftrags	O_FN
1309	UINT32	r/w	Startverzögerung für Folgefahrauftrag	O_FT
1310	2 * UINT16	wo	Kopieren eines Fahrauftrags	OCOPY
Einrichtbet	rieb Lage			
1350	UINT32	r/w	Referenzfahrtart	NREF
1351	UINT32	r/w	Referenzfahrtrichtung	DREF
1352	UINT32	r/w	Beschleunigungsrampe (Tippen/ Referenzieren)	ACCR
1353	UINT32	r/w	Bremsrampe	DECR
1354	UINT32	r/w	Referenzoffset	ROFFS
1355	UINT32	ro	Referenzfahrtgeschwindigkeit	VREF
1356	UINT32	ro	Tippgeschwindigkeit	VJOG
Istwerte		'		•
1400	INTEGER32	ro	Istlage 20 Bit / Umdrehung	PRD
1401	INTEGER32	ro	Drehzahl	-
1402	INTEGER32	ro	Inkrementeller Positionsistwert	-
1403	INTEGER32	ro	SI - Positionsistwert	PFB
1404	INTEGER32	ro	SI - Geschwindigkeitsistwert	PV
1405	INTEGER32	ro	SI - Schleppfehler	PE
1406	INTEGER32	ro	Effektivstrom	I
1407	INTEGER32	ro	SI - Drehzahlistwert	V
1408	INTEGER32	ro	Kühlkörpertemperatur	TEMPH
1409	INTEGER32	ro	Innentemperatur	TEMPE
1410	INTEGER32	ro	Zwischenkreisspannung	VBUS
1411	INTEGER32	ro	Ballastleistung	PBAL
1412	INTEGER32	ro	I ² t - Belastung	I2T
1413	INTEGER32	ro	Betriebsdauer	TRUN
Digital I/O -	- Konfiguration		·	
1450	UINT32	r/w	Funktion des digitalen Eingangs 1	IN1MODE
1451	UINT32	r/w	Funktion des digitalen Eingangs 2	IN2MODE
1452	UINT32	r/w	Funktion des digitalen Eingangs 3	IN3MODE
1453	UINT32	r/w	Funktion des digitalen Eingangs 4	IN4MODE
1454	INTEGER32	r/w	Hilfsvariable für digitalen Eingang 1	IN1TRIG
1455	INTEGER32	r/w	Hilfsvariable für digitalen Eingang 2	IN2TRIG
1456	INTEGER32	r/w	Hilfsvariable für digitalen Eingang 3	IN3TRIG
1457	INTEGER32	r/w	Hilfsvariable für digitalen Eingang 4	IN4TRIG
1458	INTEGER32	r/w	Funktion des digitalen Ausgangs 1	O1MODE
1459	INTEGER32	r/w	Funktion des digitalen Ausgangs 2	O2MODE
1460	UINT32	r/w	Hilfsvariable für digitalen Ausgang 1	O1TRIG
1461	UINT32	r/w	Hilfsvariable für digitalen Ausgang 2	O2TRIG



PNU	Datentyp	Zugriff	Kurzbeschreibung	SERVOSTAR™AS- CII-Befehl
1462	UINT32	r/w	Zustand von vier Digitalen Inputs, Enable, 2 digitalen Outputs	STATIO
Analog - K	Configuration			
1500	UINT32	r/w	Konfiguration der analogen Eingangsfunktionen	ANCNFG
1501	UINT32	r/w	Konfiguration Monitorfunktion Analogausgang 1	ANOUT1
1502	UINT32	r/w	Offsetspannung für Analogeingang 1	ANOFF1
1503	UINT32	r/w	Filterzeitkonstante für Analogeingang 1	AVZ1
1504	UINT32	r/w	Skalierungsfaktor Geschwindigkeit Analogeing. 1	VSCALE1
1505	UINT32	r/w	Skalierungsfaktor Strom Analogeingang 1	ISCALE1
1506	UINT32	r/w	Konfiguration Monitorfunktion Analogausgang 2	ANOUT2
1507	UINT32	r/w	Offsetspannung für Analogeingang 2	ANOFF2
1508	UINT32	r/w	Skalierungsfaktor Geschwindigkeit Analogeing. 2	VSCALE2
1509	UINT32	r/w	Skalierungsfaktor Strom Analogeingang 2	ISCALE2
Motorpara	meter			•
1550	UINT32	r/w	Konfiguration Bremse	MBRAKE
1551	UINT32	r/w	Motornummer aus Motordatenbank	MNUMBER

