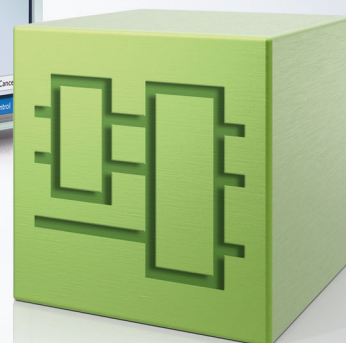
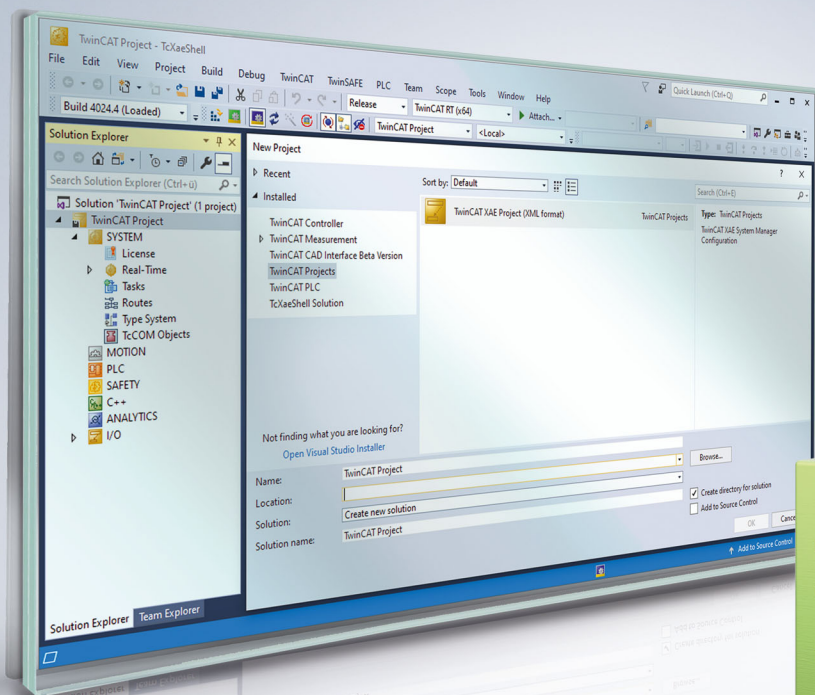


BECKHOFF New Automation Technology

Handbuch | DE

TE1000

TwinCAT 3 | PLC-Bibliothek: Tc3_IOLink



Inhaltsverzeichnis

1	Vorwort.....	5
1.1	Hinweise zur Dokumentation	5
1.2	Zu Ihrer Sicherheit.....	6
1.3	Hinweise zur Informationssicherheit	7
2	Übersicht.....	8
3	Datentypen und Enums	9
3.1	Identification und Diagnose.....	9
3.1.1	E_IolIad_Function	9
3.1.2	E_IolIad_State.....	9
3.1.3	E_IolProfileIdentifier	10
3.1.4	E_IolProfileIdentifier_Bounds.....	11
3.1.5	ST_IolIdentificationObjects	11
3.2	Geräte	12
3.2.1	E_IolPort	12
3.2.2	E_IolPortError	12
3.2.3	E_IolPortState	13
3.3	Spezifikation	14
3.3.1	E_IolDeviceStatus.....	14
3.3.2	E_IolError	14
3.3.3	E_IolSystemcommand	15
3.3.4	ST_IolAccessLocks.....	16
3.3.5	ST_IolDetailedDeviceStatusEntry.....	16
3.3.6	ST_IolDirectParameter1.....	16
3.3.7	ST_IolDirectParameter2.....	17
3.3.8	T_IolDetailedDeviceStatus	17
3.3.9	T_IolProfileCharacteristics	17
3.3.10	T_IolMaxOctedString	17
3.3.11	T_IolMaxString	17
3.3.12	ST_IolIndex	18
3.4	Smart Sensor Profile	18
3.4.1	E_IolMdc_ValueStatus.....	18
4	Funktionen.....	19
4.1	F_IolGetAdsErrorTxt	19
4.2	F_IolGetChannelStateTxt.....	19
4.3	F_IolGetErrorTxt	20
4.4	F_IolGetIolError.....	21
4.5	F_IolGetIolErrorTxt.....	21
4.6	F_IolGetPortError	22
4.7	F_IolGetPortState	22
4.8	F_IolGetProfileIdTxt	23
4.9	F_IolIsBusy	24
5	Funktionsbausteine	25
5.1	FB_IolRead	25

5.2	FB_IoIWrite	27
5.3	IOL_IdentificationAndDiagnosis	29
5.4	IOL_MeasurementDataChannel	32
6	Objektorientierte Funktionsbausteine	34
6.1	I_IoISlave	34
6.2	FB_IOISlave	35
6.2.1	Methoden	35
6.2.2	Beispiel.....	50
6.3	FB_IoIStdVarSlave	52
6.3.1	Methoden	52
6.3.2	Beispiel.....	63
6.4	FB_IoIExtIdentSlave.....	65
6.4.1	Methoden	65
6.4.2	Beispiel.....	67
7	Beispiele	69
8	Anhang.....	70
8.1	Parameter Datenaustausch	70
8.2	Fehlercodes	74
8.2.1	IO-Link-Fehler Codes.....	74
8.2.2	ADS Return Codes.....	77
8.3	Troubleshooting	80

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, für jede Installation und Inbetriebnahme die zu dem betreffenden Zeitpunkt veröffentlichte 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 weiter entwickelt.

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®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® und XPlanar® 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 Bezeichnungen führen.

Patente

Die EtherCAT-Technologie ist patentrechtlich geschützt, insbesondere durch folgende Anmeldungen und Patente:

EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702

mit den entsprechenden Anmeldungen und Eintragungen in verschiedenen anderen Ländern.



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.

Zuwendungen verpflichten zu Schadenersatz. Alle Rechte für den Fall der Patent-, Gebrauchsmuster- oder Geschmacksmustereintragung vorbehalten.

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.

WARNUNG

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 SPS-Bibliothek Tc3_IoLink wird zur Kommunikation mit IoLink-Devices genutzt.

Dazu stehen Funktionsbausteine bereit, die das „Common Profile“ und „Smart Sensor Profile“ unterstützen, sowie das Auslesen und Schreiben von Parametern ermöglichen.

Systemvoraussetzung

Target System	WinXP, WES, Win7, WES7, WEC7 IPC or CX, (x86, x64, ARM)
Min. TwinCAT-Version	3.1.4024.25
Min. TwinCAT-Level	TC1200 TC3 PLC

3 Datentypen und Enums

3.1 Identification und Diagnose

Die Enums und Strukturen werden in Kombination mit dem [IOL_IdentificationAndDiagnosis \[► 29\]](#) verwendet.

3.1.1 E_Iollad_Function

Gibt die Funktion am Funktionsbaustein [IOL_IdentificationAndDiagnosis \[► 29\]](#) an.

Name	Wert	Beschreibung
NoFunction	0x00	Ein Request wird vernachlässigt, es wird keine Funktion ausgeführt.
ReadAll	0x01	Ein Request startet das Rücklesen der aktuellen Identifikations und Diagnoseparameterwerte aus dem Device.
Read	0x02	Ein Request startet das Rücklesen der aktuellen Diagnoseparameterwerte durch Lesen von DeviceStatus und DetailedDeviceStatus aus dem Device.
WritelIdent	0x03	Ein Request bewirkt, dass ein zuvor angelegter Wert für ApplicationSpecificTagIn, LocationTagIn und FunctionTagIn in das Device geschrieben wird.

3.1.2 E_Iollad_State

Zustand des Funktionsbausteins [IOL_IdentificationAndDiagnosis \[► 29\]](#).

Name	Beschreibung
Idle	Funktionsbaustein ist im Leerlaufzustand
Startup	Funktionsbaustein wird gestartet
Done	Funktionsbaustein ist fertig
ReadProfileCharacteristic	Identifikationsparameter schreiben
ReadVendorIDAndDeviceID	Liest Hersteller- und Geräte-ID
ReadVendorName	Liest Herstellername
ReadVendorText	Liest Herstellertext
ReadProductName	Liest Produktname
ReadProductID	Liest die Produkt ID
ReadProductText	Liest den Produkttext
ReadSerialNumber	Liest die Seriennummer
ReadHardwareRev	Liest die Hardwarerevision
ReadFirmwareRev	Liest die Firmwarerevision
ReadApplicationTag	Liest den ApplicationSpecificTag
ReadFunctionTag	Liest den FunctionTag
ReadLocationTag	Liest den LocationTag
ReadDiag	Liest die Diagnose aus
ReadDiagDetails	Liest die DetailedDiagnosis aus
WriteApplicationTag	Schreibt den ApplicationTag
WriteLocationTag	Schreibt den LocationTag
WriteFunctionTag	Schreibt den FunctionTag
WriteBackup	Löst den Datastorage aus

3.1.3 E_IoIProfileIdentifizier

Definiert die Profile, die ein IO-Link-Gerät haben kann. Ein Gerät kann mehrere Profile unterstützen.

Name	Wert	Beschreibung
NoProfileSupported	0x0000	Kein Profil wird unterstützt
FunctionClass	0x0001	Funktionsklassen werden unterstützt
Switching sensor profile type 1		
SSP_1_1	0x0002	Definiert den Fixed Switching Sensor mit dem Profil 0x8005 ohne 0x800C
SSP_1_2	0x0003	Definiert den Fixed Switching Sensor mit dem Profil 0x8005 und 0x800C
Switching sensor profile type 2		
SSP_2_1	0x0004	Definiert den Adjustable Switching Sensor mit dem Profil 0x8007 ohne 0x800C
SSP_2_2	0x0005	Definiert den Adjustable Switching Sensor mit dem Profil 0x8008 ohne 0x800C
SSP_2_3	0x0006	Definiert den Adjustable Switching Sensor mit dem Profil 0x8009 ohne 0x800C
SSP_2_4	0x0007	Definiert den Adjustable Switching Sensor mit dem Profil 0x8007 mit 0x800C
SSP_2_5	0x0008	Definiert den Adjustable Switching Sensor mit dem Profil 0x8008 mit 0x800C
SSP_2_6	0x0009	Definiert den Adjustable Switching Sensor mit dem Profil 0x8009 mit 0x800C
Measuring Device profile types 3		
SSP_3_1	0x000A	Definiert die Profile 0x800A ohne 0x800C
SSP_3_2	0x000B	Definiert die Profile 0x800B ohne 0x800C
SSP_3_3	0x000C	Definiert die Profile 0x800A mit 0x800C
SSP_3_4	0x000D	Definiert die Profile 0x800B mit 0x800C
Overview of Functionclasses		
Dev_Ident	0x8000	Mit diesem Profil müssen die Parameter Firmware Revision, ApplicationSpecificTag und ProductID zwingend vom IO-Link-Gerät unterstützt werden
Multi_Ch	0x8001	Definiert die Prozessdatenart des IO-Link-Geräts
PDV	0x8002	Definiert die Prozessdatenübertragung des Geräts.
Dev_Diag	0x8003	Mit diesem Profil müssen die Parameter Device Status und Detailed Device Status zwingend vom IO-Link-Gerät unterstützt werden
Teach_Ch	0x8004	
Fixed_SSC	0x8005	Fest schaltende Sensoren im Smart Sensor Profil sind Geräte, die genau ein binäres Ausgangssignal (Schaltsignal) unterstützen. Der Sollwert dieses Schaltausgangs wird während des Herstellungsprozesses vordefiniert und ist somit für die Anwendung fest vorgegeben.
Adjust_SSC	0x8006	
Teach_Single_Value	0x8007	Ein Sollwert wird durch einen "Teachpoint" (TP) definiert. Der Teach-In-Vorgang ist "statisch", d.h., der Messwert ist während des TeachIn konstant.
Teach_Two_Value	0x8008	Ein Sollwert wird durch zwei "Teachpoints" (TP) definiert.
Teach_Dyn	0x8009	"Dynamisches Teach-in" (innerhalb eines Zeitraums) Ein einzelner Sollwert oder beide Sollwerte eines SSC werden über erfasste Messwerte während eines bestimmten Zeitraums eingelernt. Der Teach-In ist "dynamisch", das heißt, der Messwert ist während des Teach-Ins nicht konstant. In der

Name	Wert	Beschreibung
		Regel werden die Minimal- und Maximalwerte innerhalb dieser Zeitspanne zur Festlegung der Sollwerte herangezogen.
Meas_Data_Ch_Std	0x800A	Die Funktionsklasse bietet eine standardisierte Prozessdatenstruktur und einige zusätzliche Informationen, wie die übertragenen Daten zu interpretieren sind, wie z.B. physikalische Einheit oder Messgrenzen. (Standardauflösung)
Meas_Data_Ch_High	0x800B	Die Funktionsklasse bietet eine standardisierte Prozessdatenstruktur und einige zusätzliche Informationen, wie die übertragenen Daten zu interpretieren sind, wie z.B. physikalische Einheit oder Messgrenzen. (Hohe Auflösung)
Sensor_Ctrl	0x800C	Das Deaktivierungssignal kann verwendet werden, um den Sensorwandler auszuschalten.
Example of the profile identification of an extended profile		
Ident_And_Diag	0x4000	Kombiniert die Profile 0x8000, 0x8002, 0x8003 und 0x8100
Visual_Localizations	0x8100	Mit diesem Profil müssen die Parameter FunctionTag und Location zwingend vom IO-Link-Gerät unterstützt werden

3.1.4 E_IolProfileIdentifizier_Bounds

Wertebereich der Profile und Funktionsklassen.

Name	Wert	Beschreibung
NoProfileSupported	0x0000	Kein Profil unterstützt
DeviceProfileID_Start	0x0001	Profilbereich Start
DeviceProfileID_End	0x3FFF	Profilbereich Ende
CommonApplicationProfileID_start	0x4000	CommonApplicationProfileID Start
CommonApplicationProfileID_end	0x7FFF	CommonApplicationProfileID Ende
FunctionClassID_start	0x8000	Funktionsklasse Bereich Start
FunctionClassID_end	0xBFFF	Funktionsklasse Bereich Ende
Reserved_start	0xC000	Reservierter Bereich Start
Reserved_end	0xFFFF	Reservierter Bereich Ende

3.1.5 ST_IolIdentificationObjects

Identifikationsobjekte als Struktur, die von dem Funktionsbaustein IOL_IdentifierAndDiagnosis [► 29] ausgegeben wird.

Name	Datentyp
nVendorID	WORD
nDeviceID	DWORD
sVendorname	STRING(64)
sVendorText	STRING(64)
sProductName	STRING(64)
sProductID	STRING(64)
sProductText	STRING(64)
sSerialNumber	STRING(16)
sHardwareRevision	STRING(64)
sFirmwareRevision	STRING(64)
sApplicationSpecificTag	STRING(64)

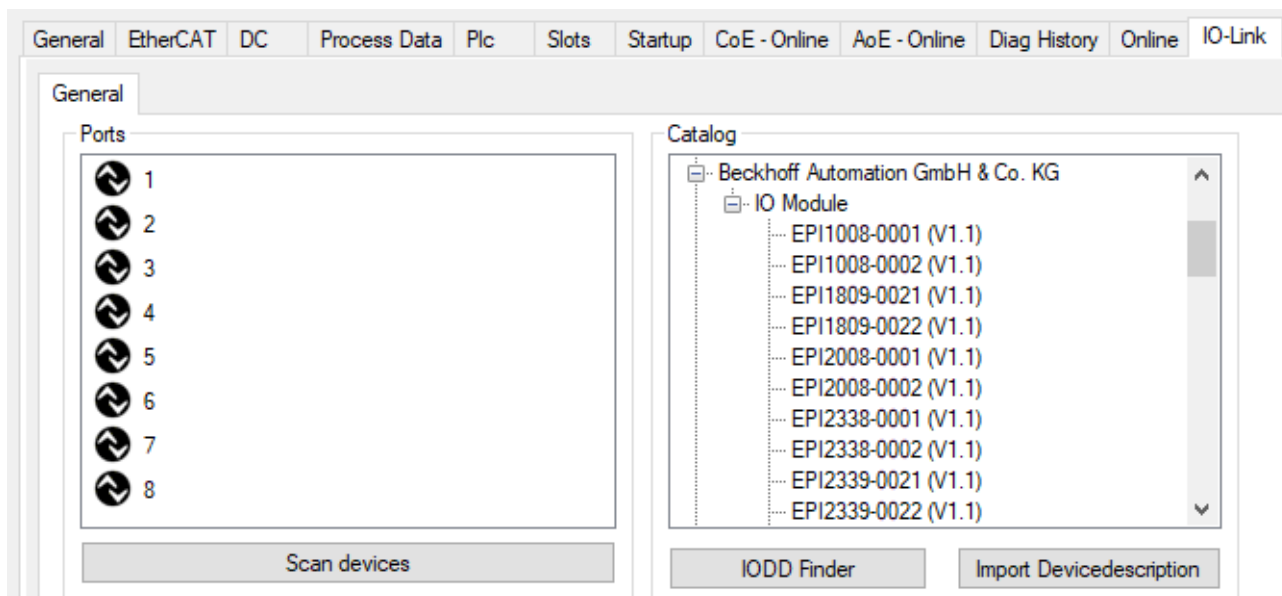
Name	Datentyp
sLocationTag	STRING(64)
sFunctionTag	STRING(64)

3.2 Geräte

Können im Zusammenhang mit dem State Channel verwendet werden, um den Zustand der Kommunikation festzustellen.

3.2.1 E_IolPort

Gibt die Nummer des Portes über eine Enumeration an.



