

Handbuch | DE

TX1200

TwinCAT 2 | PLC-Bibliothek: TcModbusTCPBC



Inhaltsverzeichnis

1	Vorwort	5
1.1	Hinweise zur Dokumentation	5
1.2	Sicherheitshinweise	6
1.3	Hinweise zur Informationssicherheit	7
2	Übersicht	8
3	Funktionsbausteine	9
3.1	FB_ModbusTcpOpen	9
3.2	FB_ModbusTcpClose	10
3.3	FB_ModbusTcpRequest	12
4	Anhang	14
4.1	BC9xxx return codes	14

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 

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.

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

1.2 Sicherheitshinweise

Sicherheitsbestimmungen

Beachten Sie die folgenden Sicherheitshinweise und Erklärungen!
Produktspezifische Sicherheitshinweise finden Sie auf den folgenden Seiten oder in den Bereichen Montage, Verdrahtung, Inbetriebnahme usw.

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.

Erklärung der Symbole

In der vorliegenden Dokumentation werden die folgenden Symbole mit einem nebenstehenden Sicherheitshinweis oder Hinweistext verwendet. Die Sicherheitshinweise sind aufmerksam zu lesen und unbedingt zu befolgen!

GEFAHR

Akute Verletzungsgefahr!

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

WARNUNG

Verletzungsgefahr!

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

VORSICHT

Schädigung von Personen!

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

HINWEIS

Schädigung von Umwelt oder Geräten

Wenn der Hinweis neben diesem Symbol nicht beachtet wird, können Umwelt oder Geräte geschädigt werden.

Tipp oder Fingerzeig

i Dieses Symbol kennzeichnet Informationen, die zum besseren Verständnis beitragen.

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 Bibliothek beinhaltet Funktionsbausteine die einen Datenaustausch zwischen dem **BC9xxx**-Bus-Controller und einem Remote-Partner auf dem MODBUS/TCP Port ermöglichen.

Bemerkungen

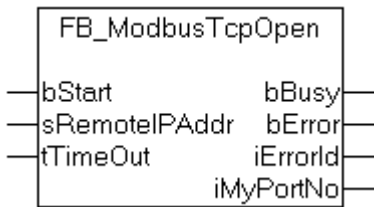
- Um den Ressourcenverbrauch zu minimieren, wurde die maximale Anzahl der gleichzeitig geöffneten Verbindungen auf 4 begrenzt. Um eine weitere Verbindung öffnen zu können muss eine der nicht benötigten Verbindungen zuerst geschlossen werden.
- Nach ca. 10 Sekunden wird eine in dieser Zeit nicht benutzte Verbindung automatisch geschlossen und muss neu geöffnet werden.

Funktionsbausteine

Name	Beschreibung
FB_ModbusTcpOpen [► 9]	Eine MODBUS/TCP-Verbindung zu einem Remote-Partner öffnen.
FB_ModbusTcpClose [► 10]	Eine MODBUS/TCP-Verbindung zu einem Remote-Partner schließen.
FB_ModbusTcpRequest [► 12]	Ein MODBUS-Request zu einem Remote-Partner senden.

3 Funktionsbausteine

3.1 FB_ModbusTcpOpen



Der Funktionsbaustein öffnet eine TCP/IP-Verbindung zu einem Remote-Partner auf dem MODBUS/TCP-Port. Um den Ressourcenverbrauch minimal zu halten wurde die maximale Anzahl der geöffneten Verbindungen auf 4 begrenzt. Um eine weitere Verbindung öffnen zu können, kann eine zur Zeit nicht benötigte Verbindung mit dem Funktionsbaustein [FB_ModbusTcpClose \[► 10\]](#) zuerst geschlossen werden. Der FB_ModbusTcpOpen-Funktionsbaustein liefert beim Erfolg in der *iMyPortNo*-Ausgangsvariable eine Portnummer. Diese Portnummer wird dazu benötigt, um mit dem [FB_ModbusTcpRequest \[► 12\]](#)- oder [FB_ModbusTcpClose \[► 10\]](#)-Funktionsbaustein einen MODBUS-Request zu senden, bzw. die Verbindung zu schließen. Mit dem Eingangsparameter *tTimeout* wird beim Öffnen der Verbindung auch die Timeout-Zeit für den MODBUS-Request und den Verbindungsabbau festgelegt. Nach ca. 10 Sekunden wird eine geöffnete und in dieser Zeit nicht benutzte Verbindung automatisch geschlossen und muss neu geöffnet werden.