7.3 Fehlercodes

Errld (hex)	Errid (dez)	Beschreibung
0x0	0	Kein Fehler
0x8001	32769	Timeout-Fehler bei der Konfiguration
0x8002	32770	Konfigurationsbaustein passt nicht zum Klemmentyp
0x8010	32784	FB_KL1501Config: Ungültiger Zählertyp gewählt. Siehe Eingang iSetCounterType.
0x8011	32785	FB_KL1501Config: Ungültiger Zählertyp ausgelesen. Siehe Ausgang sCounterType.
0x8018	32792	FB_KL27x1Config: Ungültige Rampenzeit gewählt. Siehe Eingang iSetRampTime.
0x8019	32793	FB_KL27x1Config: Ungültiger Dimmermodus gewählt. Siehe Eingang iSetDimmerMode.
0x801A	32794	FB_KL27x1Config: Ungültige Rampenzeit ausgelesen. Siehe Ausgang sRampTime.
0x801B	32795	FB_KL27x1Config: Ungültiger Dimmermodus ausgelesen. Siehe Ausgang sDimmerMode.
0x801C	32796	FB_KL27x1Config: Ungültiges Nach-Kurzschluss-Verhalten ausgelesen. Siehe Ausgang sAfterShortCircuit.
0x801D	32797	FB_KL27x1Config: Ungültige Netzfrequenz ausgelesen. Siehe Ausgang sLineFrequency.
0x8020	32800	FB_KL3208Config: Ungültiger Sensortyp gewählt. Siehe Eingang iSetSensorType.
0x8021	32801	FB_KL3208Config: Ungültiger Sensortyp ausgelesen. Siehe Ausgang sSensorType.
0x8028	32808	FB_KL320xConfig: Zweileiteranschluss ist für KL3204 nicht zulässig.



Errld (hex)	Errid (dez)	Beschreibung
0x8029	32809	FB_KL320xConfig: Ungültiger Sensortyp gewählt. Siehe Eingang iSetSensorType.
0x802A	32810	FB_KL320xConfig: Ungültiger Sensortyp ausgelesen. Siehe Ausgang sSensorType.
0x8030	32816	FB_KL3228Config: Ungültiger Sensortyp gewählt. Siehe Eingang iSetSensorType.
0x8031	32817	FB_KL3228Config: Ungültiger Sensortyp ausgelesen. Siehe Ausgang sSensorType.

7.4 ADS Return Codes

Gruppierung der Fehlercodes:

Globale Fehlercodes: <u>0x0000 [▶ 163]</u>... (0x9811_0000 ...) Router Fehlercodes: <u>0x0500 [▶ 164]</u>... (0x9811_0500 ...) Allgemeine ADS Fehler: <u>0x0700 [▶ 164]</u>... (0x9811_0700 ...) RTime Fehlercodes: <u>0x1000 [▶ 165]</u>... (0x9811_1000 ...)