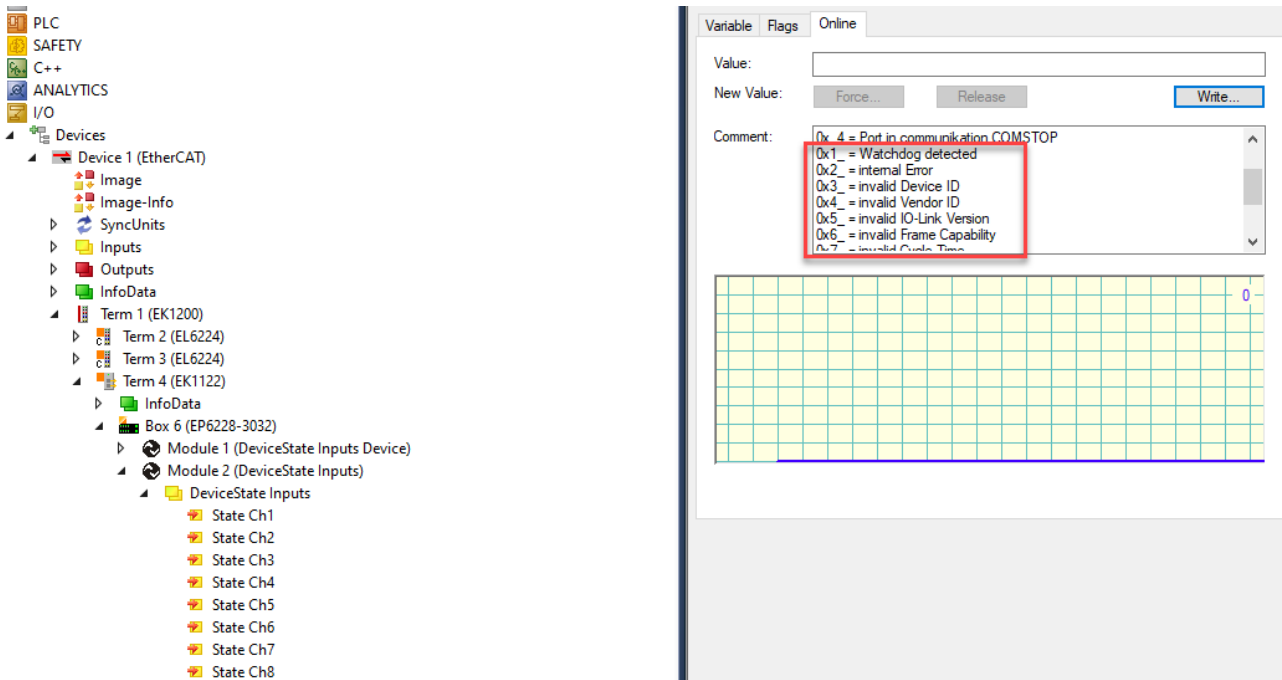
Name	Wert
Port1	1
Port2	2
Port3	3
Port4	4
Port5	5
Port6	6
Port7	7
Port8	8

Beispiel

```
fbIolSlave.Port:=E_IolPort.Port1;
```

3.2.2 E_IolPortError

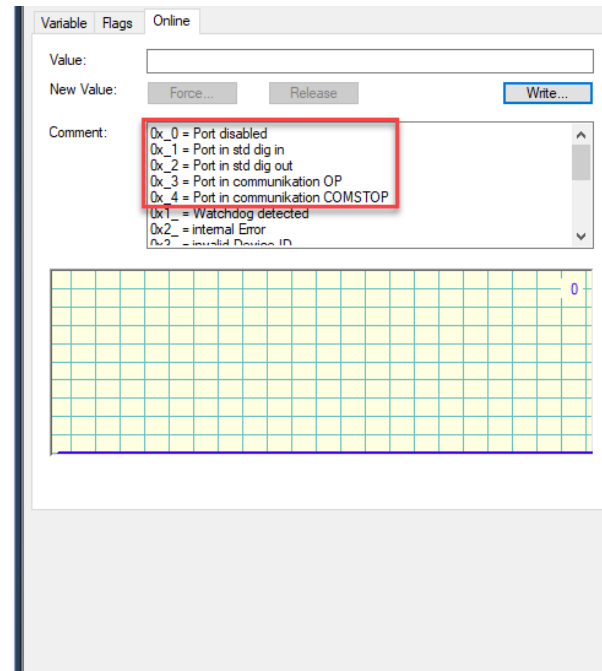
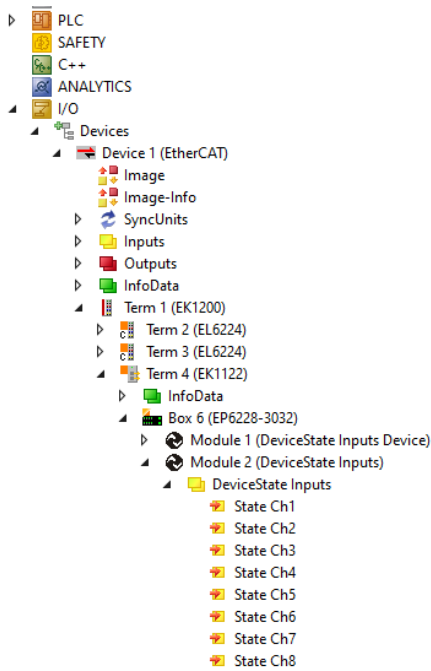
Gibt die Fehler der IO-Link Kommunikation des State Channels aus.



Name	Wert
NoError	0
WatchdogDetected	1
InternalError	2
InvalidDeviceID	3
InvalidVendorID	4
InvalidIOLinkVersion	5
InvalidFrameCapability	6
InvalidCycletime	7
InvalidPdInLength	8
InvalidPdOutLenght	9
NoDeviceDetected	10

3.2.3 E_IoIPortState

Information der Kommunikationsart des State Channels



Name	Wert
Disabled	0
stDigIn	1
stDigOut	2
CommunicationOP	3
CommunicationComstop	4

3.3 Spezifikation

3.3.1 E_IoIDeviceStatus

Gibt den Zustand des IO-Link-Geräts an und stammt aus dem Index [0x0024](#) [[▶ 70](#)].

Name	Wert	Beschreibung
Operating	0x00	IO-Link-Gerät funktioniert einwandfrei
MaintenanceRequired	0x01	Obwohl die Prozessdaten gültig sind, zeigt die interne Diagnose an, dass das Gerät kurz davor ist, seine Fähigkeit zur korrekten Funktion zu verlieren.
OutOfSpecification	0x02	Obwohl die Prozessdaten gültig sind, zeigt die interne Diagnose an, dass das Gerät außerhalb des spezifizierten Messbereichs oder der Umgebungsbedingungen arbeitet.
FunctionalCheck	0x03	Prozessdaten sind aufgrund von beabsichtigten Manipulationen am Gerät vorübergehend ungültig
Failure	0x04	Prozessdaten ungültig aufgrund einer Fehlfunktion des Geräts oder seiner Peripheriegeräte. Das Gerät ist nicht in der Lage, seine vorgesehene Funktion auszuführen.

3.3.2 E_IoIError

Enthält fest definierte [IO-Link Fehler Codes](#) [[▶ 74](#)].

Name	Wert	Beschreibung
NoError	0x0000	Keine Fehlermeldung
DeviceApplicationError	0x8000	Gerätespezifischer Applikationsfehler

Name	Wert	Beschreibung
IndexNotAvailable	0x8011	Der Index ist nicht verfügbar
SubindexNotAvailable	0x8012	Der Subindex ist nicht verfügbar
ServiceTemporarilyNotAvailable	0x8020	Ein Parameter für einen Lese- oder Schreibdienst ist nicht zugänglich, weil der aktuelle Zustand der Device-Anwendung nicht zugänglich ist.
ServiceTemporarilyNotAvailable_LocalControl	0x8021	Ein Parameter ist aufgrund eines laufenden lokalen Vorgangs am Gerät (z. B. Bedienung oder Parametrierung über ein bordseitiges Gerätebedienfeld) für einen Lese- oder Schreibdienst nicht zugänglich.
ServiceTemporarilyNotAvailable_DeviceControl	0x8022	Ein Lese- oder Schreibdienst ist aufgrund eines fernausgelösten Zustands der Geräteanwendung nicht zugänglich (z. B. Parametrierung während eines fernausgelösten Lernvorgangs oder Kalibrierung).
AccessDenied	0x8023	Ein Write-Dienst versucht, auf einen schreibgeschützten Parameter zuzugreifen. Oder ein Read-Dienst versucht, auf einen schreibgeschützten Parameter zuzugreifen.
ParameterValueOutOfRange	0x8030	Ein Schreibdienst auf einen Parameter, der außerhalb seines zulässigen Wertebereichs
ParameterValueAboveLimit	0x8031	Ein Schreibdienst greift auf einen Parameter oberhalb des angegebenen Wertebereichs zu.
ParameterValueBelowLimit	0x8032	Ein Schreibdienst greift auf einen Parameter unterhalb des angegebenen Wertebereichs zu.
ParameterLengthOverrun	0x8033	Der Inhalt eines Schreibdienstes für einen Parameter ist größer als die für den Parameter angegebene Länge. Oder ein Datenobjekt ist zu groß, um von der Geräteanwendung verarbeitet werden zu können (z. B. ISDU-Pufferbeschränkung).
ParameterLengthUnderrun	0x8034	Der Inhalt eines Schreibdienstes für einen Parameter ist kleiner als die für den Parameter angegebene Länge. Oder ein Datenobjekt ist zu klein, um von der Geräteanwendung verarbeitet werden zu können (z. B. ISDU-Pufferbeschränkung).
FunctionNotAvailable	0x8035	Einen Schreibdienst, der von der Geräteanwendung nicht unterstützt wird (zum Beispiel ein System Command mit einem nicht implementierten Wert).
FunctionTemporarilyUnavailable	0x8036	Ein Schreibdienst, der eine Device Funktion aufruft, die die aufgrund des aktuellen Zustands der Device-Anwendung (z. B. System Command) nicht verfügbar ist.
InvalidParameterSet	0x8040	die Einzelparameterübertragung gesendeten Werte nicht mit anderen Ist-Parametereinstellungen übereinstimmen (z. B. überlappende Sollwerte für eine Binärdateneinstellung).
InconsistentParameterSet	0x8041	Am Ende eines Blockparameter-Transfers mit ParamDownloadEnd oder ParamDownloadStore, wenn die Plausibilitätsprüfung Inkonsistenzen ergibt
ApplicationNoReady	0x8082	Wenn ein Lese- oder Schreibdienst aufgrund einer vorübergehend nicht verfügbaren Anwendung (z. B. periphere Steuerungen während des Starts) verweigert wird.

3.3.3 E_IoSystemcommand

Gibt die Werte für den System Command vor.

Name	Wert	Beschreibung
ParamUploadStart	0x01	Startet Parameter Upload
ParamUploadEnd	0x02	Stoppt Parameter Upload

Name	Wert	Beschreibung
ParamDownloadStart	0x03	Startet Parameter-Download
ParamDownloadEnd	0x04	Stoppt Parameter-Download
ParamDownloadStore	0x05	Finalisiert die Parametrisierung und startet den Datastorage
DevcieReset	0x80	Reset zum Initialisierungswert
ApplicationReset	0x81	Applikationsspezifische Parameter werden zurückgesetzt
RestoreFactorySettings	0x82	Werkseinstellungen

3.3.4 ST_IoAccessLocks

Struktur gibt die AccessLocks vor. Die AccessLocks bestimmen welcher Zugriff von dem IO-Link Gerät physikalisch erlaubt sind.

Name	Datentyp
bParameterAccess	BIT
bDataStorage	BIT
bLocalParametrization	BIT
bLocalUserInterface	BIT
bReserved1	BIT
bReserved2	BIT
bReserved3	BIT
bReserved4	BIT
nReserved1	BYTE

3.3.5 ST_IoDetailedDeviceStatusEntry

Struktur des Detailed Device Status Entry

Name	Datentyp
nEventqualifier	BYTE
nEventCode	WORD

3.3.6 ST_IoDirectParameter1

Definiert die DirectParameter1 als Struktur.

Name	Datentyp
Reserved1	BYTE
MasterCycleTime	BYTE
MinCycleTime	BYTE
FrameCapability	BYTE
VersionID	BYTE
ProcessDataInLen	BYTE
ProcessDataOutLen	BYTE
VendorID1	BYTE
VendorID2	BYTE
DeviceID1	BYTE
DeviceID2	BYTE
DeviceID3	BYTE
Reserved2	BYTE
Reserved3	BYTE
Reserved4	BYTE

Name	Datentyp
Reserved5	BYTE

3.3.7 ST_IolDirectParameter2

Struktur definiert die DirectParameter2

Name	Datentyp
DeviceSpecificParameter01	BYTE
DeviceSpecificParameter02	BYTE
DeviceSpecificParameter03	BYTE
DeviceSpecificParameter04	BYTE
DeviceSpecificParameter05	BYTE
DeviceSpecificParameter06	BYTE
DeviceSpecificParameter07	BYTE
DeviceSpecificParameter08	BYTE
DeviceSpecificParameter09	BYTE
DeviceSpecificParameter10	BYTE
DeviceSpecificParameter11	BYTE
DeviceSpecificParameter12	BYTE
DeviceSpecificParameter13	BYTE
DeviceSpecificParameter14	BYTE
DeviceSpecificParameter15	BYTE
DeviceSpecificParameter16	BYTE

3.3.8 T_IolDetailedDeviceStatus

Ereignishistorie des IO-Link-Geräts

```
TYPE T_IolDetailedDeviceStatus :
ARRAY[0..63] OF ST_IolDetailedDeviceStatusEntry;
END_TYPE
```

3.3.9 T_IolProfileCharacteristics

Profilcharakteristik des IO-Link-Geräts

```
TYPE T_IolProfileCharacteristics :
ARRAY[0..31] OF WORD;
END_TYPE
```

3.3.10 T_IolMaxOctedString

Definition der maximalen String Größe

```
TYPE T_IolMaxOctedString :
ARRAY[0..231] OF BYTE;
END_TYPE
```

3.3.11 T_IolMaxString

Definiert die maximale Größe eines IO-Link Strings.

```
TYPE T_IolMaxString : STRING(232);
END_TYPE
```

3.3.12 ST_IoIIndex

Gibt den Index und Subindex an dem IO-Link Parameter an.

Name	Datentyp
nIndex	WORD
nSubindex	BYTE

3.4 Smart Sensor Profile

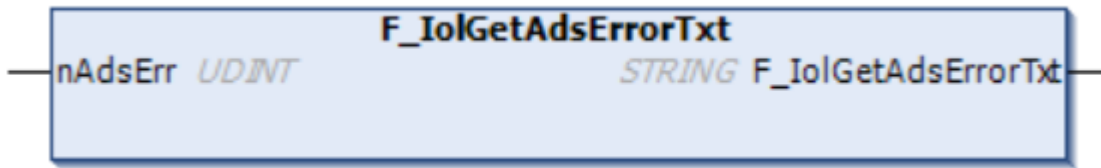
3.4.1 E_IoIMdc_ValueStatus

Definiert den Status der Prozessdaten

Name	Wert	Beschreibung
OK	0	Prozessdaten sind gültig
PDInvalid	1	Prozessdaten sind ungültig
NoData	2	Keine Daten
OutOfRangePos	3	Prozessdaten sind außerhalb des Wertebereichs
OutOfRangeNeg	4	Prozessdaten sind außerhalb des Wertebereichs
StateNotDefined	5	Undefinierter Status

4 Funktionen

4.1 F_IolGetAdsErrorTxt



Funktion wandelt einen ADS-Fehler in Textform um.

📥 Eingänge

Name	Typ	Beschreibung
nAdsErr	UDINT	ADS Return Codes

📤 Ausgänge

Name	Typ	Beschreibung
F_IolGetAdsErrorTxt	STRING	Zeigt den ADS Return Code als Textstring an.

Beispiel

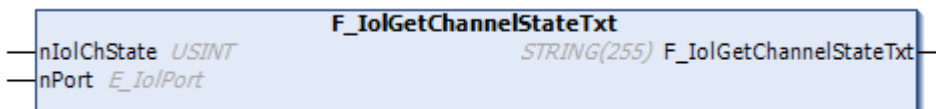
```
VAR
  hr          :HRESULT;
  nADSError  :E_AdsErr := E_AdsErr.NOERR;
  sADSErrorText :STRING;
END_VAR

nAdsError := F_IolGetAdsError(hr);
sADSErrorText := F_IolGetAdsErrorTxt(nAdsError);
```

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

4.2 F_IolGetChannelStateTxt



Funktion wandelt State Channel und Port in Textform um.

📥 Eingänge

Name	Typ	Beschreibung
nIolChState	USINT	IO-Link Channel State
nPort	E_IolPort [▶_12]	IO-Link Port

Ausgänge

Name	Typ	Beschreibung
F_IoIGetChannelStateTxt	String(255)	Wandelt den Channel State in eine Textinformation um.

Beispiel

```

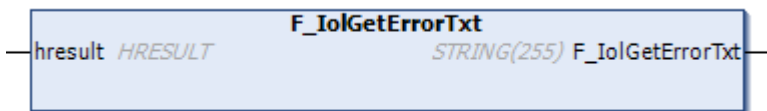
VAR
  nStateCh2 AT%I*      :USINT;
  nPort             :E_IoIPort:=E_IoIPort.Port2;
  sChannelStateText :STRING(255);
END_VAR

sChannelStateText := F_IoIGetChannelStateTxt(nStateCh2,nPort);
    
```

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

4.3 F_IoIGetErrorTxt



Gibt den IO-Link-Fehler als String Format in Textform zurück.

Eingänge

Name	Typ	Beschreibung
hresult	HRESULT	Rückgabewert

Ausgänge

Name	Typ	Beschreibung
F_IoIGetErrorTxt	String(255)	Zeigt den IO-Link-Fehler als String

Beispiel

```

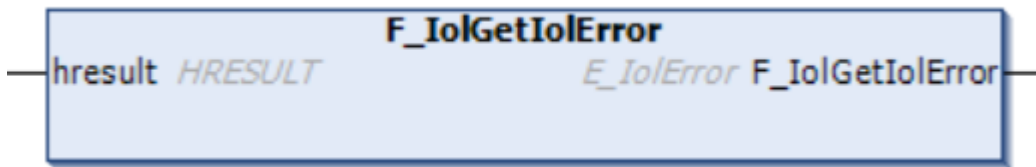
VAR
  hr      :HRESULT;
  sErrorText :STRING;
END_VAR

sErrorText := F_IoIGetErrorTxt(hr);
    
```

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

4.4 F_IolGetIolError



Funktion gibt einen IO-Link-Fehlercode als [E_IolError \[► 14\]](#) wieder.

👉 Eingänge

Name	Typ	Beschreibung
hresult	HRESULT	Rückgabewert

👈 Ausgänge

Name	Typ	Beschreibung
F_IolGetIolError	E_IolError [► 14]	IO-Link-Fehlercode als Enum

Beispiel

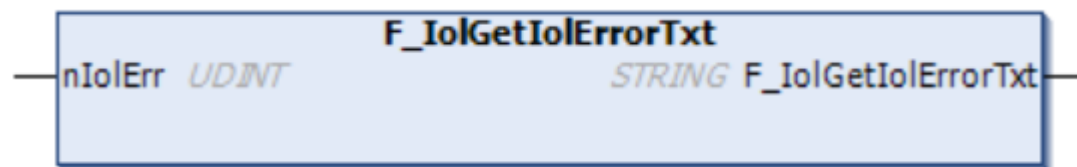
```
VAR
  hr           :HRESULT;
  nIolError   :E_IolErr := E_IolErr.NOERR;
END_VAR

nIolError := F_IolGetIolError(hr);
```

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

4.5 F_IolGetIolErrorTxt



Die Funktion gibt den IO-Link-Fehlercode als String wieder

👉 Eingänge

Name	Typ	Beschreibung
nIolErr	UDINT	IO-Link-Fehler

👈 Ausgänge

Name	Typ	Beschreibung
F_IolGetIolErrorTxt	String	IO-Link-Fehler Code als Text

Beispiel

```

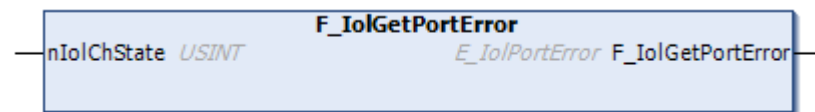
VAR
  hr           :HRESULT;
  nIoLError   :E_IolErr := E_IolErr.NOERR;
  sIoErrorText :STRING;
END_VAR

nIoLError := F_IolGetIoLError(hr);
sIoErrorText := F_IolGetIoErrorTxt(nIoLError);
    
```

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

4.6 F_IolGetPortError



Funktion gibt den State Channel als E_IolPortError aus.