Globale Fehlercodes

Hex	Dec	HRESULT	Name	Beschreibung	
0x0	0	0x98110000	ERR_NOERROR	Kein Fehler.	
0x1	1	0x98110001	ERR_INTERNAL	Interner Fehler.	
0x2	2	0x98110002	ERR_NORTIME	Keine Echtzeit.	
0x3	3	0x98110003	ERR_ALLOCLOCKEDMEM	Zuweisung gesperrt - Speicherfehler.	
0x4	4	0x98110004	ERR_INSERTMAILBOX	Postfach voll – Es konnte die ADS Nachricht nicht versendet werden. Reduzieren der Anzahl der ADS Nachrichten pro Zyklus bringt Abhilfe.	
0x5	5	0x98110005	ERR_WRONGRECEIVEHMSG	Falsches HMSG.	
0x6	6	0x98110006	ERR_TARGETPORTNOTFOUND	Ziel-Port nicht gefunden – ADS Server ist nicht gestartet, nicht erreichbar oder nicht installiert.	
0x7	7	0x98110007	ERR_TARGETMACHINENOTFOUND	Zielrechner nicht gefunden – AMS Route wurde nicht gefunden.	
0x8	8	0x98110008	ERR_UNKNOWNCMDID	Unbekannte Befehl-ID.	
0x9	9	0x98110009	ERR_BADTASKID	Ungültige Task-ID.	
0xA	10	0x9811000A	ERR_NOIO	Kein IO.	
0xB	11	0x9811000B	ERR_UNKNOWNAMSCMD	Unbekannter AMS-Befehl.	
0xC	12	0x9811000C	ERR_WIN32ERROR	Win32 Fehler.	
0xD	13	0x9811000D	ERR_PORTNOTCONNECTED	Port nicht verbunden.	
0xE	14	0x9811000E	ERR_INVALIDAMSLENGTH	Ungültige AMS-Länge.	
0xF	15	0x9811000F	ERR_INVALIDAMSNETID	Ungültige AMS Net ID.	
0x10	16	0x98110010	ERR_LOWINSTLEVEL	Installations-Level ist zu niedrig –TwinCAT 2 Lizenzfehler.	
0x11	17	0x98110011	ERR_NODEBUGINTAVAILABLE	Kein Debugging verfügbar.	
0x12	18	0x98110012	ERR_PORTDISABLED	Port deaktiviert – TwinCAT System Service nicht gestartet.	
0x13	19	0x98110013	ERR_PORTALREADYCONNECTED	Port bereits verbunden.	
0x14	20	0x98110014	ERR_AMSSYNC_W32ERROR	AMS Sync Win32 Fehler.	
0x15	21	0x98110015	ERR_AMSSYNC_TIMEOUT	AMS Sync Timeout.	
0x16	22	0x98110016	ERR_AMSSYNC_AMSERROR	AMS Sync Fehler.	
0x17	23	0x98110017	ERR_AMSSYNC_NOINDEXINMAP	Keine Index-Map für AMS Sync vorhanden.	
0x18	24	0x98110018	ERR_INVALIDAMSPORT	Ungültiger AMS-Port.	
0x19	25	0x98110019	ERR_NOMEMORY	Kein Speicher.	
0x1A	26	0x9811001A	ERR_TCPSEND	TCP Sendefehler.	
0x1B	27	0x9811001B	ERR_HOSTUNREACHABLE	Host nicht erreichbar.	
0x1C	28	0x9811001C	ERR_INVALIDAMSFRAGMENT	Ungültiges AMS Fragment.	
0x1D	29	0x9811001D	ERR_TLSSEND	TLS Sendefehler – Secure ADS Verbindung fehlgeschlagen.	
0x1E	30	0x9811001E	ERR_ACCESSDENIED	Zugriff Verweigert – Secure ADS Zugriff verweigert.	



Router Fehlercodes

Hex	Dec	HRESULT	Name	Beschreibung	
0x500	1280	0x98110500	ROUTERERR_NOLOCKEDMEMORY	Lockierter Speicher kann nicht zugewiesen werden.	
0x501	1281	0x98110501	ROUTERERR_RESIZEMEMORY Die Größe des Routerspeichers konnte n werden.		
0x502	1282	0x98110502	ROUTERERR_MAILBOXFULL	Das Postfach hat die maximale Anzahl der möglichen Meldungen erreicht.	
0x503	1283	0x98110503	ROUTERERR_DEBUGBOXFULL	Das Debug Postfach hat die maximale Anzahl der möglichen Meldungen erreicht.	
0x504	1284	0x98110504	ROUTERERR_UNKNOWNPORTTYPE	Der Porttyp ist unbekannt.	
0x505	1285	0x98110505	ROUTERERR_NOTINITIALIZED	Router ist nicht initialisiert.	
0x506	1286	0x98110506	ROUTERERR_PORTALREADYINUSE	Die Portnummer ist bereits vergeben.	
0x507	1287	0x98110507	ROUTERERR_NOTREGISTERED	Der Port ist nicht registriert.	
0x508	1288	0x98110508	ROUTERERR_NOMOREQUEUES	Die maximale Portanzahl ist erreicht.	
0x509	1289	0x98110509	ROUTERERR_INVALIDPORT	Der Port ist ungültig.	
0x50A	1290	0x9811050A	ROUTERERR_NOTACTIVATED	Der Router ist nicht aktiv.	
0x50B	1291	0x9811050B	ROUTERERR_FRAGMENTBOXFULL	Das Postfach hat die maximale Anzahl für fragmentierte Nachrichten erreicht.	
0x50C	1292	0x9811050C	ROUTERERR_FRAGMENTTIMEOUT	Fragment Timeout aufgetreten.	
0x50D	1293	0x9811050D	ROUTERERR_TOBEREMOVED	Port wird entfernt.	

Allgemeine ADS Fehlercodes

Hex	Dec	HRESULT	Name	Beschreibung
0x700	1792	0x98110700	ADSERR_DEVICE_ERROR	Allgemeiner Gerätefehler.
0x701	1793	0x98110701	ADSERR_DEVICE_SRVNOTSUPP	Service wird vom Server nicht unterstützt.
0x702	1794	0x98110702	0702 ADSERR_DEVICE_INVALIDGRP Ungültige Index-Gruppe.	
0x703	1795	0x98110703	ADSERR_DEVICE_INVALIDOFFSET	Ungültiger Index-Offset.
0x704	1796	0x98110704	ADSERR_DEVICE_INVALIDACCESS	Lesen oder Schreiben nicht gestattet. Mehrere Ursachen sind möglich. Beispielsweise beim Anlegen von Routen, dass ein falsches Passwort angegeben wurde.
0x705	1797	0x98110705	ADSERR_DEVICE_INVALIDSIZE	Parametergröße nicht korrekt.
0x706	1798	0x98110706	ADSERR_DEVICE_INVALIDDATA	Ungültige Daten-Werte.
0x707	1799	0x98110707	ADSERR_DEVICE_NOTREADY	Gerät nicht betriebsbereit.
0x708	1800	0x98110708	ADSERR_DEVICE_BUSY	Gerät beschäftigt.
0x709	1801	0x98110709	ADSERR_DEVICE_INVALIDCONTEXT	Ungültiger Kontext vom Betriebssystem - Kann durch Verwendung von ADS Bausteinen in unterschiedlichen Tasks auftreten. Abhilfe kann die Multitasking-Syncronisation in der SPS geben.
0x70A	1802	0x9811070A	ADSERR_DEVICE_NOMEMORY	Nicht genügend Speicher.
0x70B	1803	0x9811070B	ADSERR_DEVICE_INVALIDPARM	Ungültige Parameter-Werte.
0x70C	1804	0x9811070C	ADSERR_DEVICE_NOTFOUND	Nicht gefunden (Dateien,).
0x70D	1805	0x9811070D	ADSERR_DEVICE_SYNTAX	Syntax-Fehler in Datei oder Befehl.
0x70E	1806	0x9811070E	ADSERR_DEVICE_INCOMPATIBLE	Objekte stimmen nicht überein.
0x70F	1807	0x9811070F	ADSERR_DEVICE_EXISTS	Objekt ist bereits vorhanden.
0x710	1808	0x98110710	ADSERR_DEVICE_SYMBOLNOTFOUND	Symbol nicht gefunden.
0x711	1809	0x98110711	ADSERR_DEVICE_SYMBOLVERSIONINVALID	Symbol-Version ungültig – Kann durch einen Online-Change auftreten. Erzeuge einen neuen Handle.
0x712	1810	0x98110712	ADSERR_DEVICE_INVALIDSTATE	Gerät (Server) ist im ungültigen Zustand.
0x713	1811	0x98110713	ADSERR_DEVICE_TRANSMODENOTSUPP	AdsTransMode nicht unterstützt.
0x714	1812	0x98110714	ADSERR_DEVICE_NOTIFYHNDINVALID	Notification Handle ist ungültig.
0x715	1813	0x98110715	ADSERR_DEVICE_CLIENTUNKNOWN	Notification-Client nicht registriert.
0x716	1814	0x98110716	ADSERR_DEVICE_NOMOREHDLS	Keine weiteren Handles verfügbar.
0x717	1815	0x98110717	ADSERR_DEVICE_INVALIDWATCHSIZE	Größe der Notification zu groß.
0x718	1816	0x98110718	ADSERR_DEVICE_NOTINIT	Gerät nicht initialisiert.
0x719	1817	0x98110719	ADSERR_DEVICE_TIMEOUT	Gerät hat einen Timeout.
0x71A	1818	0x9811071A	ADSERR_DEVICE_NOINTERFACE	Interface Abfrage fehlgeschlagen.
0x71B	1819	0x9811071B	ADSERR_DEVICE_INVALIDINTERFACE	Falsches Interface angefordert.