Eingänge

Name	Typ	Beschreibung
nIoChState	USINT	IO-Link Channel State

Ausgänge

Name	Typ	Beschreibung
E_IolGetPortError	E_IolPortError > 12	Anzeige IO-Link Port Error

Beispiel

```

VAR
  nStateChl AT%I* :USINT;
  nIoPortError :E_IolPortError;
END_VAR

nIoPortError := F_IolGetPortError(nStateChl);

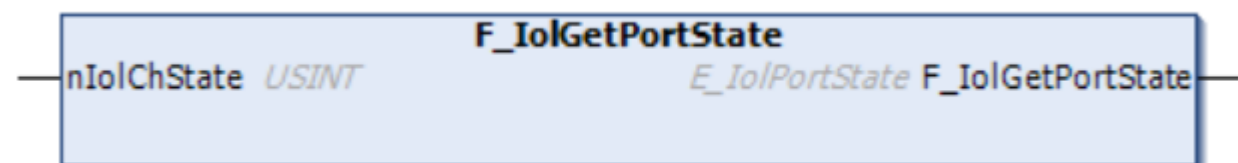
nIoPortError.NoDeviceDe := F_IolGetPortError(nStateChl.160);
    
```

NoDeviceDetected
'E_IolPortError.NoDeviceDetected' represents raw value '10'

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

4.7 F_IolGetPortState



Funktion wandelt die Variable State Ch in das E_IolPortState um.

Der Datentyp E_IolPortstate kann anschließend weiter weiterverwendet werden.

Eingänge

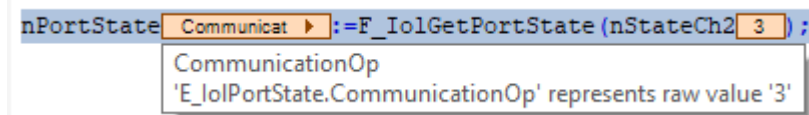
Name	Typ	Beschreibung
nIolChState	USINT	IO-Link Channel State

Ausgänge

Name	Typ	Beschreibung
E_IolGetPortState	E_IolPortState [▶_13]	Zeigt den Port State als Enum an

Beispiel

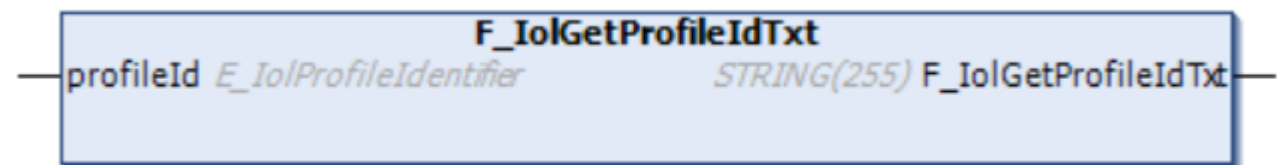
```
VAR
  nStateCh2 AT%I*      :USINT;
  nPortState          :E_IolPortState;
END_VAR
nPortState := F_IolGetPortState(nStateCh2);
```



Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

4.8 F_IolGetProfileIdTxt



Wandelt den Typen E_IolProfileIdentifier in ein Textformat um.

Eingänge

Name	Typ	Beschreibung
profileId	E_IolProfileIdentifier [▶_10]	IO-Link Profile

Ausgänge

Name	Typ	Beschreibung
F_IolGetProfileIdTxt	String(255)	IO-Link Profile als Textform

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

4.9 F_IolIsBusy



Funktion gibt an, ob ein Baustein noch in Bearbeitung ist.

👉 Eingänge

Name	Typ	Beschreibung
hresult	HRESULT	Rückgabewert

👉 Ausgänge

Name	Typ	Beschreibung
F_IolIsBusy	BOOL	True wenn Baustein noch in Bearbeitung ist

Beispiel

```

VAR
    hr                :HRESULT;
    fbIIolStdVarSlave :FB_IolStdVarSlave;
    nRead             :DWORD;
END_VAR

hr := fbIIolStdVarSlave.read(16#0019, 0, ADR(nRead), SIZEOF(nRead));
IF NOT F_IolIsBusy(hr) THEN
    IF FAILED(hr) THEN
        //Error Handling
    ELSE
        //Success
    END_IF
END_IF

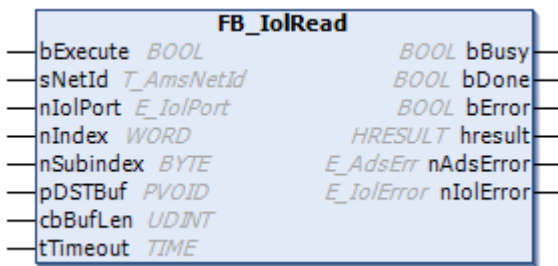
```

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

5 Funktionsbausteine

5.1 FB_IoIRead



Funktionsbaustein zum Lesen von IO-Link Parametern in einer nicht objektorientierten Lösung.

Eingänge

Name	Typ	Beschreibung
bExecute	BOOL	Steigende Flanke zum Starten des Funktionsblocks
sNetid	T-AmsNetid	AMS NETID des IO-Link Masters
nIoIPort	E_IoIPort [▶ 12]	IO-Link Port des IO-Link-Geräts
nIndex	WORD	Index des IO-Link-Geräteparameters
nSubindex	BYTE	Subindex des IO-Link-Geräteparameters
pDSTBuf	PVOID	Zeiger auf Zielpuffer
cbBufLen	UDINT	Größe der zu lesenden Datenbytes
tTimeout	TIME	Maximal zulässige Ausführungszeit

Ausgänge

Name	Typ	Beschreibung
bBusy	BOOL	Busy-Flag des Funktionsblocks
bDone	BOOL	Done-Merker des Funktionsblocks
bError	BOOL	Fehlermerker des Funktionsblocks
hResult	HRESULT	Rückgabewert dieses Funktionsblocks
nADSError	E_AdsErr	ADS-Fehlercode
nIoIError	E_IoIError [▶ 14]	IO-Link-Fehlercode

Beispiel

Der folgende FB ist Teil des IoLink Beispielprojekts. Weitere Informationen finden Sie im Kapitel [Beispiele](#) [[▶ 69](#)].

```
FUNCTION_BLOCK FB_Sample_01_IoIRead
VAR_INPUT
    bExecute      :BOOL := FALSE;
END_VAR

VAR_OUTPUT
    bBusy         :BOOL := FALSE;
    bDone         :BOOL := FALSE;
    bError        :BOOL := FALSE;
    nAdsError     :E_AdsErr := E_AdsErr.NOERR;
    nIoIError     :E_IoIError := E_IoIError.NoError;
    sErrorTxt     :STRING(255) := '';
    sApplicationTag :STRING(32) := '';
END_VAR
```

```

VAR
  bExecuteOld      :BOOL;
  nState           :INT := 0;
  fbIolRead        :FB_IolRead;
END_VAR

////
// Execute/Busy State Machine
//
IF bExecute AND NOT bExecuteOld AND NOT bBusy THEN
  bBusy := TRUE;
  nState := 10; // READ
ELSIF NOT bExecute THEN
  bBusy := FALSE;
  bError := FALSE;
  bDone := FALSE;
  nAdsError := E_AdsErr.NOERR;
  nIolError := E_IolError.NoError;
  sErrorTxt := '';
END_IF
bExecuteOld := bExecute;

////
// FB State Machine
//
CASE nState OF
// IDLE
0:;
// READ
10:
  sApplicationTag := '';
  fbIolRead(bExecute := FALSE);
  fbIolRead.sNetId := F_CreateAmsNetId(GVL_IO.sTerm2AmsNetId);
  fbIolRead.nIolPort := E_IolPort.Port1;
  fbIolRead.pDSTBuf := ADR(sApplicationTag);
  fbIolRead.cbBufLen := SIZEOF(sApplicationTag);
  fbIolRead.nIndex := 16#18;
  fbIolRead.nSubindex := 0;
  fbIolRead.bExecute := TRUE;
  nState := 20; // WAIT_FOR_DONE

// WAIT_FOR_DONE
20:
  IF fbIolRead.bDone THEN
    bDone := TRUE;
    bBusy := FALSE;
    nState := 0; // IDLE
  END_IF
END_CASE

////
// call FBs
//
fbIolRead();

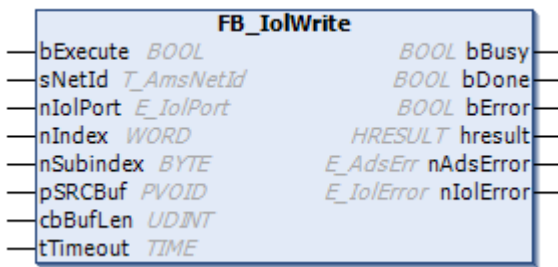
////
// Error Handling
//
IF NOT bError AND bBusy THEN
  IF fbIolRead.bError THEN
    nState := 0;
    bBusy := FALSE;
    bError := TRUE;
    nAdsError := fbIolRead.nAdsError;
    nIolError := fbIolRead.nIolError;
    sErrorTxt := F_IolGetErrorTxt(fbIolRead.hresult);
  END_IF
END_IF

```

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

5.2 FB_IolWrite



Funktionsbaustein zum Schreiben von IO-Link Parametern in einer nicht objektorientierten Lösung.

Eingänge

Name	Typ	Beschreibung
bExecute	BOOL	Steigende Flanke zum Starten des Funktionsblocks
sNetid	T-AmsNetid	AMS NETID des IO-Link Masters
nIolPort	E_IolPort [▶ 12]	IO-Link Port des IO-Link-Geräts
nIndex	WORD	Index des IO-Link-Geräteparameters
nSubindex	BYTE	Subindex des IO-Link-Geräteparameters
pSCRBuf	PVOID	Zeiger auf Quellpuffer
cbBufLen	UDINT	Größe der zu lesenden Datenbytes
tTimeout	TIME	Maximal zulässige Ausführungszeit

Ausgänge

Name	Typ	Beschreibung
bBusy	BOOL	Busy-Flag des Funktionsblocks
bDone	BOOL	Done-Merker des Funktionsblocks
bError	BOOL	Fehlermerker des Funktionsblocks
hResult	HRESULT	Rückgabewert dieses Funktionsblocks
nADSError	E_AdsErr	ADS-Fehlercode
nIolError	E_IolError [▶ 14]	IO-Link-Fehlercode

Beispiel

Der folgende FB ist Teil des IoLink Beispielprojekts. Weitere Informationen finden Sie im Kapitel [Beispiele \[▶ 69\]](#).

```

FUNCTION_BLOCK FB_Sample_02_IolWrite
VAR_INPUT
    bExecute          : BOOL := FALSE;
    sApplicationTag   : STRING(32) := '';
END_VAR
VAR_OUTPUT
    bBusy             : BOOL := FALSE;
    bDone             : BOOL := FALSE;
    bError            : BOOL := FALSE;
    nAdsError         : E_AdsErr := E_AdsErr.NOERR;
    nIolError         : E_IolError := E_IolError.NoError;
    sErrorTxt         : STRING(255) := '';
END_VAR
VAR
    bExecuteOld      : BOOL;
    nState            : INT := 0;
    fbIolWrite        : FB_IolWrite;
END_VAR
    
```

```

////
// Execute/Busy State Machine
//
IF bExecute AND NOT bExecuteOld AND NOT bBusy THEN
    bBusy := TRUE;
    nState := 10; // WRITE
ELSIF NOT bExecute THEN
    bBusy := FALSE;
    bError := FALSE;
    bDone := FALSE;
    nAdsError := E_AdsErr.NOERR;
    nIolError := E_IolError.NoError;
    sErrorTxt := '';
END_IF
bExecuteOld := bExecute;

////
// FB State Machine
//
CASE nState OF

// IDLE
0:;

// WRITE
10:
    sApplicationTag := '';

    fbIolWrite(bExecute := FALSE);
    fbIolWrite.sNetId := F_CreateAmsNetId(GVL_IO.sTerm2AmsNetId);
    fbIolWrite.nIolPort := GVL_IO.nTerm2EPI1008;
    fbIolWrite.pSRCBuf := ADR(sApplicationTag);
    fbIolWrite.cbBufLen := INT_TO_UDINT(LEN(sApplicationTag));
    fbIolWrite.nIndex := 16#18;
    fbIolWrite.nSubindex := 0;
    fbIolWrite.bExecute := TRUE;

    nState := 20; // WAIT_FOR_DONE

// WAIT_FOR_DONE
20:
    IF fbIolWrite.bDone THEN
        bDone := TRUE;
        bBusy := FALSE;
        nState := 0; // IDLE
    END_IF
END_CASE

////
// call FBs
//
fbIolWrite();

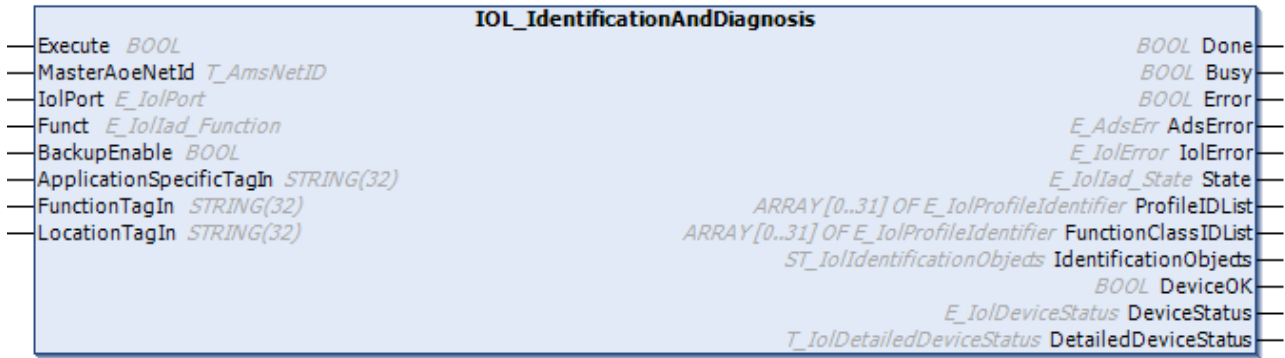
////
// Error Handling
//
IF NOT bError AND bBusy THEN
    IF fbIolWrite.bError THEN
        nState := 0;
        bBusy := FALSE;
        bError := TRUE;
        nAdsError := fbIolWrite.nAdsError;
        nIolError := fbIolWrite.nIolError;
        sErrorTxt := F_IolGetErrorTxt(fbIolWrite.hresult);
    END_IF
END_IF

```

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

5.3 IOL_IdentificationAndDiagnosis



Funktionsbaustein zur Identifikation und Diagnose eines IO-Link Devices. Dieser Funktionsbaustein wird durch das Common IO-Link Profile implementiert. Der ApplicationSpecificTag, der FunctionTag und der LocationTag eines IO-Link Devices kann geschrieben werden. Die Funktionsklassen und ProfildIDs können ausgelesen werden, die ein Gerät besitzt. Zusätzlich gibt der Funktionsbaustein eine Diagnose über den Zustand eines Geräts und die Identifikation wie Vendor Name, Vendor ID, Device Name, Device ID.