Hex	Dec	HRESULT	Name	Beschreibung
0x71C	1820	0x9811071C	ADSERR DEVICE INVALIDCLSID	Class-ID ist ungültig.
0x71D	1821	0x9811071D	ADSERR DEVICE INVALIDOBJID	Object-ID ist ungültig.
0x71E	1822	0x9811071E	ADSERR_DEVICE_PENDING	Anforderung steht aus.
0x71F	1823	0x9811071F	ADSERR_DEVICE_ABORTED	Anforderung wird abgebrochen.
0x720	1824	0x98110720	ADSERR_DEVICE_WARNING	Signal-Warnung.
0x721	1825	0x98110721	ADSERR_DEVICE_INVALIDARRAYIDX	Ungültiger Array-Index.
0x722	1826	0x98110722	ADSERR_DEVICE_SYMBOLNOTACTIVE	Symbol nicht aktiv.
0x723	1827	0x98110723	ADSERR_DEVICE_ACCESSDENIED	Zugriff verweigert. Mehrere Ursachen sind möglich. Beispielsweise, dass eine Unidirectionale ADS Route in die umgekehrte Richtung verwendet wird.
0x724	1828	0x98110724	ADSERR_DEVICE_LICENSENOTFOUND	Fehlende Lizenz.
0x725	1829	0x98110725	ADSERR_DEVICE_LICENSEEXPIRED	Lizenz abgelaufen.
0x726	1830	0x98110726	ADSERR_DEVICE_LICENSEEXCEEDED	Lizenz überschritten.
0x727	1831	0x98110727	ADSERR_DEVICE_LICENSEINVALID	Lizenz ungültig.
0x728	1832	0x98110728	ADSERR_DEVICE_LICENSESYSTEMID	Lizenzproblem: System-ID ist ungültig.
0x729	1833	0x98110729	ADSERR_DEVICE_LICENSENOTIMELIMIT	Lizenz nicht zeitlich begrenzt.
0x72A	1834	0x9811072A	ADSERR_DEVICE_LICENSEFUTUREISSUE	Lizenzproblem: Zeitpunkt in der Zukunft.
0x72B	1835	0x9811072B	ADSERR_DEVICE_LICENSETIMETOLONG	Lizenz-Zeitraum zu lang.
0x72C	1836	0x9811072C	ADSERR_DEVICE_EXCEPTION	Exception beim Systemstart.
0x72D	1837	0x9811072D	ADSERR_DEVICE_LICENSEDUPLICATED	Lizenz-Datei zweimal gelesen.
0x72E	1838	0x9811072E	ADSERR_DEVICE_SIGNATUREINVALID	Ungültige Signatur.
0x72F	1839	0x9811072F	ADSERR_DEVICE_CERTIFICATEINVALID	Zertifikat ungültig.
0x730	1840	0x98110730	ADSERR_DEVICE_LICENSEOEMNOTFOUND	Public Key vom OEM nicht bekannt.
0x731	1841	0x98110731	ADSERR_DEVICE_LICENSERESTRICTED	Lizenz nicht gültig für diese System.ID.
0x732	1842	0x98110732	ADSERR_DEVICE_LICENSEDEMODENIED	Demo-Lizenz untersagt.
0x733	1843	0x98110733	ADSERR_DEVICE_INVALIDFNCID	Funktions-ID ungültig.
0x734	1844	0x98110734	ADSERR_DEVICE_OUTOFRANGE	Außerhalb des gültigen Bereiches.
0x735	1845	0x98110735	ADSERR_DEVICE_INVALIDALIGNMENT	Ungültiges Alignment.
0x736	1846	0x98110736	ADSERR_DEVICE_LICENSEPLATFORM	Ungültiger Plattform Level.
0x737	1847	0x98110737	ADSERR_DEVICE_FORWARD_PL	Kontext – Weiterleitung zum Passiv-Level.
0x738	1848	0x98110738	ADSERR_DEVICE_FORWARD_DL	Kontext – Weiterleitung zum Dispatch-Level.
0x739	1849	0x98110739	ADSERR_DEVICE_FORWARD_RT	Kontext – Weiterleitung zur Echtzeit.
0x740	1856	0x98110740	ADSERR_CLIENT_ERROR	Clientfehler.
0x741	1857	0x98110741	ADSERR_CLIENT_INVALIDPARM	Dienst enthält einen ungültigen Parameter.
0x742	1858	0x98110742	ADSERR_CLIENT_LISTEMPTY	Polling-Liste ist leer.
0x743	1859	0x98110743	ADSERR_CLIENT_VARUSED	Var-Verbindung bereits im Einsatz.
0x744	1860	0x98110744	ADSERR_CLIENT_DUPLINVOKEID	Die aufgerufene ID ist bereits in Benutzung.
0x745	1861	0x98110745	ADSERR_CLIENT_SYNCTIMEOUT	Timeout ist aufgetreten – Die Gegenstelle antwortet nicht im vorgegebenen ADS Timeout. Die Routeneinstellung der Gegenstelle kann falsch konfiguriert sein.
0x746	1862	0x98110746	ADSERR_CLIENT_W32ERROR	Fehler im Win32 Subsystem.
0x747	1863	0x98110747	ADSERR_CLIENT_TIMEOUTINVALID	Ungültiger Client Timeout-Wert.
0x748	1864	0x98110748	ADSERR_CLIENT_PORTNOTOPEN	Port nicht geöffnet.
0x749	1865	0x98110749	ADSERR_CLIENT_NOAMSADDR	Keine AMS Adresse.
0x750	1872	0x98110750	ADSERR_CLIENT_SYNCINTERNAL	Interner Fehler in Ads-Sync.
0x751	1873	0x98110751	ADSERR_CLIENT_ADDHASH	Überlauf der Hash-Tabelle.
0x752	1874	0x98110752	ADSERR_CLIENT_REMOVEHASH	Schlüssel in der Tabelle nicht gefunden.
0x753	1875	0x98110753	ADSERR_CLIENT_NOMORESYM	Keine Symbole im Cache.
0x754	1876	0x98110754	ADSERR_CLIENT_SYNCRESINVALID	Ungültige Antwort erhalten.
0x755	1877	0x98110755	ADSERR_CLIENT_SYNCPORTLOCKED	Sync Port ist verriegelt.
0x756	1878	0x98110756	ADSERR_CLIENT_REQUESTCANCELLED	Die Anfrage wurde abgebrochen.