Eingänge

Name	Typ	Beschreibung
Execute	BOOL	Ein Trigger bewirkt, dass die mit der Variable Function ausgewählte Funktion ausgeführt wird:
MasterAoENetid	T_AmsNetid	AMS NETID des IO-Link Masters
IolPort	E_IolPort [▶ 12]	IO-Link Port des IO-Link-Geräts
Func	E_Iollad_Function [▶ 9]	Diese Variable wählt die Funktionalität aus, die durch einen Request ausgelöst werden soll: 0 = no_func Ein Request wird vernachlässigt, es wird keine Funktion ausgeführt 1 = rd_all Ein Request startet das Rücklesen der aktuellen Identifikations und Diagnoseparameterwerte aus dem Device. 2 = rd_diag Ein Request startet das Rücklesen der aktuellen Diagnoseparameterwerte durch Lesen von DeviceStatus und DetailedDeviceStatus aus dem Gerät. 3 = wr_ident Ein Request bewirkt, dass ein zuvor angelegter Wert für ApplicationSpecificTagIn, LocationTagIn und FunctionTagIn in das Gerät geschrieben wird.
BackupEnable	BOOL	Diese Variable konfiguriert das Verhalten des FBs im Falle der angeforderten Funktion wr_ident. "true" = aktiviert Der Backup-Mechanismus wird durch den FB ausgelöst. "false" = deaktiviert Der Backup-Mechanismus wird nicht durch den FB ausgelöst.
ApplicationSpecificIN	String(32)	Parameter des IO-Link-Geräts

Name	Typ	Beschreibung
FunctionTagIN	String(32)	Parameter des IO-Link-Geräts
LoactionTagIN	String(32)	Parameter des IO-Link-Geräts

Ausgänge

Name	Typ	Beschreibung
Done	BOOL	Das Signal ist gesetzt, wenn der FB eine angeforderte Operation abgeschlossen hat.
Busy	BOOL	Das Signal wird gesetzt, wenn der FB eine angeforderte Operation ausführt.
Error	BOOL	Das Signal wird gesetzt, wenn bei der Ausführung eines angeforderten Vorgangs ein Fehler aufgetreten ist.
AdsError	E_ADSErr	ADS-Fehlercode
IoLError	E_IolError [▶ 14]	IO-Link-Fehler
State	E_Iollad_State [▶ 9]	Der Wert stellt den aktuellen Status des FB-Betriebs und der ausgeführten Funktionen dar. Der Inhalt ist systemspezifisch und enthält die Statusinformationen.
ProfileIDList	ARRAY [0..31] of E_IolProfileIdentifier [▶ 10]	Liste der vom Gerät unterstützten ProfileIDs
FunctionClassIDList	ARRAY [0..31] of E_IolProfileIdentifier [▶ 10]	Liste der vom Gerät unterstützten FunctionClassIDs
IdentificationObjects	ST_IolIdentificationObjects [▶ 11]	Strukturierte Liste der Identifikationsobjekte, für weitere Details.
DeviceOK	BOOL	Das Signal ist gesetzt, wenn keine weiteren Diagnoseinformationen verfügbar sind, es ist false, wenn weitere Informationen bei DeviceStatus und DetailedDeviceStatus verfügbar sind.
DeviceStatus	E_IolDeviceStatus [▶ 14]	Status des IO-Link-Geräts
DetailedDeviceStatus	T_IolDetailedDeviceStatus [▶ 17]	ARRAY [0..63] von ST_DetailedDeviceStatusEntry

Beispiel

Der folgende FB ist Teil des IoLink Beispielprojekts. Weitere Informationen finden Sie im Kapitel [Beispiele](#) [[▶ 69](#)].

```

FUNCTION_BLOCK FB_Sample_06_IdentificationAndDiagnosis
VAR_INPUT
    sApplicationTagIn      : STRING(32) := '';
    sFunctionTagIn        : STRING(32) := '';
    sLocationTagIn        : STRING(32) := '';
    bExecute              : BOOL;
END_VAR
VAR_OUTPUT
    bBusy                 : BOOL := FALSE;
    bDone                 : BOOL := FALSE;
    bError                : BOOL := FALSE;

    sIoLError             : STRING;
    sAdsError             : STRING;
    aProfileIDList        : ARRAY [0..31] OF E_IolProfileIdentifier;
    aFunctionClassIDList  : ARRAY [0..31] OF E_IolProfileIdentifier;
    stIdentObjects        : ST_IolIdentificationObjects;
    bDeviceOK             : BOOL;
    nDeviceStatus         : E_IolDeviceStatus;
    DetailedDeviceStatus  : T_IolDetailedDeviceStatus;
END_VAR
VAR
    bExecuteOld           : BOOL;

```

```

    nState          : INT := 0;
    IolDiagAndIdent : IOL_IdentificationAndDiagnosis;
END_VAR
////
// Execute/Busy State Machine
//
IF bExecute AND NOT bExecuteOld AND NOT bBusy THEN
    bBusy := TRUE;
    nState := 10; // START
ELSIF NOT bExecute THEN
    bBusy := FALSE;
    bError := FALSE;
    bDone := FALSE;
    sIolError:='';
    sADSError:='';
END_IF
bExecuteOld := bExecute;

CASE nState OF

// IDLE
0:;

// START
10:
    IolDiagAndIdent(Execute := FALSE);

    IolDiagAndIdent.MasterAoeNetId := F_CreateAmsNetId(GVL_IO.sTerm2AmsNetId);
    IolDiagAndIdent.IolPort := E_iolport.Port4;
    IolDiagAndIdent.Funct := E_IolIad_Function.ReadAll;
    IolDiagAndIdent.ApplicationSpecificTagIn := sApplicationTagIn;
    IolDiagAndIdent.FunctionTagIn := sFunctionTagIn;
    IolDiagAndIdent.LocationTagIn := sLocationTagIn;

    IolDiagAndIdent.Execute := TRUE;
    nState := 20; // WAIT FOR DONE

// WAIT FOR DONE
20:
    IF NOT IolDiagAndIdent.Busy AND NOT IolDiagAndIdent.Error THEN
        nState := 30; // DONE
    END_IF

// DONE
30:
    aProfileIDList := IolDiagAndIdent.ProfileIDList;
    aFunctionClassIDList := IolDiagAndIdent.FunctionClassIDList;
    stIdentObjects := IolDiagAndIdent.IdentificationObjects;
    bDeviceOK := IolDiagAndIdent.DeviceOK;
    nDeviceStatus := IolDiagAndIdent.DeviceStatus;
    DetailedDeviceStatus := IolDiagAndIdent.DetailedDeviceStatus;

    IolDiagAndIdent.Execute := FALSE;
    nState := 0; // INIT

END_CASE

////
// call FBs
//

IolDiagAndIdent();

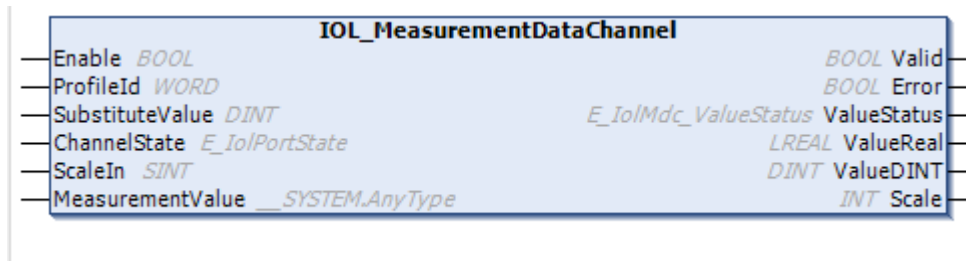
////
// Error Handling
//
IF NOT bError AND bBusy THEN
    IF IolDiagAndIdent.Error THEN
        nState := 0;
        bBusy := FALSE;
        bError := TRUE;
        sIolError := F_IolGetIolErrorTxt(ioldiagandident.IolError);
        sAdsError := F_IolGetadsErrorTxt(ioldiagandident.AdsError);
    END_IF
END_IF

```

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

5.4 IOL_MeasurementDataChannel



Der Messdatenkanal definiert die Prozessdatenstruktur, Funktionen und Darstellung von Messsensoren. Der Funktionsblock wird definiert, um abgeleitete Statussignale bereitzustellen und eine standardisierte Schnittstelle für Anwenderprogramme zu ermöglichen.

Eingänge

Name	Typ	Beschreibung
Enable	BOOL	Aktiviert die Verwendung des Funktionsblocks
ProfileID	WORD	Ausgewählte ProfilID, bzw. Prozessdatenlayout 1 = SSP 3.1 = Profil 0x000A 2 = SSP 3.2 = Profil 0x000B 3 = SSP 3.3 = Profil 0x000C 4 = SSP 3.4 = Profil 0x000D
SubstituteValue	DINT	Der angegebene Wert wird auf ValueReal und ValueDINT angewendet, wenn ValueStatus ungleich 0 ist.
ChannelState	E_IolPortState ▶ 13	
ScaleIN	SINT	Skala des IO-Link-Geräts
ProcessData	ANY_INT	Messwert

Ausgänge

Name	Typ	Beschreibung
Valid	BOOL	Bei "true" sind die gelieferten Werte gültig und können für weitere Berechnungen verwendet werden.
Error	BOOL	Bei "true" wird ein interner Fehler erkannt und weitere Informationen werden vom Funktionsblock bereitgestellt.
ValueStatus	E_IoLMDC_ValueStatus ▶ 18	Liefert Fehlercodes
ValueReal	LREAL	Prozessdaten im Real-Format zur Auswertung in der SPS
ValueINT	DINT	Prozessdaten im Dint-Format für die Auswertung in der SPS
Scale	INT	Prozessdaten Skalierungsfaktor

Beispiel

Der folgende FB ist Teil des IoLink Beispielprojekts. Weitere Informationen finden Sie im Kapitel [Beispiele ▶ 69](#).


```

FUNCTION_BLOCK FB_Sample_07_MeasurementDataChannel
VAR_INPUT
    nStateChannel    AT%I*:USINT;
    nScale           AT%I*:SINT;
    MVDistance       AT%I*:INT;
END_VAR
VAR_OUTPUT
    bValid           :BOOL;
    bError           :BOOL;
    nValueStatus     :E_IolMdc_ValueStatus;
    nValueDINT       :DINT;
    nValueREAL       :REAL;
    nScaleOut        :INT;
END_VAR
VAR
    fbMeasurement:IOL_MeasurementDataChannel_INT;
END_VAR
    
```

```

fbMeasurement(
    Enable := TRUE,
    SubstituteValue := 500,
    ChannelState := nStateChannel,
    ScaleIn := nScale ,
    MeasurementValue := MVDistance,
    Valid=>bValid,
    Error=>bError,
    ValueStatus=>nValueStatus,
    ValueReal=>nValueREAL,
    ValueDINT=>nValueDINT
    Scale=>nScaleOut );
    
```

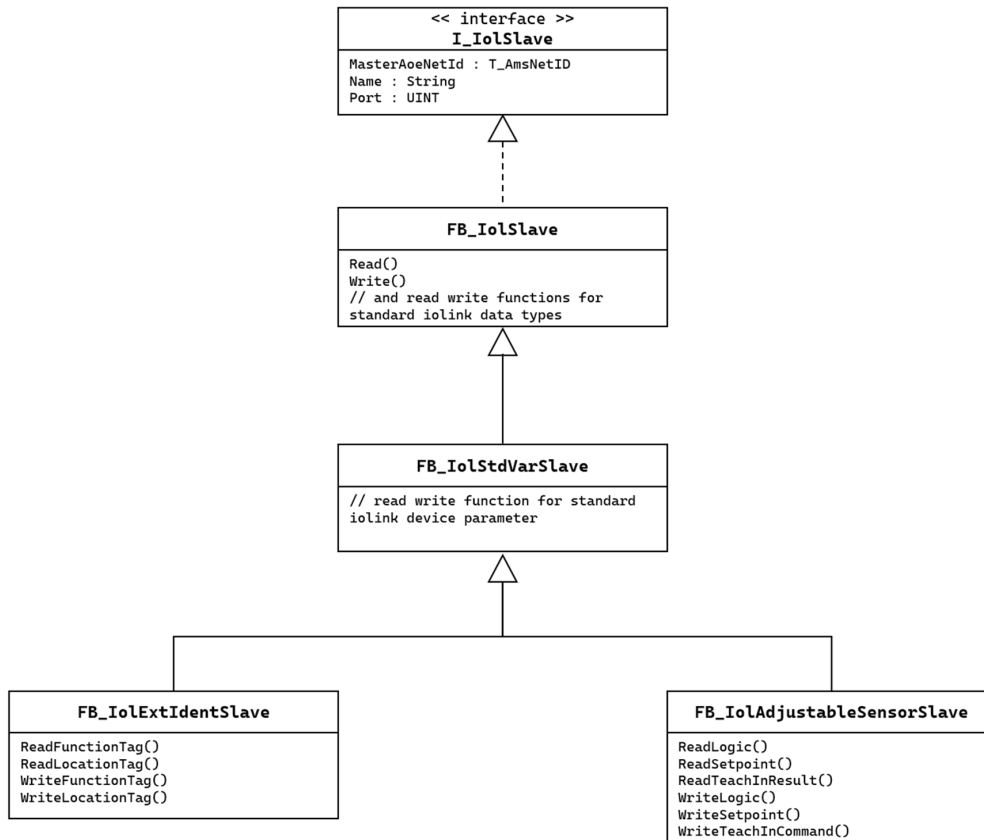
Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_loLink ab v3.3.4.0

6 Objektorientierte Funktionsbausteine

Hier sind die objektorientierten Funktionsbausteine aufgeführt. Ein direkter Aufruf des Funktionsbausteins ist nicht mehr nötig. Stattdessen konfigurieren Sie die Eigenschaften und rufen die Methoden direkt auf.

Das folgende UML-Diagramm zeigt den Aufbau und die Vererbung.



6.1 I_IoSlave

Das Interface definiert alle wichtigen Parameter, um mit einem IO-Link Gerät azyklisch zu kommunizieren.

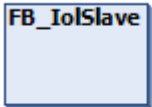
Schnittstellen

Typ	Beschreibung
I_IoSlave	Definiert IO-Link-Gerät

Eigenschaften

Name	Typ	Zugriff	Definitionsort	Initialwert	Beschreibung
MasterAoENetid	T_AmsNetid	Get	Lokal		AMS Netid vom IO-Link Master
Name	String(255)	Get, Set	Lokal		Name
Port	UINT	Get, Set	Lokal		Port des IO-Link-Geräts

6.2 FB_IOISlave



Funktionsbaustein zum Auslesen von IO-Link Parametern.

Beispiel

```

VAR
  bInitDone           : BOOL;
  sTerm3AmsNetId AT%I* : AMSNETID;
  nTerm2EPI2328      : E_IoLPort := E_IolPort.Port2;
  fbIOLSlave         : FB_IolSlave;
END_VAR

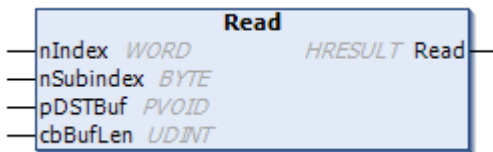
IF NOT bInitDone THEN
  bInitDone           := TRUE; //Assigns it only once
  fbIolSlave.MasterAoeNetId := F_CreateAmsNetId(sTerm3AmsNetId) //Creates NETID String
  fbIolSlave.Port      := nTerm2EPI2328; //Assigns Port
  fbIolSlave.Name      := 'EPI2328'; //Gives device a name
END_IF
    
```

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.3.0

6.2.1 Methoden

6.2.1.1 READ



Liest einen Parameter des IO-Link-Geräts aus.

Eingänge

Name	Typ	Beschreibung
nIndex	WORD	Index des IO-Link-Parameters
nSubindex	BYTE	Subindex des IO-Link-Parameters 0 für die gesamte Struktur
pDSTBuf	PVOID	Adresse des Puffers, der die gelesenen Daten aufnehmen soll. Der Programmierer ist selbst dafür verantwortlich, den Puffer auf eine Größe zu dimensionieren, die LEN-Bytes aufnehmen kann. Der Puffer kann eine einzelne Variable, ein Array oder eine Struktur sein, deren Adresse mit dem ADR-Operator ermittelt werden kann.
cbBufLen	UDINT	Anzahl der zu lesenden Daten in Bytes.

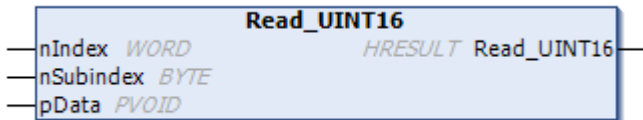
Ausgänge

Name	Typ	Beschreibung
READ	HRESULT	Rückgabewert

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

6.2.1.2 READ_UINT16



Liest einen UINT16 Parameter des IO-Link-Geräts aus und schaltet die Bytes in High Endian um.

Eingänge

Name	Typ	Beschreibung
nIndex	WORD	Index des IO-Link-Parameters 0 für die gesamte Struktur
nSubindex	BYTE	Subindex des IO-Link-Parameters 0 für die gesamte Struktur
pData	PVOID	Adresse des Puffers, der die gelesenen Daten aufnehmen soll. Der Programmierer ist selbst für die Dimensionierung des Puffers auf eine Größe verantwortlich, die LEN-Bytes aufnehmen kann. Der Puffer kann eine einzelne Variable, ein Array oder eine Struktur sein, deren Adresse mit dem ADR-Operator ermittelt werden kann.

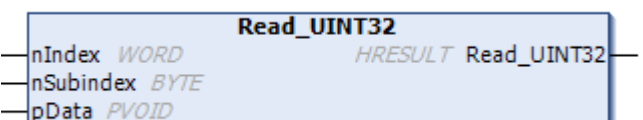
Ausgänge

Name	Typ	Beschreibung
Read_Uint16	HRESULT	Rückgabewert

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

6.2.1.3 READ_UINT32



Liest einen UINT32 Parameter des IO-Link-Geräts aus und schaltet die Bytes in High Endian um.

Eingänge

Name	Typ	Beschreibung
nIndex	WORD	Index des IO-Link-Parameters 0 für die gesamte Struktur
nSubindex	BYTE	Subindex des IO-Link-Parameters 0 für die gesamte Struktur
pData	PVOID	Adresse des Puffers, der die gelesenen Daten aufnehmen soll. Der Programmierer ist selbst für die Dimensionierung des Puffers auf eine Größe verantwortlich, die LEN-Bytes aufnehmen kann. Der Puffer kann eine einzelne Variable, ein Array oder eine Struktur sein, deren Adresse mit dem ADR-Operator ermittelt werden kann.

Ausgänge

Name	Typ	Beschreibung
Read_Uint32	HRESULT	Rückgabewert

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

6.2.1.4 READ_UINT64



Liest einen UINT64 Parameter des IO-Link-Geräts aus und schaltet die Bytes in High Endian um.

Eingänge

Name	Typ	Beschreibung
nIndex	WORD	Index des IO-Link-Parameters
nSubindex	BYTE	Subindex des IO-Link-Parameters 0 für die gesamte Struktur
nData	Reference to ULINT	Referenz zu dem Datentypen ULINT

Ausgänge

Name	Typ	Beschreibung
Read_UINT64	HRESULT	Rückgabewert

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

6.2.1.5 READ_INT16



Liest einen INT16 Parameter des IO-Link-Geräts aus und schaltet die Bytes in High Endian um.

🔴 Eingänge

Name	Typ	Beschreibung
nIndex	WORD	Index des IO-Link-Parameters
nSubindex	BYTE	Subindex des IO-Link-Parameters 0 für die gesamte Struktur
nData	Reference to INT	Referenz zu dem Datentypen INT

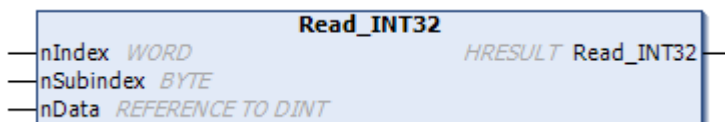
🔴 Ausgänge

Name	Typ	Beschreibung
Read_INT16	HRESULT	Rückgabewert

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

6.2.1.6 READ_INT32



Liest einen UINT32 Parameter des IO-Link-Geräts aus und schaltet die Bytes in High Endian um.

🔴 Eingänge

Name	Typ	Beschreibung
nIndex	WORD	Index des IO-Link-Parameters
nSubindex	BYTE	Subindex des IO-Link-Parameters 0 für die gesamte Struktur
nData	Reference to DINT	Referenz zu dem Datentypen DINT

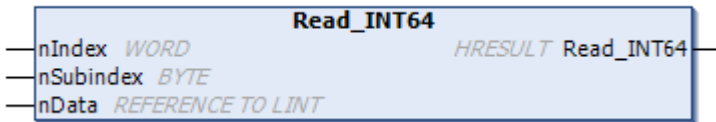
🔴 Ausgänge

Name	Typ	Beschreibung
Read_INT32	HRESULT	Rückgabewert

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

6.2.1.7 READ_INT64



Liest einen INT64 Parameter des IO-Link-Geräts aus und schaltet die Bytes in High Endian um.

Eingänge

Name	Typ	Beschreibung
nIndex	WORD	Index des IO-Link-Parameters
nSubindex	BYTE	Subindex des IO-Link-Parameters 0 für die gesamte Struktur
nData	Reference to LINT	Referenz zu dem Datentypen LINT

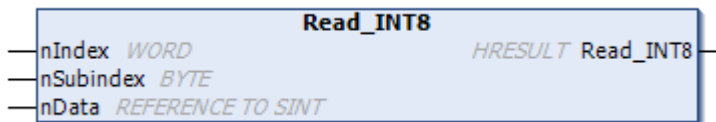
Ausgänge

Name	Typ	Beschreibung
Read_INT64	HRESULT	Rückgabewert

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

6.2.1.8 READ_INT8



Liest einen UINT8 Parameter des IO-Link-Geräts aus.

Eingänge

Name	Typ	Beschreibung
nIndex	WORD	Index des IO-Link-Parameters
nSubindex	BYTE	Subindex des IO-Link-Parameters 0 für die gesamte Struktur
nData	Reference to SINT	Referenz zu dem Datentypen SINT

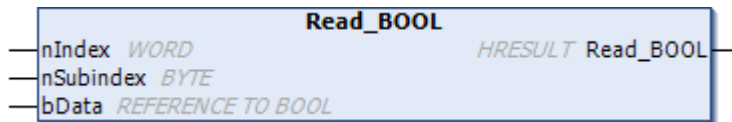
Ausgänge

Name	Typ	Beschreibung
Read_INT8	HRESULT	Rückgabewert

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

6.2.1.9 READ_BOOL



Liest einen BOOL Parameter des IO-Link-Geräts aus.

🔴 Eingänge

Name	Typ	Beschreibung
nIndex	WORD	Index des IO-Link-Parameters
nSubindex	BYTE	Subindex des IO-Link-Parameters 0 für die gesamte Struktur
bData	Reference to BOOL	Referenz zum Datentypen BOOL

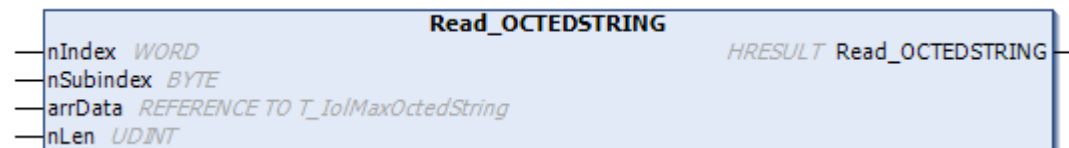
🔴 Ausgänge

Name	Typ	Beschreibung
Read_BOOL	HRESULT	Rückgabewert

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

6.2.1.10 READ_OctedString



Liest einen OctetString Parameter des IO-Link-Geräts aus.

🔴 Eingänge

Name	Typ	Beschreibung
nIndex	WORD	Index des IO-Link-Parameters
nSubindex	BYTE	Subindex des IO-Link-Parameters 0 für die gesamte Struktur
arrData	Reference to T_IolMaxOctedString [▶ 17]	Referenz zu dem Type T_IolMaxOctedString
nLen	UDINT	Größe der Variablen

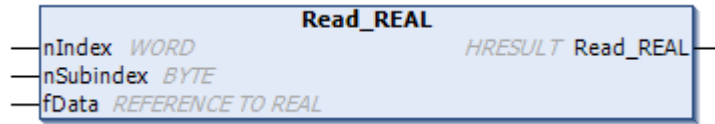
🔴 Ausgänge

Name	Typ	Beschreibung
Read_OctedString	HRESULT	Rückgabewert

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

6.2.1.11 READ_Real



Liest einen REAL Parameter des IO-Link-Geräts aus und schaltet die Bytes in High Endian um.

Eingänge

Name	Typ	Beschreibung
nIndex	WORD	Index des IO-Link-Parameters
nSubindex	BYTE	Subindex des IO-Link-Parameters 0 für die gesamte Struktur
fData	Reference to REAL	Referenz zu dem Datentypen REAL

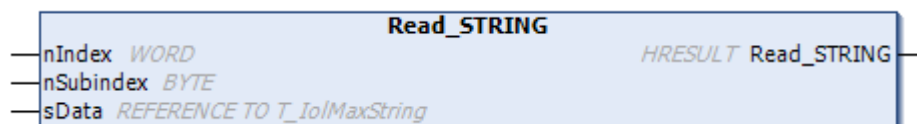
Ausgänge

Name	Typ	Beschreibung
Read_REAL	HRESULT	Rückgabewert

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

6.2.1.12 READ_String



Liest einen STRING Parameter des IO-Link-Geräts aus und schaltet die Bytes in High Endian um.

Eingänge

Name	Typ	Beschreibung
nIndex	WORD	Index des IO-Link-Parameters
nSubindex	BYTE	Subindex des IO-Link-Parameters 0 für die gesamte Struktur
sData	Reference to T_IolMaxString [▶ 17]	Referenz zum Type T_IolMaxString

Ausgänge

Name	Typ	Beschreibung
Read_STRING	HRESULT	Rückgabewert

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

6.2.1.13 READ_String_FixedLen

Read_STRING_FixedLen	
nIndex <i>WORD</i>	<i>HRESULT</i> Read_STRING_FixedLen
nSubindex <i>BYTE</i>	
sData <i>REFERENCE TO T_IolMaxString</i>	
nLen <i>UDINT</i>	

Liest einen STRING Parameter mit einer begrenzten Länge des IO-Link-Geräts aus und schaltet die Bytes in High Endian um.

🔴 Eingänge

Name	Typ	Beschreibung
nIndex	WORD	Index des IO-Link-Parameters
nSubindex	BYTE	Subindex des IO-Link-Parameters 0 für die gesamte Struktur
sData	Reference to T_IolMaxString [► 17]	Referenz zum Type T_IolMaxString
nLen	UDINT	Länge des Strings

🔴 Ausgänge

Name	Typ	Beschreibung
Read_String_FixedLen	HRESULT	Rückgabewert

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

6.2.1.14 READ_Uint8

Read_UINT8	
nIndex <i>WORD</i>	<i>HRESULT</i> Read_UINT8
nSubindex <i>BYTE</i>	
nData <i>REFERENCE TO USINT</i>	

Liest einen UINT8 Parameter des IO-Link-Geräts aus und schaltet die Bytes in High Endian um.

🔴 Eingänge

Name	Typ	Beschreibung
nIndex	WORD	Index des IO-Link-Parameters
nSubindex	BYTE	Subindex des IO-Link-Parameters 0 für die gesamte Struktur
nData	Reference to USINT	Referenz zum Datentypen USINT;

Ausgänge

Name	Typ	Beschreibung
Read_UINT8	HRESULT	Rückgabewert

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

6.2.1.15 Write



Schreibt einen Parameter eines IO-Link Geräts.

Eingänge

Name	Typ	Beschreibung
nIndex	WORD	Index des IO-Link-Parameters
nSubindex	BYTE	Subindex des IO-Link-Parameters 0 für die gesamte Struktur
pDSTBuf	PVOID	Adresse des Puffers, der die geschriebenen Daten aufnehmen soll. Der Programmierer ist selbst dafür verantwortlich, den Puffer auf eine Größe zu dimensionieren, die LEN-Bytes aufnehmen kann. Der Puffer kann eine einzelne Variable, ein Array oder eine Struktur sein, deren Adresse mit dem ADR-Operator ermittelt werden kann.
cbBufLen	UDINT	Anzahl der zu schreibenden Daten in Bytes.

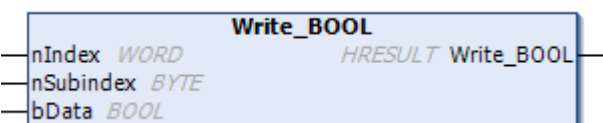
Ausgänge

Name	Typ	Beschreibung
Write	HRESULT	Rückgabewert

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

6.2.1.16 Write_BOOL



Schreibt einen BOOL Parameter des IO-Link-Geräts.

🔍 Eingänge

Name	Typ	Beschreibung
nIndex	WORD	Index des IO-Link-Parameters
nSubindex	BYTE	Subindex des IO-Link-Parameters 0 für die gesamte Struktur
bData	BOOL	Wert der geschrieben wird

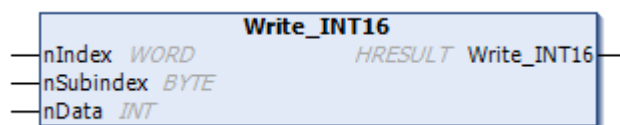
➡ Ausgänge

Name	Typ	Beschreibung
hresult	HRESULT	Rückgabewert

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

6.2.1.17 Write_INT16



Schreibt einen INT16 Parameter des IO-Link-Geräts und schaltet die Bytes in High Endian um.

🔍 Eingänge

Name	Typ	Beschreibung
nIndex	WORD	Index des IO-Link-Parameters
nSubindex	BYTE	Subindex des IO-Link-Parameters 0 für die gesamte Struktur
nData	INT	INT Wert der geschrieben wird

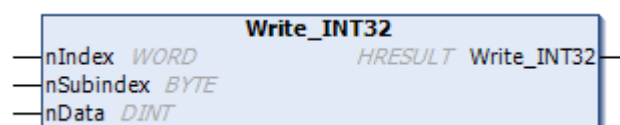
➡ Ausgänge

Name	Typ	Beschreibung
Write_INT16	HRESULT	Rückgabewert

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

6.2.1.18 Write_INT32



Schreibt einen INT32 Parameter des IO-Link-Geräts und schaltet die Bytes in High Endian um.

Eingänge

Name	Typ	Beschreibung
nIndex	WORD	Index des IO-Link-Parameters
nSubindex	BYTE	Subindex des IO-Link-Parameters 0 für die gesamte Struktur
nData	DINT	DINT Wert der geschrieben wird

Ausgänge

Name	Typ	Beschreibung
Write_INT32	HRESULT	Rückgabewert

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

6.2.1.19 Write_INT64



Schreibt einen INT64 Parameter des IO-Link-Geräts und schaltet die Bytes in High Endian um.

Eingänge

Name	Typ	Beschreibung
nIndex	WORD	Index des IO-Link-Parameters
nSubindex	BYTE	Subindex des IO-Link-Parameters 0 für die gesamte Struktur
nData	LINT	LINT Wert der geschrieben wird

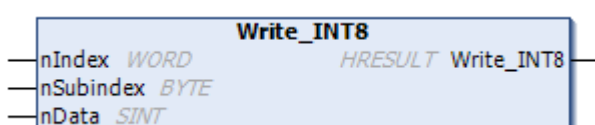
Ausgänge

Name	Typ	Beschreibung
Write_INT64	HRESULT	Rückgabewert

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

6.2.1.20 Write_INT8



Schreibt einen INT8 Parameter des IO-Link-Geräts.

🔌 Eingänge

Name	Typ	Beschreibung
nIndex	WORD	Index des IO-Link-Parameters
nSubindex	BYTE	Subindex des IO-Link-Parameters 0 für die gesamte Struktur
nData	SINT	SINT Wert der geschrieben wird

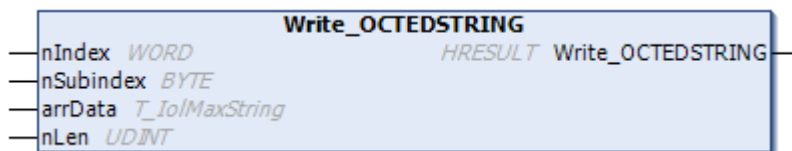
➡️ Ausgänge

Name	Typ	Beschreibung
Write_INT8	HRESULT	Rückgabewert

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

6.2.1.21 Write_OCTEDSTRING



Schreibt einen OCTEDSTRING Parameter des IO-Link-Geräts.

🔌 Eingänge

Name	Typ	Beschreibung
nIndex	WORD	Index des IO-Link-Parameters
nSubindex	BYTE	Subindex des IO-Link-Parameters 0 für die gesamte Struktur
arrData	T_IoMaxString [▶ 17]	T_IoMaxString Wert der geschrieben wird
nLen	UDINT	Länge des Strings

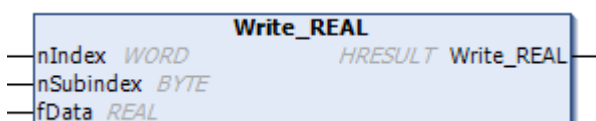
➡️ Ausgänge

Name	Typ	Beschreibung
Write_OCTEDSTRING	HRESULT	Rückgabewert

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

6.2.1.22 Write_REAL



Schreibt einen REAL Parameter des IO-Link-Geräts und schaltet die Bytes in High Endian um.

 **Eingänge**

Name	Typ	Beschreibung
nIndex	WORD	Index des IO-Link-Parameters
nSubindex	BYTE	Subindex des IO-Link-Parameters 0 für die gesamte Struktur
fData	REAL	REAL Wert der geschrieben wird

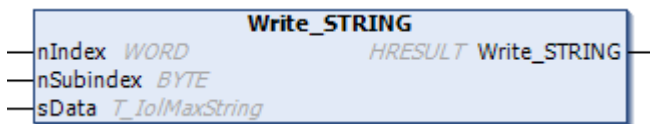
 **Ausgänge**

Name	Typ	Beschreibung
Write_Real	HRESULT	Rückgabewert

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

6.2.1.23 Write_STRING



Schreibt einen String Parameter des IO-Link-Geräts.

 **Eingänge**

Name	Typ	Beschreibung
nIndex	WORD	Index des IO-Link-Parameters
nSubindex	BYTE	Subindex des IO-Link-Parameters 0 für die gesamte Struktur
sData	T_IoMaxString [▶ 17]	T_IoMaxString Wert, der geschrieben wird.

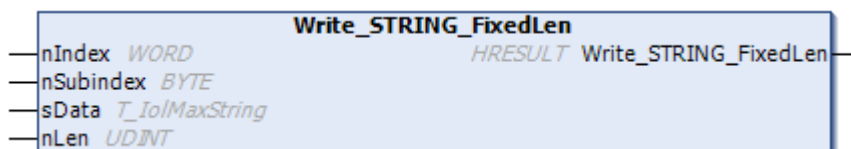
 **Ausgänge**

Name	Typ	Beschreibung
Write_STRING	HRESULT	Rückgabewert

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

6.2.1.24 Write_STRING_FixedLen



Schreibt einen String Parameter mit einer begrenzten Länge des IO-Link-Geräts.

🔌 Eingänge

Name	Typ	Beschreibung
nIndex	WORD	Index des IO-Link-Parameters
nSubindex	BYTE	Subindex des IO-Link-Parameters 0 für die gesamte Struktur
sData	T_IoLinkString_ [▶_17]	T_IoLinkString Wert, der geschrieben wird
nLen	UDINT	Anzahl der Zeichen

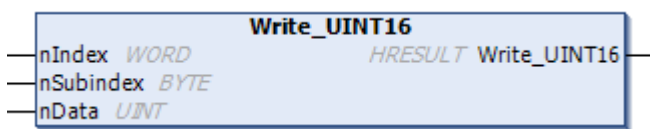
🔌 Ausgänge

Name	Typ	Beschreibung
Write_STRING_FixedLen	HRESULT	Rückgabewert

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

6.2.1.25 Write_UINT16



Schreibt einen UINT16 Parameter des IO-Link-Geräts und schaltet die Bytes in High Endian um.

🔌 Eingänge

Name	Typ	Beschreibung
nIndex	WORD	Index des IO-Link-Parameters
nSubindex	BYTE	Subindex des IO-Link-Parameters 0 für die gesamte Struktur
nData	INT	INT Wert der geschrieben wird

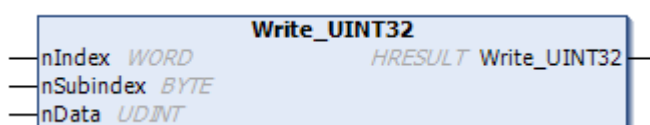
🔌 Ausgänge

Name	Typ	Beschreibung
Write_UINT16	HRESULT	Rückgabewert

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

6.2.1.26 Write_UINT32



Schreibt einen UINT32 Parameter des IO-Link-Geräts und schaltet die Bytes in High Endian um.

Eingänge

Name	Typ	Beschreibung
nIndex	WORD	Index des IO-Link-Parameters
nSubindex	BYTE	Subindex des IO-Link-Parameters 0 für die gesamte Struktur
nData	UDINT	UDINT Wert, der geschrieben wird

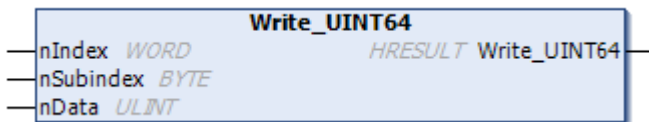
Ausgänge

Name	Typ	Beschreibung
Write_UINT32	HRESULT	Rückgabewert

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

6.2.1.27 Write_UINT64



Schreibt einen UINT64 Parameter des IO-Link-Geräts und schaltet die Bytes in High Endian um.

Eingänge

Name	Typ	Beschreibung
nIndex	WORD	Index des IO-Link-Parameters
nSubindex	BYTE	Subindex des IO-Link-Parameters 0 für die gesamte Struktur
nData	ULINT	ULINT Wert, der geschrieben wird

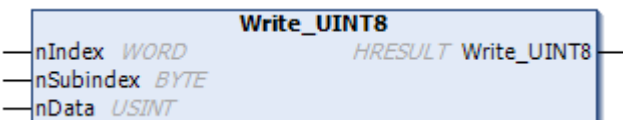
Ausgänge

Name	Typ	Beschreibung
Write_UINT64	HRESULT	Rückgabewert

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

6.2.1.28 Write_UINT8



Schreibt einen UINT8 Parameter des IO-Link-Geräts.

Eingänge

Name	Typ	Beschreibung
nIndex	WORD	Index des IO-Link-Parameters
nSubindex	BYTE	Subindex des IO-Link-Parameters 0 für die gesamte Struktur
nData	USINT	USINT Wert, der geschrieben wird

Ausgänge

Name	Typ	Beschreibung
Write_UINT8	HRESULT	Rückgabewert

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

6.2.2 Beispiel

So kann die Verwendung einzelner Funktionen aussehen. In diesem Beispiel wird der ApplicationSpecificTag per Write geschrieben und anschließend per Read Methode ausgelesen und überprüft.