RTime Fehlercodes

Hex	Dec	HRESULT	Name	Beschreibung
0x1000	4096	0x98111000	RTERR_INTERNAL	Interner Fehler im Echtzeit-System.
0x1001	4097	0x98111001	RTERR_BADTIMERPERIODS	Timer-Wert nicht gültig.
0x1002	4098	0x98111002	RTERR_INVALIDTASKPTR	Task-Pointer hat den ungültigen Wert 0 (null).



Hex	Dec	HRESULT	Name	Beschreibung
0x1003	4099	0x98111003	RTERR_INVALIDSTACKPTR	Stack-Pointer hat den ungültigen Wert 0 (null).
0x1004	4100	0x98111004	RTERR_PRIOEXISTS	Die Request Task Priority ist bereits vergeben.
0x1005	4101	0x98111005	RTERR_NOMORETCB	Kein freier TCB (Task Control Block) verfügbar. Maximale Anzahl von TCBs beträgt 64.
0x1006	4102	0x98111006	RTERR_NOMORESEMAS	Keine freien Semaphoren zur Verfügung. Maximale Anzahl der Semaphoren beträgt 64.
0x1007	4103	0x98111007	RTERR_NOMOREQUEUES	Kein freier Platz in der Warteschlange zur Verfügung. Maximale Anzahl der Plätze in der Warteschlange beträgt 64.
0x100D	4109	0x9811100D	RTERR_EXTIRQALREADYDEF	Ein externer Synchronisations-Interrupt wird bereits angewandt.
0x100E	4110	0x9811100E	RTERR_EXTIRQNOTDEF	Kein externer Sync-Interrupt angewandt.
0x100F	4111	0x9811100F	RTERR_EXTIRQINSTALLFAILED	Anwendung des externen Synchronisierungs-Interrupts ist fehlgeschlagen.
0x1010	4112	0x98111010	RTERR_IRQLNOTLESSOREQUAL	Aufruf einer Service-Funktion im falschen Kontext
0x1017	4119	0x98111017	RTERR_VMXNOTSUPPORTED	Intel® VT-x Erweiterung wird nicht unterstützt.
0x1018	4120	0x98111018	RTERR_VMXDISABLED	Intel® VT-x Erweiterung ist nicht aktiviert im BIOS.
0x1019	4121	0x98111019	RTERR_VMXCONTROLSMISSING	Fehlende Funktion in Intel® VT-x Erweiterung.
0x101A	4122	0x9811101A	RTERR_VMXENABLEFAILS	Aktivieren von Intel® VT-x schlägt fehl.

Spezifische positive HRESULT Return Codes:

HRESULT	Name	Beschreibung
0x0000_0000	S_OK	Kein Fehler.
0x0000_0001	S_FALSE	Kein Fehler. Bsp.: erfolgreiche Abarbeitung, bei der jedoch ein negatives oder unvollständiges Ergebnis erzielt wurde.
0x0000_0203	S_PENDING	Kein Fehler. Bsp.: erfolgreiche Abarbeitung, bei der jedoch noch kein Ergebnis vorliegt.
0x0000_0256	S_WATCHDOG_TIMEOUT	Kein Fehler. Bsp.: erfolgreiche Abarbeitung, bei der jedoch eine Zeitüberschreitung eintrat.