Der folgende FB ist Teil des IoLink Beispielprojekts. Weitere Informationen finden Sie im Kapitel [Beispiele](#) [► 69].

```

FUNCTION_BLOCK FB_Sample_03_IoSlave
VAR_INPUT
    bExecute          : BOOL := FALSE;
    sApplicationTag   : STRING(32) := '';
END_VAR
VAR_OUTPUT
    bBusy             : BOOL := FALSE;
    bDone             : BOOL := FALSE;
    bError            : BOOL := FALSE;
    nAdsError         : E_AdsErr := E_AdsErr.NOERR;
    nIoLError         : E_IoLError := E_IoLError.NoError;
    sErrorTxt         : STRING(255) := '';
END_VAR
VAR
    bExecuteOld       : BOOL;
    nState            : INT := 0;
    fbIoSlave         : FB_IoSlave;
    bInitDone         : BOOL := FALSE;
    hr                : HRESULT;
    sApplicationTagRead : STRING(32) := '';
END_VAR

////
// Init
//
IF NOT bInitDone THEN
    bInitDone := TRUE;
    fbIoSlave.MasterAoeNetId := F_CreateAmsNetId(GVL_IO.sTerm2AmsNetId);
    fbIoSlave.Port := GVL_IO.nTerm2EPI1008;
    fbIoSlave.Name := 'EPI1008';
END_IF

////
// Execute/Busy State Machine
//
IF bExecute AND NOT bExecuteOld AND NOT bBusy THEN

```

```
    bBusy := TRUE;
    nState := 10; // WRITE
ELSIF NOT bExecute THEN
    bBusy := FALSE;
    bError := FALSE;
    bDone := FALSE;
    nAdsError := E_AdsErr.NOERR;
    nIolError := E_IolError.NoError;
    sErrorTxt := '';
END_IF
bExecuteOld := bExecute;

////
// FB State Machine
//
CASE nState OF

// IDLE
0:;

// WRITE
10:
    hr := fbIolSlave.Write(16#0018, 0, ADR(sApplicationTag), 32);

    IF NOT F_IolIsBusy(hr) THEN
        IF FAILED(hr) THEN
            bBusy := FALSE;
            bError := TRUE;
            sErrorTxt := fbIolSlave.ErrorTxt;
            nAdsError := F_IolGetAdsError(hr);
            nIolError := F_IolGetIolError(hr);
            nState := 0; // IDLE;
        ELSE
            nState := 20; // READ;
        END_IF
    END_IF

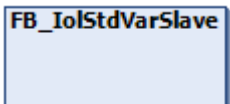
// READ
20:
    hr := fbIolSlave.Read(16#0018, 16#00, ADR(sApplicationTagRead), 32);

    IF NOT F_IolIsBusy(hr) THEN
        IF FAILED(hr) THEN
            bBusy := FALSE;
            bError := TRUE;
            sErrorTxt := fbIolSlave.ErrorTxt;
            nAdsError := F_IolGetAdsError(hr);
            nIolError := F_IolGetIolError(hr);
            nState := 0; // IDLE;
        ELSE
            nState := 30; // CHECK;
        END_IF
    END_IF

// CHECK
30:
    IF sApplicationTag <> sApplicationTagRead THEN
        bBusy := FALSE;
        bError := TRUE;
        sErrorTxt := 'sApplicationTag and sApplicationTagRead are not equal!';
    ELSE
        bBusy := FALSE;
        bDone := TRUE;
        nState := 0;
    END_IF
END_CASE

////
// call FBs
//
fbIolSlave();
```

6.3 FB_IolStdVarSlave



Dieser Funktionsbaustein erbt von dem FB [IOISlave](#) [► 35]. Er ist für das Schreiben und Auslesen von standardisierten IO-Link Parametern gedacht.

Beispiel

```

VAR
  bInitDone           : BOOL;
  sTerm3AmsNetId AT%I* : AMSNETID;
  nTerm2EPI2328       : E_IoLPort := E_IolPort.Port2;
  fbIOLStdVarSlave    : FB_IolStdVarSlave;
END_VAR

IF NOT bInitDone THEN
  bInitDone           := TRUE;           //Assigns it only once
  fbIOLStdVarSlave.MasterAoeNetId := F_CreateAmsNetId(sTerm3AmsNetId); //Creates NETID String
  fbIOLStdVarSlave.Port      := nTerm2EPI2328; //Assigns Port
  fbIOLStdVarSlave.Name      := 'EPI2328'; //Gives device a name
END_IF

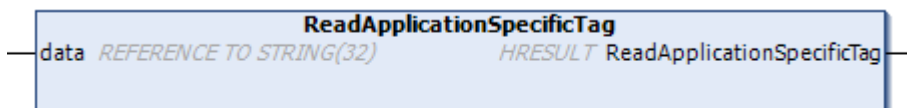
```

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

6.3.1 Methoden

6.3.1.1 ReadApplicationSpecificTag



Liest den ApplicationSpecificTag aus.

👉 Eingänge

Name	Typ	Beschreibung
data	Reference to STRING(32)	Variable des Puffers, der die gelesenen Daten aufnehmen soll.

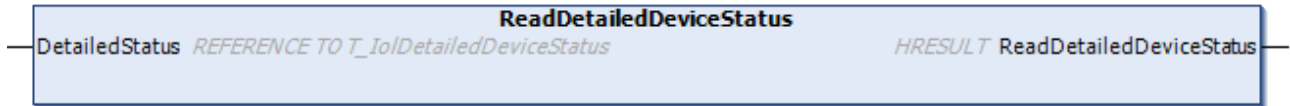
👈 Ausgänge

Name	Typ	Beschreibung
hresult	HRESULT	Rückgabewert

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

6.3.1.2 ReadDetailedDeviceStatus



Liest den Detailed Device Status eines IO-Link Geräts aus.

Eingänge

Name	Typ	Beschreibung
DetailedStatus	Reference to T_IoDetailedDeviceStatus [▶_17]	Variable des Puffers, der die gelesenen Daten aufnehmen soll.

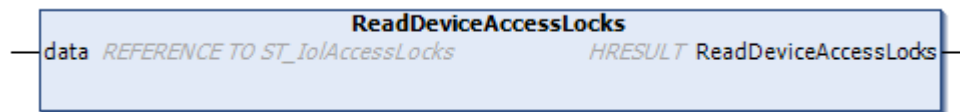
Ausgänge

Name	Typ	Beschreibung
hresult	HRESULT	Rückgabewert

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

6.3.1.3 ReadDeviceAccessLocks



Liest die Gerätezugriffssperre eines IO-Link Geräts aus.

Eingänge

Name	Typ	Beschreibung
data	Reference to ST_IoAccessLocks [▶_16]	Gibt die Gerätezugriffssperre an

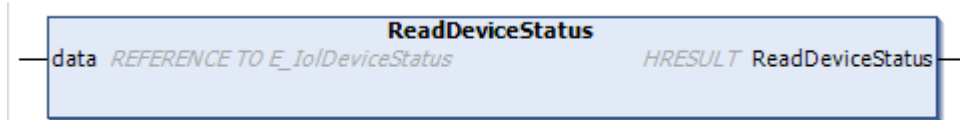
Ausgänge

Name	Typ	Beschreibung
hresult	HRESULT	Rückgabewert

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

6.3.1.4 ReadDeviceStatus



Liest den Gerätstatus des IO-Link-Geräts aus.

🔌 Eingänge

Name	Typ	Beschreibung
data	Reference to E_IoDeviceStatus [▶ 14]	Variable des Puffers, der die gelesenen Daten aufnehmen soll.

🔌 Ausgänge

Name	Typ	Beschreibung
hresult	HRESULT	Rückgabewert

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

6.3.1.5 ReadDirectParameter1



Liest die DirectParameter1 Seite des IO-Link-Geräts aus.

🔌 Eingänge

Name	Typ	Beschreibung
data	Reference to ST_IoDirectParameter1 [▶ 16]	Variable des Puffers, der die gelesenen Daten aufnehmen soll.

🔌 Ausgänge

Name	Typ	Beschreibung
hresult	HRESULT	Rückgabewert

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

6.3.1.6 ReadDirectParameter2



Liest die DirectParameter2 aus einem IO-Link Gerät aus.

 **Eingänge**

Name	Typ	Beschreibung
data	Reference to ST_IoDirectParameter2 [▶_17]	Variable des Puffers, der die gelesenen Daten aufnehmen soll.

 **Ausgänge**

Name	Typ	Beschreibung
hresult	HRESULT	Rückgabewert

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

6.3.1.7 ReadFirmwareVersion



Liest die Firmwareversion eine IO-Link Gerätes aus.

 **Eingänge**

Name	Typ	Beschreibung
data	Reference to String(64)	Variable des Puffers, der die gelesenen Daten aufnehmen soll.

 **Ausgänge**

Name	Typ	Beschreibung
hresult	HRESULT	Rückgabewert

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

6.3.1.8 ReadHardwareVersion



Liest die Hardwareversion des IO-Link-Geräts aus.

 **Eingänge**

Name	Typ	Beschreibung
data	Reference to STRING(64)	Variable des Puffers, der die gelesenen Daten aufnehmen soll.

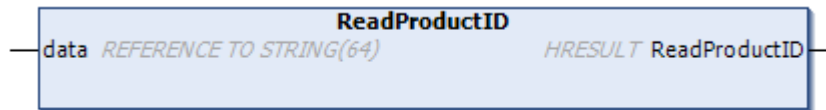
Ausgänge

Name	Typ	Beschreibung
hresult	HRESULT	Rückgabewert

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_loLink ab v3.3.4.0

6.3.1.9 ReadProductID



Liest die Produkt ID des IO-Link-Geräts aus.

Eingänge

Name	Typ	Beschreibung
data	Reference to STRING(64)	Variable des Puffers, der die gelesenen Daten aufnehmen soll.

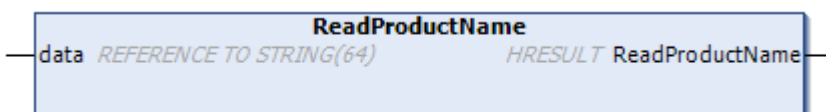
Ausgänge

Name	Typ	Beschreibung
hresult	HRESULT	Rückgabewert

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_loLink ab v3.3.4.0

6.3.1.10 ReadProductName



Liest den Produktnamen des IO-Link-Geräts aus.

Eingänge

Name	Typ	Beschreibung
data	Reference to STRING(64)	Variable des Puffers, der die gelesenen Daten aufnehmen soll.

Ausgänge

Name	Typ	Beschreibung
hresult	HRESULT	Rückgabewert

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

6.3.1.11 ReadProductText



Liest den Produkttext eines IO-Link-Geräts aus.

 **Eingänge**

Name	Typ	Beschreibung
data	Reference to STRING(64)	Variable des Puffers, der die gelesenen Daten aufnehmen soll.

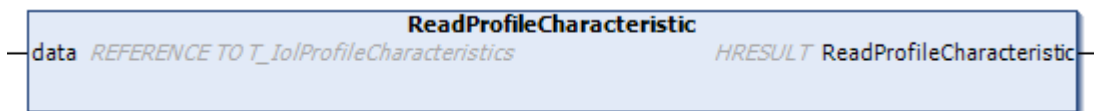
 **Ausgänge**

Name	Typ	Beschreibung
hresult	HRESULT	Rückgabewert

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

6.3.1.12 ReadProfileCharacteristic



Liest die Profilcharakteristik eines IO-Link-Geräts aus.

 **Eingänge**

Name	Typ	Beschreibung
data	Reference to <u>T_IolProfileCharacteristics</u> [▶_171]	Variable des Puffers, der die gelesenen Daten aufnehmen soll.

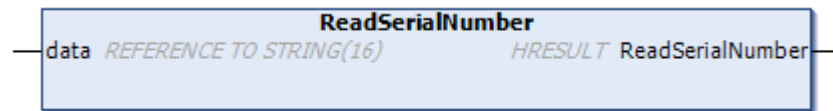
 **Ausgänge**

Name	Typ	Beschreibung
hresult	HRESULT	Rückgabewert

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

6.3.1.13 ReadSerialNumber



Methode liest die Seriennummer des IO-Link-Geräts aus.

Eingänge

Name	Typ	Beschreibung
data	Reference to STRING(16)	Variable des Puffers, der die gelesenen Daten aufnehmen soll.

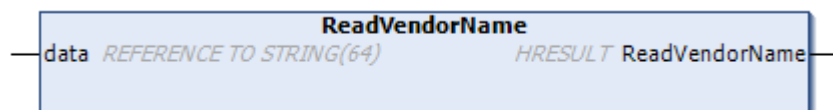
Ausgänge

Name	Typ	Beschreibung
hresult	HRESULT	Rückgabewert

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

6.3.1.14 ReadVendorName



Liest den Herstellernamen des IO-Link-Geräts aus.

Eingänge

Name	Typ	Beschreibung
data	Reference to STRING(64)	Variable des Puffers, der die gelesenen Daten aufnehmen soll.

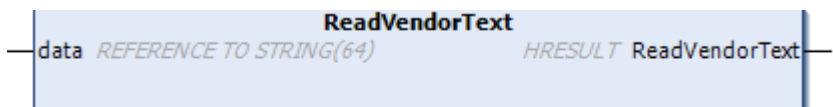
Ausgänge

Name	Typ	Beschreibung
hresult	HRESULT	Rückgabewert

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

6.3.1.15 ReadVendorText



Liest den Hersteller Text des IO-Link-Geräts aus.

Eingänge

Name	Typ	Beschreibung
data	Reference to STRING(64)	Variable des Puffers, der die gelesenen Daten aufnehmen soll.

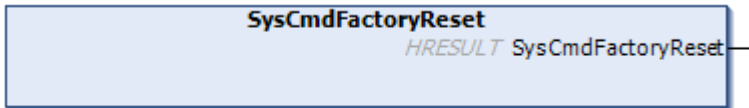
Ausgänge

Name	Typ	Beschreibung
hresult	HRESULT	Rückgabewert

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

6.3.1.16 SysCmdFactoryReset



Setzt das IO-Link-Gerät auf Werkseinstellungen zurück.

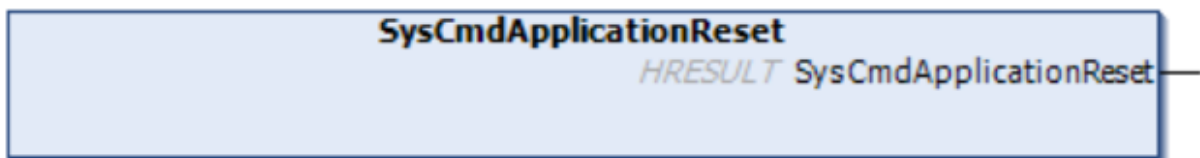
Ausgänge

Name	Typ	Beschreibung
hresult	HRESULT	Rückgabewert

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

6.3.1.17 SysCmdApplicationReset



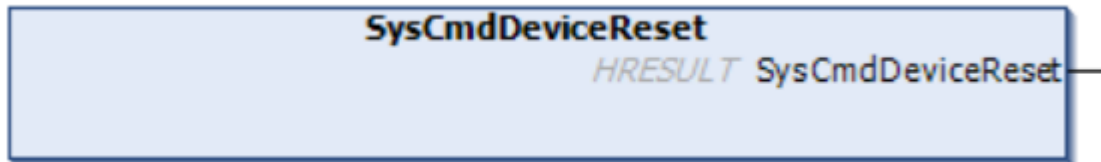
Setzt das IO-Link-Gerät auf die Applikationseinstellungen zurück.

Ausgänge

Name	Typ	Beschreibung
hresult	HRESULT	Rückgabewert

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

6.3.1.18 SysCmdDeviceReset

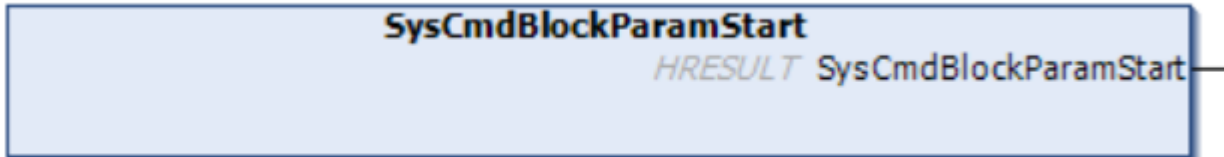
Setzt das IO-Link-Gerät auf Geräteeinstellungen zurück.

➡ Ausgänge

Name	Typ	Beschreibung
hresult	HRESULT	Rückgabewert

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

6.3.1.19 SysCmdBlockParamStart

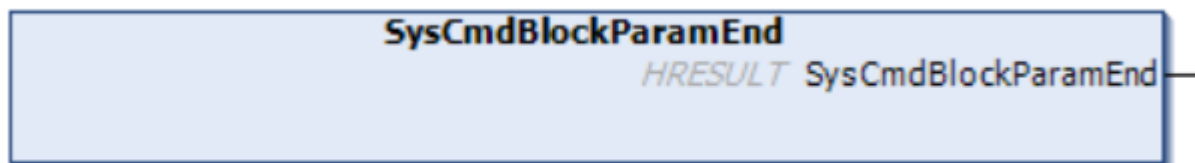
Startet die Block Parametrisierung.

➡ Ausgänge

Name	Typ	Beschreibung
hresult	HRESULT	Rückgabewert

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

6.3.1.20 SysCmdBlockParamEnd

Beendet die Block Parametrisierung.

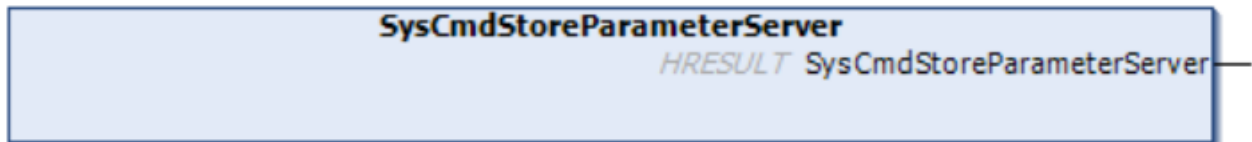
Ausgänge

Name	Typ	Beschreibung
hresult	HRESULT	Rückgabewert

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

6.3.1.21 SysCmdStoreParameterServer



Setzt den Datastorage über die PLC ab.

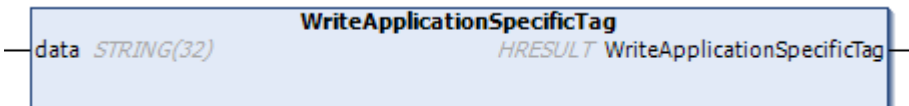
Ausgänge

Name	Typ	Beschreibung
hresult	HRESULT	Rückgabewert

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

6.3.1.22 WriteApplicationSpecificTag



Schreibt den IO-Link Parameter ApplicationSpecifigTag.

Eingänge

Name	Typ	Beschreibung
data	STRING(32)	Variable des Puffers, der die gelesenen Daten aufnehmen soll.

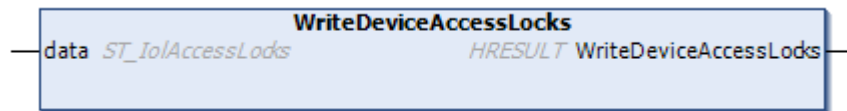
Ausgänge

Name	Typ	Beschreibung
hresult	HRESULT	Rückgabewert

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

6.3.1.23 WriteDeviceAccessLocks



Schreibt die DeviceAccessLocks eines IO-Link-Geräts.

Eingänge

Name	Typ	Beschreibung
data	Reference to ST_IolAccessLocks [▶_16]	Variable des Puffers, der die geschriebenen Daten aufnehmen soll.

Ausgänge

Name	Typ	Beschreibung
hresult	HRESULT	Rückgabewert

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

6.3.1.24 WriteDirectParameter2



Schreibt den Parameter DirectParameter2 des IO-Link-Geräts.

Eingänge

Name	Typ	Beschreibung
data	Reference to ST_IolDirectParameter2 [▶_17]	Variable des Puffers, der die gelesenen Daten aufnehmen soll.

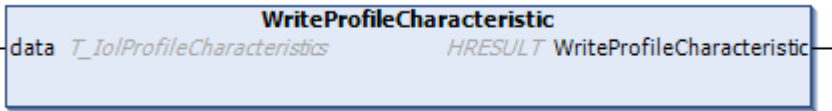
Ausgänge

Name	Typ	Beschreibung
hresult	HRESULT	Rückgabewert

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

6.3.1.25 WriteProfileCharacteristic



Schreibt den Parameter ProfileCharacteristic des IO-Link-Geräts.

Eingänge

Name	Typ	Beschreibung
data	Reference to <u>T_IolProfileCharacteristics</u> [▶ 17]	Variable des Puffers, der die geschriebenen Daten aufnehmen soll.

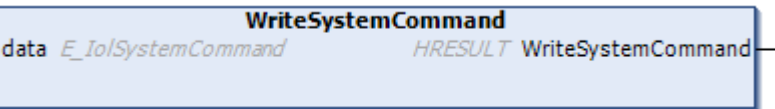
Ausgänge

Name	Typ	Beschreibung
hresult	HRESULT	Rückgabewert

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

6.3.1.26 WriteSystemCommand



Schreibt einen beliebigen System Command für das IO-Link-Gerät.

Eingänge

Name	Typ	Beschreibung
data	Reference to <u>E_IolSystemCommand</u> [▶ 15]	Variable des Puffers, der die geschriebenen Daten aufnehmen soll.

Ausgänge

Name	Typ	Beschreibung
hresult	HRESULT	Rückgabewert

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

6.3.2 Beispiel

In diesem Beispiel wird der ApplicationSpecificTag mit den Methoden WriteApplicationSpecificTag geschrieben und anschließend mit der Methode ReadApplicationSpecificTag ausgelesen und überprüft.

```

FUNCTION_BLOCK FB_Sample_04_IolStdVarSlave
VAR_INPUT
  bExecute      :BOOL := FALSE;
  sApplicationTag :STRING(32) := '';
END_VAR
VAR_OUTPUT
  bBusy        :BOOL := FALSE;
  bDone        :BOOL := FALSE;
  bError       :BOOL := FALSE;
  nAdsError    :E_AdsErr := E_AdsErr.NOERR;
  nIolError    :E_IolError := E_IolError.NoError;
  sErrorTxt    :STRING(255) := '';
END_VAR
VAR
  bExecuteOld   :BOOL;
  nState        :INT := 0;
  fbIolSlave    :FB_IolStdVarSlave;
  bInitDone     :BOOL := FALSE;
  hr            :HRESULT;
  sApplicationTagRead : STRING(32) := '';
END_VAR

////
// Init
//
IF NOT bInitDone THEN
  bInitDone := TRUE;
  fbIolSlave.MasterAoeNetId := F_CreateAmsNetId(GVL_IO.sTerm2AmsNetId); //Creates NETID String
  fbIolSlave.Port := GVL_IO.nTerm2EPI1008;
  fbIolSlave.Name := 'EPI1008';
END_IF

////
// Execute/Busy State Machine
//
IF bExecute AND NOT bExecuteOld AND NOT bBusy THEN
  bBusy := TRUE;
  nState := 10; // WRITE
ELSIF NOT bExecute THEN
  bBusy := FALSE;
  bError := FALSE;
  bDone := FALSE;
  nAdsError := E_AdsErr.NOERR;
  nIolError := E_IolError.NoError;
  sErrorTxt := '';
END_IF
bExecuteOld := bExecute;

////
// FB State Machine
//
CASE nState OF

// IDLE
0;;

// WRITE
10:
  hr := fbIolSlave.WriteApplicationSpecificTag(sApplicationTag);

  IF NOT F_IolIsBusy(hr) THEN
    IF FAILED(hr) THEN
      bBusy := FALSE;
      bError := TRUE;
      sErrorTxt := fbIolSlave.ErrorTxt; // TODO
      nAdsError := F_IolGetAdsError(hr);
      nIolError := F_IolGetIolError(hr);
      nState := 0; // IDLE;
    ELSE
      nState := 20; // WRITE;
    END_IF
  END_IF

// READ
20:
  hr := fbIolSlave.ReadApplicationSpecificTag(sApplicationTagRead);

  IF NOT F_IolIsBusy(hr) THEN
    IF FAILED(hr) THEN
      bBusy := FALSE;

```

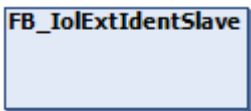


```

bError := TRUE;
sErrorTxt := fbIolSlave.ErrorTxt; // TODO
nAdsError := F_IolGetAdsError(hr);
nIolError := F_IolGetIolError(hr);
nState := 0; // IDLE;
ELSE
nState := 30; // CHECK;
END_IF
END_IF
// CHECK
30:
IF sApplicationTag <> sApplicationTagRead THEN
bBusy := FALSE;
bError := TRUE;
sErrorTxt := 'sApplicationTag and sApplicationTagRead are not equal!';
ELSE
bBusy := FALSE;
bDone := TRUE;
nState := 0;
END_IF
END_CASE
////
// call FBs
//
fbIolSlave();

```

6.4 FB_IolExtIdentSlave



FB_IolExtSlave erbt von dem Funktionsbaustein FB_IolSlave.

Nur IO-Link-Geräte mit dem Profil: [0x8100 \[► 10\]](#) haben diese Parameter.

Beispiel

```

VAR
bInitDone : BOOL;
sTerm3AmsNetId AT%I* : AMSNETID;
nTerm2EPI2328 : E_IolPort := E_IolPort.Port2;
fbIOLExtIdentSlave : FB_IolExtIdentSlave;
END_VAR

IF NOT bInitDone THEN
bInitDone := TRUE; //Assigns it only once
fbIOLExtIdentSlave.MasterAoeNetId := F_CreateAmsNetId(sTerm3AmsNetId) //Creates NETID String
fbIOLExtIdentSlave.Port := nTerm2EPI2328; //Assigns Port
fbIOLExtIdentSlave.Name := 'EPI2328'; //Gives device a name
END_IF

```

Voraussetzungen

Entwicklungsumgebung	Erforderliche SPS-Bibliothek
TwinCAT ab v3.1.4024.25	Tc3_IoLink ab v3.3.4.0

6.4.1 Methoden

6.4.1.1 ReadFunctionTag



Funktion zum Auslesen des FunctionTag.

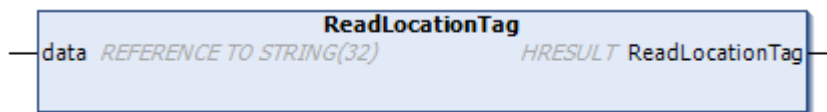
📥 Eingänge

Name	Typ	Beschreibung
data	Reference to STRING(32)	Variable des Puffers, der die gelesenen Daten aufnehmen soll.

📤 Ausgänge

Name	Typ	Beschreibung
hresult	HRESULT	Rückgabewert

6.4.1.2 ReadLocationTag



Funktion zum Auslesen des LocationTag. Der LocationTag ist ein Parameter, der die Ortsangabe des IO-Link Geräts beinhaltet.

📥 Eingänge

Name	Typ	Beschreibung
data	Reference to STRING(32)	Variable des Puffers, der die gelesenen Daten aufnehmen soll.

📤 Ausgänge

Name	Typ	Beschreibung
hresult	HRESULT	Rückgabewert

6.4.1.3 WriteFunctionTag



Schreibt den Parameter FunctionTag.

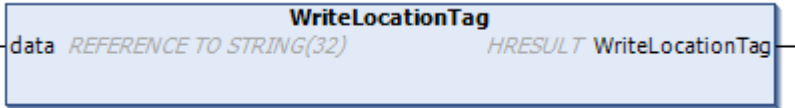
📥 Eingänge

Name	Typ	Beschreibung
data	STRING(32)	Variable des Puffers, der die gelesenen Daten aufnehmen soll.

📤 Ausgänge

Name	Typ	Beschreibung
hresult	HRESULT	Rückgabewert

6.4.1.4 WriteLocationTag



Schreibt den LocationTag eines IO-Link Geräts.

Eingänge

Name	Typ	Beschreibung
data	STRING(32)	Variable des Puffers, der die gelesenen Daten aufnehmen soll.

Ausgänge

Name	Typ	Beschreibung
hresult	HRESULT	Rückgabewert

6.4.2 Beispiel

In diesem Beispiel wird der ApplicationSpecificTag mit den Methoden WriteApplicationSpecificTag geschrieben und anschließend mit der Methode ReadApplicationSpecificTag ausgelesen und überprüft.

```

FUNCTION_BLOCK FB_Sample_04_IolStdVarSlave
VAR_INPUT
    bExecute          :BOOL := FALSE;
    sApplicationTag   :STRING(32) := '';
END_VAR
VAR_OUTPUT
    bBusy            :BOOL := FALSE;
    bDone            :BOOL := FALSE;
    bError           :BOOL := FALSE;
    nAdsError        :E_AdsErr := E_AdsErr.NOERR;
    nIolError        :E_IolError := E_IolError.NoError;
    sErrorTxt        :STRING(255) := '';
END_VAR
VAR
    bExecuteOld      :BOOL;
    nState           :INT := 0;
    fbIolSlave       :FB_IolStdVarSlave;
    bInitDone        :BOOL := FALSE;
    hr               :HRESULT;
    sApplicationTagRead : STRING(32) := '';
END_VAR

////
// Init
//
IF NOT bInitDone THEN
    bInitDone := TRUE;
    fbIolSlave.MasterAoeNetId := F_CreateAmsNetId(GVL_IO.sTerm2AmsNetId); //Creates NETID String
    fbIolSlave.Port := GVL_IO.nTerm2EPI1008;
    fbIolSlave.Name := 'EPI1008';
END_IF

////
// Execute/Busy State Machine
//
IF bExecute AND NOT bExecuteOld AND NOT bBusy THEN
    bBusy := TRUE;
    nState := 10; // WRITE
ELSIF NOT bExecute THEN
    bBusy := FALSE;
    bError := FALSE;
    bDone := FALSE;
    nAdsError := E_AdsErr.NOERR;
    nIolError := E_IolError.NoError;
    sErrorTxt := '';
END_IF
    
```

```
bExecuteOld := bExecute;

////
// FB State Machine
//
CASE nState OF

// IDLE
0:;

// WRITE
10:
  hr := fbIolSlave.WriteApplicationSpecificTag(sApplicationTag);

  IF NOT F_IolIsBusy(hr) THEN
    IF FAILED(hr) THEN
      bBusy := FALSE;
      bError := TRUE;
      sErrorTxt := fbIolSlave.ErrorTxt; // TODO
      nAdsError := F_IolGetAdsError(hr);
      nIolError := F_IolGetIolError(hr);
      nState := 0; // IDLE;
    ELSE
      nState := 20; // WRITE;
    END_IF
  END_IF

// READ
20:
  hr := fbIolSlave.ReadApplicationSpecificTag(sApplicationTagRead);

  IF NOT F_IolIsBusy(hr) THEN
    IF FAILED(hr) THEN
      bBusy := FALSE;
      bError := TRUE;
      sErrorTxt := fbIolSlave.ErrorTxt; // TODO
      nAdsError := F_IolGetAdsError(hr);
      nIolError := F_IolGetIolError(hr);
      nState := 0; // IDLE;
    ELSE
      nState := 30; // CHECK;
    END_IF
  END_IF

// CHECK
30:
  IF sApplicationTag <> sApplicationTagRead THEN
    bBusy := FALSE;
    bError := TRUE;
    sErrorTxt := 'sApplicationTag and sApplicationTagRead are not equal!';
  ELSE
    bBusy := FALSE;
    bDone := TRUE;
    nState := 0;
  END_IF
END_CASE

////
// call FBs
//
fbIolSlave();
```

7 Beispiele

Beispielcode und -konfigurationen für dieses Produkt können über das entsprechende Repository auf GitHub bezogen werden: https://github.com/Beckhoff/PlcLibrary_Tc3_IoLink_Sample. Sie haben dort die Möglichkeit das Repository zu clonen oder ein ZIP-File mit dem Sample herunterzuladen.

8 Anhang

8.1 Parameter Datenaustausch

Ein intelligenter IO-Link- Sensor/Aktuator kann eine Parametrierung durch ISDU (Indexed **S**ervice **D**ata **U**nit) unterstützen. Diese azyklischen Servicedaten müssen von der SPS explizit angefragt oder, als solche gekennzeichnet, gesendet werden.

● Zugang ISDU



TwinCAT unterstützt den Zugriff über ADS und über das CoE-Verzeichnis.

Über den sogenannten ISDU-Index wird der entsprechende Parameter adressiert. Verfügbar sind die Bereiche:

Bezeichnung	Bereich Index
System	0x00..0x0F
Identification	0x10..0x1F
Diagnostic	0x20..0x2F
Communication	0x30..0x3F
Prefered Index	0x40..0xFE
Extended Index	0x0100..0x3FFF
	der Bereich 0x4000..0xFFFF ist reserviert

Die Nutzung der Implementierung dieser Bereiche obliegt dem Sensor/Aktor-Hersteller. Zur Verdeutlichung sehen Sie hier nur einige mögliche Indexe mit Bezeichnung aufgeführt. Beachten Sie dazu im Infosystem das entsprechende Kapitel Objektbeschreibung und Parametrierung.

Direkte Parameter

Die Indexe 0x0000 und 0x0001 sind spezielle Parameter, die jedes IO-Link-Gerät besitzt. Ein IO-Link Master benötigt die Informationen aus dem Direct Parameter 1, um eine IO-Link Kommunikation mit dem Gerät aufzubauen.

General EtherCAT DC Process Data Plc Startup CoE - Online AoE - Online Diag History Online IO-Link

General X Port1::Parameter

Compare Read Write Set Default Export / Import Store

All Objects Enable Block Parametrization

Index	Name	Flags	Type	Value
0x0000:0	Direct Parameters 1	ro		>16<
0x0000:01	Reserved	ro	UINT8	0
0x0000:02	Master Cycle Time	ro	UINT8	5
0x0000:03	Min Cycle Time	ro	UINT8	5
0x0000:04	M-Sequence Capability	ro	UINT8	1
0x0000:05	IO-Link Version ID	ro	UINT8	17
0x0000:06	Process Data Input Length	ro	UINT8	80
0x0000:07	Process Data Output Length	ro	UINT8	16
0x0000:08	Vendor ID 1	ro	UINT8	1
0x0000:09	Vendor ID 2	ro	UINT8	32
0x0000:0A	Device ID 1	ro	UINT8	35
0x0000:0B	Device ID 2	ro	UINT8	57
0x0000:0C	Device ID 3	ro	UINT8	0
0x0000:0D	Reserved	ro	UINT8	0
0x0000:0E	Reserved	ro	UINT8	0
0x0000:0F	Reserved	ro	UINT8	255
0x0000:10	System Command	ro	UINT8	255
0x0001:0	Direct Parameters 2	rw		>16<
0x0001:01	Device Specific Parameter 1	rw	UINT8	0
0x0001:02	Device Specific Parameter 2	rw	UINT8	0
0x0001:03	Device Specific Parameter 3	rw	UINT8	0
0x0001:04	Device Specific Parameter 4	rw	UINT8	0
0x0001:05	Device Specific Parameter 5	rw	UINT8	0
0x0001:06	Device Specific Parameter 6	rw	UINT8	0
0x0001:07	Device Specific Parameter 7	rw	UINT8	0
0x0001:08	Device Specific Parameter 8	rw	UINT8	0
0x0001:09	Device Specific Parameter 9	rw	UINT8	0
0x0001:0A	Device Specific Parameter 10	rw	UINT8	0
0x0001:0B	Device Specific Parameter 11	rw	UINT8	0
0x0001:0C	Device Specific Parameter 12	rw	UINT8	0
0x0001:0D	Device Specific Parameter 13	rw	UINT8	241
0x0001:0E	Device Specific Parameter 14	rw	UINT8	0
0x0001:0F	Device Specific Parameter 15	rw	UINT8	0
0x0001:10	Device Specific Parameter 16	rw	UINT8	0

Index	Subindex	Name	Flag	Type	Definition
Direct Parameter1					

Index	Subindex	Name	Flag	Type	Definition
0x0000	0x01	Reserved	ro	UINT8	Reserviert
0x0000	0x02	Master Cycle Time	ro	UINT8	Master Zykluszeit
0x0000	0x03	Min Cycle Time	ro	UINT8	Minimale Zykluszeit
0x0000	0x04	M-Sequence Capability	ro	UINT8	M-Sequenz
0x0000	0x05	IO-Link Version ID	ro	UINT8	IO-Link Revisions-ID
0x0000	0x06	Process Data Input Length	ro	UINT8	Prozessdateneingangsgroße
0x0000	0x07	Process Data Output length	ro	UINT8	Prozessdatenaustragsgroße
0x0000	0x08	Vendor ID 1	ro	UINT8	Hersteller-ID 1
0x0000	0x09	Vendor ID 2	ro	UINT8	Hersteller-ID 2
0x0000	0xA	Device ID1	ro	UINT8	Geräte-ID 1
0x0000	0xB	Device ID 2	ro	UINT8	Geräte-ID 2
0x0000	0xC	Device ID3	ro	UINT8	Geräte-ID 3
0x0000	0xD	Reserved	ro	UINT8	Reserviert
0x0000	0xE	Reserved	ro	UINT8	Reserviert
0x0000	0xF	Reserved	ro	UINT8	Reserviert
Direct Parameter 2					
0x0001	0x01	Device Specific Parameter 1	rw	UINT8	Gerätespezifisch
0x0001	0x02	Device Specific Parameter 2	rw	UINT8	Gerätespezifisch
0x0001	0x03	Device Specific Parameter 3	rw	UINT8	Gerätespezifisch
0x0001	0x04	Device Specific Parameter 4	rw	UINT8	Gerätespezifisch
0x0001	0x05	Device Specific Parameter 5	rw	UINT8	Gerätespezifisch
0x0001	0x06	Device Specific Parameter 6	rw	UINT8	Gerätespezifisch
0x0001	0x07	Device Specific Parameter 7	rw	UINT8	Gerätespezifisch
0x0001	0x08	Device Specific Parameter 8	rw	UINT8	Gerätespezifisch
0x0001	0x09	Device Specific Parameter 9	rw	UINT8	Gerätespezifisch
0x0001	0xA	Device Specific Parameter 10	rw	UINT8	Gerätespezifisch
0x0001	0xB	Device Specific Parameter 11	rw	UINT8	Gerätespezifisch
0x0001	0xC	Device Specific Parameter 12	rw	UINT8	Gerätespezifisch
0x0001	0xD	Device Specific Parameter 13	rw	UINT8	Gerätespezifisch
0x0001	0xE	Device Specific Parameter 14	rw	UINT8	Gerätespezifisch
0x0001	0xF	Device Specific Parameter 15	rw	UINT8	Gerätespezifisch
0x0001	0x10	Device Specific Parameter 16	rw	UINT8	Gerätespezifisch

Spezifizierte Parameter

Die Standard IO-Link Parameter sind zusätzlich hier gelistet.