TCP Winsock-Fehlercodes

Hex	Dec	Name	Beschreibung	
0x274C	10060	WSAETIMEDOUT	Verbindungs Timeout aufgetreten - Fehler beim Herstellen der Verbindung, da die Gegenstelle nach einer bestimmten Zeitspanne nicht ordnungsgemäß reagiert hat, oder die hergestellte Verbindung konnte nicht aufrecht erhalten werden, da der verbundene Host nicht reagiert hat.	
0x274D	10061	WSAECONNREFUSED	Verbindung abgelehnt - Es konnte keine Verbindung hergestellt werden, da der Zielcomputer dies explizit abgelehnt hat. Dieser Fehler resultiert normalerweise aus dem Versuch, eine Verbindung mit einem Dienst herzustellen, der auf dem fremden Host inaktiv ist—das heißt, einem Dienst, für den keine Serveranwendung ausgeführt wird.	
0x2751	10065	WSAEHOSTUNREACH	Keine Route zum Host - Ein Socketvorgang bezog sich auf einen nicht verfügbaren Host.	
Weitere Winsock-Fehlercodes: Win32-Fehlercodes				



7.5 Support und Service

Beckhoff und seine weltweiten Partnerfirmen bieten einen umfassenden Support und Service, der eine schnelle und kompetente Unterstützung bei allen Fragen zu Beckhoff Produkten und Systemlösungen zur Verfügung stellt.

Downloadfinder

Unser <u>Downloadfinder</u> beinhaltet alle Dateien, die wir Ihnen zum Herunterladen anbieten. Sie finden dort Applikationsberichte, technische Dokumentationen, technische Zeichnungen, Konfigurationsdateien und vieles mehr.

Die Downloads sind in verschiedenen Formaten erhältlich.

Beckhoff Niederlassungen und Vertretungen

Wenden Sie sich bitte an Ihre Beckhoff Niederlassung oder Ihre Vertretung für den <u>lokalen Support und</u> Service zu Beckhoff Produkten!

Die Adressen der weltweiten Beckhoff Niederlassungen und Vertretungen entnehmen Sie bitte unserer Internetseite: www.beckhoff.com

Dort finden Sie auch weitere Dokumentationen zu Beckhoff Komponenten.

Beckhoff Support

Der Support bietet Ihnen einen umfangreichen technischen Support, der Sie nicht nur bei dem Einsatz einzelner Beckhoff Produkte, sondern auch bei weiteren umfassenden Dienstleistungen unterstützt:

- Support
- Planung, Programmierung und Inbetriebnahme komplexer Automatisierungssysteme
- umfangreiches Schulungsprogramm für Beckhoff Systemkomponenten

Hotline: +49 5246 963-157
E-Mail: support@beckhoff.com

Beckhoff Service

Das Beckhoff Service-Center unterstützt Sie rund um den After-Sales-Service:

- · Vor-Ort-Service
- Reparaturservice
- Ersatzteilservice
- · Hotline-Service

Hotline: +49 5246 963-460
E-Mail: service@beckhoff.com

Beckhoff Unternehmenszentrale

Beckhoff Automation GmbH & Co. KG

Hülshorstweg 20 33415 Verl Deutschland

Telefon: +49 5246 963-0
E-Mail: info@beckhoff.com
Internet: www.beckhoff.com

Trademark statements
Beckhoff®, ATRO®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, MX-System®, Safety over EtherCAT®, TC/BSD®, TwinCAT®, TwinCAT®, TwinCAT®, TwinCAT®, TwinCATBSD®, TwinSAFE®, XFC®, XPlanar® and XTS® are registered and licensed trademarks of Beckhoff Automation GmbH.
Third-party trademark statements
DeviceNet and EtherNet/IP are trademarks of ODVA, Inc.
Intel, the Intel logo, Intel Core, Xeon, Intel Atom, Celeron and Pentium are trademarks of Intel Corporation or its subsidiaries.
Microsoft, Microsoft Azure, Microsoft Edge, PowerShell, Visual Studio, Windows and Xbox are trademarks of the Microsoft group of companies.

Mehr Informationen: www.beckhoff.com/te1000

Beckhoff Automation GmbH & Co. KG Hülshorstweg 20 33415 Verl Deutschland Telefon: +49 5246 9630 info@beckhoff.com www.beckhoff.com