0x0002	Standard Command	wo	UINT8	0
✓ 0x000C:0	Device Access Locks	rw s		>2<
✓ 0x000C:01	Parameter (write) Acc...	rw s	BOOL	0
✓ 0x000C:02	Data Storage Lock	rw s	BOOL	0
✓ 0x000C:03	Local Parameterizatio...	rw s	BOOL	0
✓ 0x000C:04	Local User Interface ...	rw s	BOOL	0
✓ 0x0010	Vendor Name	ro	String	Beckhoff Automation GmbH ...
✓ 0x0011	Vendor Text	ro	String	www.beckhoff.com
✓ 0x0012	Product Name	ro	String	EPI2338-0001
✓ 0x0013	Product ID	ro	String	EPI2338-0001
✓ 0x0014	Product Text	ro	String	8 DIO Module
✓ 0x0015	Serial Number	ro	String	9362
✓ 0x0016	Hardware Version	ro	String	01
✓ 0x0017	Firmware Version	ro	String	05
✓ 0x0018	Application Specific T...	rw	String	*****

Index	Subindex	Name	Flag	Type	Definition
Device Access Locks					
0x000C	0x01	Parameter (Write) Access Lock	rw s	BOOL	Sperrt den Schreibzugriff
0x000C	0x02	Data Storage Lock	rw s	BOOL	Sperrt den Data Storage Zugriff
0x000C	0x03	Local Parametrization Lock	rw s	BOOL	Sperrt die lokalen Parameterzugriff
0x000C	0x04	Local Userinterface Lock	rw s	BOOL	Sperrt das lokale Userinterface
Erkennungsparameter					
0x0010	0x00	Vendor Name	ro	STRING	Herstellername
0x0011	0x00	Vendor Text	ro	STRING	Hersteller Text
0x0012	0x00	Product Name	ro	STRING	Produktname
0x0013	0x00	Product ID	ro	STRING	Produkt-ID
0x0014	0x00	Product Text	ro	STRING	Produkt Text
0x0015	0x00	Serial Number	ro	STRING	Seriennummer
0x0016	0x00	Hardware Version	ro	STRING	Hardwareversion
0x0017	0x00	FirmwareVersion	ro	STRING	Firmwareversion
0x0018	0x00	ApplicationSpecificTag	rw	STRING	Applikationsspezifischer Text
0x0019	0x00	Function Tag	rw	STRING	Funktionstext

Device Status und Detailed Device Status

Die Parameter 0x0024 Device Status und 0x0025 Detailed Device Description sind bei standartmäßigen IO-Link Geräten ein optionaler Parameter. Wenn ein IO-Link Gerät allerdings Profile unterstützt, sind diese Parameter zwingend notwendig.

Index	Subindex	Name	Flag	Type	Definition
0x0024	0x00	Device Status	ro	Uint8	Gerätestatus
Detailed Device Status					
0x0025	0x01	Detailed Device Status	ro s	OctetString	Event eines IO-Link Geräts

Index	Subindex	Name	Flag	Type	Definition
0x0025	0x02	Detailed Device Status	ro s	OctetString	Event eines IO-Link Geräts
0x0025	0x03	Detailed Device Status	ro s	OctetString	Event eines IO-Link Geräts
0x0025	0x04	Detailed Device Status	ro s	OctetString	Event eines IO-Link Geräts

Wenn ein IO-Link Gerät den Parameter 0x0025 nicht unterstützt wird, kann die Event History zusätzlich aus dem CoE Parameter 0x10F3 per Baustein FB_EcCoeSdoRead ausgelesen werden.

The screenshot shows the 'CoE - Online' tab in the software interface. The 'Update List' section includes buttons for 'Update List', 'Advanced...', and 'Add to Startup...'. There are checkboxes for 'Auto Update', 'Single Update', and 'Show Offline Data'. A text box for 'Module OD (AoE Port):' contains the value '0'. Below this is a table of parameters:

Index	Name	Flags	Value
10F0:0	Backup parameter handling	RO	> 1 <
10F2	Backup parameter storage	RW	00 00 00 00
10F3:0	Diagnosis History	RO	> 21 <
10F3:01	Maximum Messages	RO	0x10 (16)
10F3:02	Newest Message	RO	0x11 (17)
10F3:03	Newest Acknowledged Message	RW	0x00 (0)
10F3:04	New Messages Available	RO	TRUE
10F3:05	Flags	RW	0x0000 (0)
10F3:06	Diagnosis Message 001	RO	01 E8 02 00 12 03 01 00 A0 3E ...
10F3:07	Diagnosis Message 002	RO	01 E8 02 00 12 03 01 00 80 96 D...
10F3:08	Diagnosis Message 003	RO	01 E8 02 00 12 03 01 00 60 8C B...
10F3:09	Diagnosis Message 004	RO	01 E8 02 00 12 03 01 00 C0 8F 8...
10F3:0A	Diagnosis Message 005	RO	01 E8 02 00 12 03 01 00 C0 81 8...
10F3:0B	Diagnosis Message 006	RO	01 E8 02 00 12 03 01 00 00 2F E...
10F3:0C	Diagnosis Message 007	RO	01 E8 02 00 12 03 01 00 00 1B 5...
10F3:0D	Diagnosis Message 008	RO	01 E8 02 00 12 03 01 00 40 64 8...
10F3:0E	Diagnosis Message 009	RO	01 E8 02 00 12 03 01 00 80 B7 7...
10F3:0F	Diagnosis Message 010	RO	01 E8 02 00 12 03 01 00 60 26 D...
10F3:10	Diagnosis Message 011	RO	01 E8 02 00 12 03 01 00 40 AC ...
10F3:11	Diagnosis Message 012	RO	01 E8 02 00 12 03 01 00 60 9E 1...
10F3:12	Diagnosis Message 013	RO	01 E8 02 00 12 03 01 00 A0 0C ...
10F3:13	Diagnosis Message 014	RO	01 E8 02 00 12 03 01 00 C0 61 5...
10F3:14	Diagnosis Message 015	RO	01 E8 02 00 12 03 01 00 20 4A 6...
10F3:15	Diagnosis Message 016	RO	01 E8 02 00 12 03 01 00 C0 77 4...
10F8	Actual Time Stamp	RO	0x4070f47f53a00
1600:0	IO RxPDO-Map Outputs Ch. 1	RW	> 9 <

8.2 Fehlercodes

8.2.1 IO-Link-Fehler Codes

Beim Auftreten eines Fehlers über ADS-Zugriff auf einen IO-Link Teilnehmer werden Fehlercodes generiert.

Die möglichen Fehlercodes sind in den nachfolgenden Tabellen aufgeführt.

Beispiel eines AdsReturnCode

AdsReturnCode 0x80110700

- **80**: Device Application Error (IO-Link Spec),
- **11**: Index not Available (IO-Link Spec),
- **0700**: General ADS Error

ErrorTypes (IO-Link Spec)

Incident	Error Code	Additional Code	Name	Definition
Device application error – no details	0x80	0x00	APP_DEV	This ErrorType shall be used if the requested service has been refused by the Device application and no detailed
Index not available	0x80	0x11	IDX_NOTAVAIL	This ErrorType shall be used whenever a read or write access occurs to a not existing Index.
Subindex not available	0x80	0x12	SUBIDX_NOTAVAIL	This ErrorType shall be used whenever a read or write access occurs to a not existing Subindex.
Service temporarily not available	0x80	0x20	SERV_NOTAVAIL	This ErrorType shall be used if a parameter is not accessible for a read or write service due to the current state of the Device application.
Service temporarily not available – local control	0x80	0x21	SERV_NOTAVAIL_LOCTRL	This ErrorType shall be used if a parameter is not accessible for a read or write service due to an ongoing local operation at the Device (for example operation or parameterization via an on-board Device control panel).
Service temporarily not available – Device control	0x80	0x22	SERV_NOTAVAIL_DEVCTRL	This ErrorType shall be used if a read or write service is not accessible due to a remote triggered state of the device application (for example parameterization during a remote triggered teach-in operation or calibration).
Access denied	0x80	0x23	IDX_NOT_WRITEABLE	This ErrorType shall be used if a write service tries to access a read-only parameter.
Parameter value out of range	0x80	0x30	PAR_VALOUTOFRNG	This ErrorType shall be used for a write service to a parameter outside its permitted range of values.
Parameter value above limit	0x80	0x31	PAR_VALGTLIM	This ErrorType shall be used for a write service to a parameter above its specified value range.
Parameter value below limit	0x80	0x32	PAR_VALLTLIM	This ErrorType shall be used for a write service to a parameter below its specified value range.
Parameter length overrun	0x80	0x33	VAL_LENVERRUN	This ErrorType shall be used when the content of a write service to a parameter is greater than the parameter specified length. This ErrorType shall also be used, if a data object is too large to be processed by the Device application (for example ISDU buffer restriction).
Parameter length underrun	0x80	0x34	VAL_LENUNDRUN	This ErrorType shall be used when the content of a write service to a parameter is less than the parameter specified length (for example write access of an Unsigned16 value to an Unsigned32 parameter).
Function not available	0x80	0x35	FUNC_NOTAVAIL	This ErrorType shall be used for a write service with a command value not supported by the Device application (for example a SystemCommand with a value not implemented).
Function temporarily unavailable	0x80	0x36	FUNC_UNAVAILTEMP	This ErrorType shall be used for a write service with a command value calling a Device function not available due to the current state of the Device application (for example a SystemCommand).
Invalid parameter set	0x80	0x40	PAR_SETINVALID	This ErrorType shall be used if values sent via single parameter transfer are not consistent with other actual parameter settings (for example overlapping set points for a binary data setting)
Inconsistent parameter set	0x80	0x41	PAR_SETINCONSIST	This ErrorType shall be used at the termination of a block parameter transfer with ParamDownloadEnd or ParamDownloadStore if the plausibility check shows inconsistencies
Application not ready	0x80	0x82	APP_DEVNOTRDY	This ErrorType shall be used if a read or write service is refused due to a temporarily unavailable application (for example peripheral controllers during startup).
Vendor specific	0x81	0x00	UNSPECIFIC	This ErrorType will be propagated directly to higher level processing elements as an error (no warning) by the Master.
Vendor specific	0x81	0x01 to 0xFF	VENDOR_SPECIFIC	

Derived ErrorTypes (IO-Link Spec)

Incident	Error Code	Additional Code	Name	Definition
Master – Communication error	0x10	0x00	COM_ERR	The Master generates a negative service response with this ErrorType if a communication error occurred during a read or write service, for example the SDCI connection is interrupted.
Master – ISDU timeout	0x11	0x00	I-SERVICE_TIMEOUT	The Master generates a negative service response with this ErrorType, if a Read or Write service is pending longer than the specified I-Service timeout in the Master.
Device Event – ISDU error (DL, Error, single shot, 0x5600)	0x11	0x00	I-SERVICE_TIMEOUT	If the Master received an Event with the EventQualifier and the EventCode 0x5600, a negative service response indicating a service timeout is generated and returned to the requester (Master – ISDU timeout).
Device Event – ISDU illegal service primitive (AL, Error, single shot, 0x5800)	0x11	0x00	I-SERVICE_TIMEOUT	If the Master received an Event with the EventQualifier and the EventCode 0x5800, a negative service response indicating a service timeout is generated and returned to the requester (Master – ISDU timeout).
Master – ISDU checksum error	0x56	0x00	M_ISDU_CHECKSUM	The Master generates a negative service response with this ErrorType, if its data link layer detects an ISDU checksum error.
Master – ISDU illegal service primitive	0x57	0x00	M_ISDU_ILLEGAL	The Master generates a negative service response with this ErrorType, if its data link layer detects an ISDU illegal service primitive.
Device Event – ISDU buffer overflow (DL, Error, single shot, 0x5200)	0x80	0x33	VAL_LENORRUN	If the Master received an Event with the EventQualifier and the EventCode 0x5200, a negative service response indicating a parameter length overrun is generated and returned to the requester (see parameter length overrun) Events from legacy Devices shall be redirected in compatibility mode to this derived ErrorType

Weitere Möglichkeiten zur Fehlerdiagnose**Device State Inputs Device (0x1A05)**

Im PDO "Device Diag" (0xF101:0D), wird angezeigt, das min. ein Ereignis in der "Diag History" aufgetreten ist.

"Device State" ist das Standard-Statusbit für EtherCAT Slaves und zeigt z. B. Kommunikationsabbruch zu einen der Slaves auf.

Device State Inputs (0x1A04)

Hier wird der Status der IO-Link Teilnehmer am jeweiligen Port (siehe im Systemmanager Comment Feld) angezeigt.

Soll-/Ist Vergleich der Parameterobjekte

Die Indizes 0x90n0 (Info Data) können zur Validierung der Konfigurationsindizes 0x80n0 des angeschlossenen IO-Link-Geräts herangezogen werden.

Im Fehlerfall können diese Objekte genutzt werden, um die Konfiguration mit dem Istzustand zu vergleichen.

Lost Frame Counter

Der Lost Frame Counter im Objekt 0xA0n0:02 dient zur Diagnose der Übertragungsqualität. TwinCAT stellt hier die Möglichkeit bereit, Probleme z. B. mit der Verkabelung, EMV oder Stromversorgung zu diagnostizieren.

8.2.2 ADS Return Codes

Gruppierung der Fehlercodes:

Globale Fehlercodes: 0x0000 [▶ 77]... (0x9811_0000 ...)

Router Fehlercodes: 0x0500 [▶ 77]... (0x9811_0500 ...)

Allgemeine ADS Fehler: 0x0700 [▶ 78]... (0x9811_0700 ...)

RTime Fehlercodes: 0x1000 [▶ 79]... (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 oder erreichbar.
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_INVALIDAMSLLENGTH	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_TLSSSEND	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	Lockerter Speicher kann nicht zugewiesen werden.
0x501	1281	0x98110501	ROUTERERR_RESIZEMEMORY	Die Größe des Routerspeichers konnte nicht geändert 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.

Hex	Dec	HRESULT	Name	Beschreibung
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	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.
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-Synchronisation 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_NOTIFYHANDINVALID	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.
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.
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.

Hex	Dec	HRESULT	Name	Beschreibung
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_LICENSESETIMETOLONG	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_INVALIDPARG	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_SYNC TIMEOUT	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_NOMORESVM	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).
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_IRQNOTLESSOREQUAL	Aufruf einer Service-Funktion im falschen Kontext
0x1017	4119	0x98111017	RTERR_VMXNOTSUPPORTED	Intel VT-x Erweiterung wird nicht unterstützt.

Hex	Dec	HRESULT	Name	Beschreibung
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.

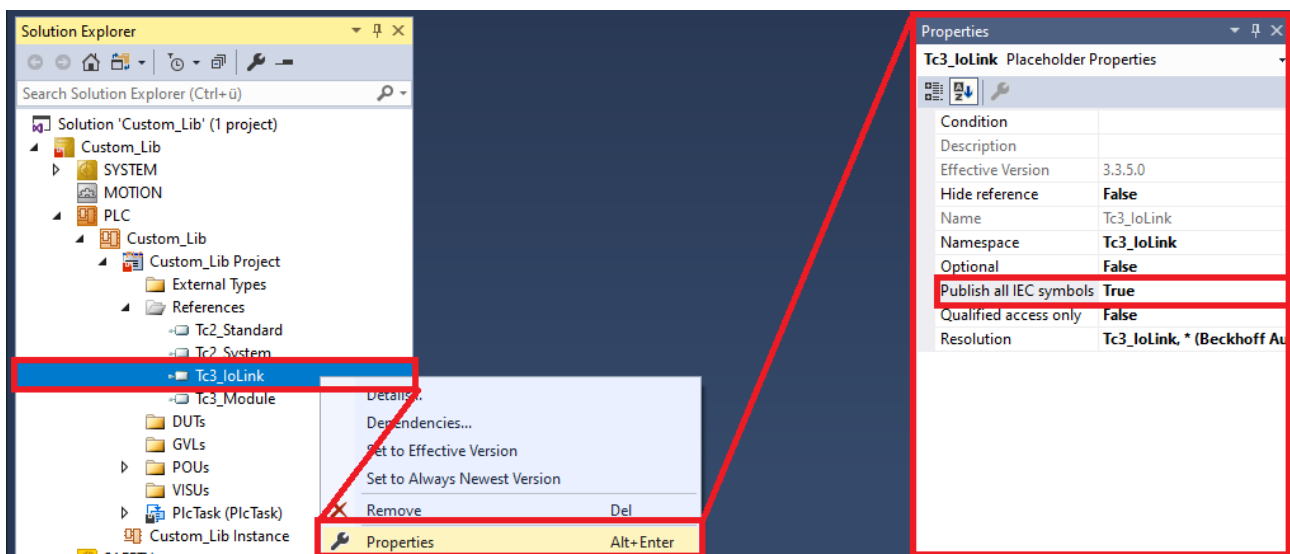
Weitere Winsock-Fehlercodes: Win32-Fehlercodes

8.3 Troubleshooting

Tc3_IoLink wird innerhalb einer weiteren Bibliothek (**Custom_Lib**) genutzt. Beim Einbinden findet der Compiler Symbole aus **Custom_Lib** nicht.

Per Default werden die Symbole einer eingebunden Lib nicht weiter veröffentlicht.
Dies lässt sich nachträglich ändern.

Öffnen Sie dazu das Eigenschaftfenster und wählen Sie **Publish all IEC symbols** aus:



Weitere Informationen finden Sie in der PLC Dokumentation unter Befehl Eigenschaften.

Mehr Informationen:
www.beckhoff.de/te1000

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Deutschland
Telefon: +49 5246 9630
info@beckhoff.com
www.beckhoff.com