VAR_INPUT

```
VAR_INPUT
    bStart      : BOOL;
    sRemoteIPAddr: STRING(15);
    tTimeOut    : TIME;
END_VAR
```

bStart: Durch eine steigende Flanke an diesem Eingang wird der Funktionsbaustein aktiviert.

sRemotelIPAddr: Die Variable ist ein String mit der Netzwerkadresse (IP-Adresse) des Remote-Partners.

tTimeOut: Gibt die Zeit an, die bei der Ausführung des Befehls nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy      : BOOL;
    bError     : BOOL;
    iErrorId   : WORD;
    iMyPortNo  : WORD;
END_VAR
```

bBusy: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

bError: Sollte ein Fehler bei der Ausführung des Befehls auftreten, dann wird dieser Ausgang gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde.

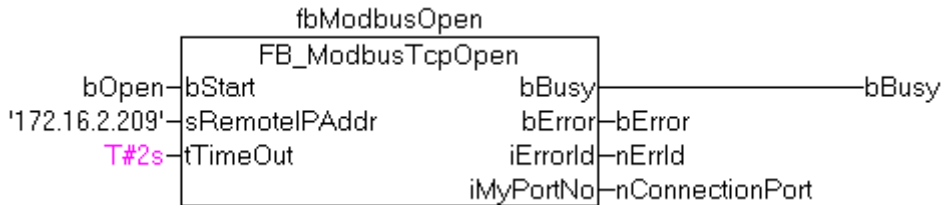
iErrorId: Liefert bei einem gesetzten bError-Ausgang den Befehlsspezifischen Fehlercode (Tabelle) oder den [gerätespezifischen Fehlercode \[► 14\]](#).

iMyPortNo : Beim Erfolg wird an diesem Ausgang die eigene zur Verbindung zugehörige Portnummer zurückgeliefert. Bei der Portnummer handelt es sich um eine dynamische Portnummer, die bei einem Verbindungsaufbau jedesmal neu erzeugt wird.

Fehlercodes	Beschreibung
0x8500	Resource error
0x8501	No socket available
0x8502	Open connection error

Beispiel für den Aufruf des Bausteins in FUP:

```
PROGRAM MAIN
VAR
    fbModbusOpen    : FB_ModbusTcpOpen;
    bOpen           : BOOL;
    bBusy           : BOOL;
    bError          : BOOL;
    nErrId         : WORD;
    nConnectionPort: WORD;
END_VAR
```

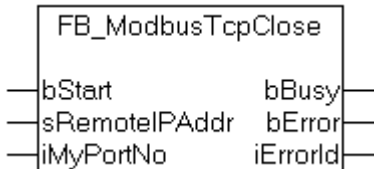


Im Beispiel wird bei einer steigenden Flanke am *bOpen*-Eingang die TCP/IP-Verbindung zu dem Remote-Teilnehmer mit der Netzwerkadresse: '172.16.2.209' aufgebaut.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.7.0 und abwärts	BC9xxx (165) firmware version >= 0xB6	TcModbusTCPBC.Lb6

3.2 FB_ModbusTcpClose



Mit dem FB_ModbusTcpClose-Funktionsbaustein wird eine mit dem [FB_ModbusTcpOpen \[► 9\]](#)-Funktionsbaustein geöffnete Verbindung zu einem Remote-Partner geschlossen. Um den Ressourcenverbrauch minimal zu halten wurde die maximale Anzahl der geöffneten Verbindungen auf 4 begrenzt. Um eine weitere Verbindung öffnen zu können, muss eine zurzeit nicht benötigte Verbindung zuerst geschlossen werden. Nach ca. 10 Sekunden wird eine Verbindung automatisch geschlossen, wenn sie in dieser Zeit nicht genutzt wurde. Beim Schließen der Verbindung wird nicht auf eine Bestätigung von dem Remote-Partner gewartet. D.h. es kann unter Umständen vorkommen (z.B. bei einer unterbrochenen Netzwerkkommunikation), dass eine Verbindung weiterhin besteht. Solche offenen Verbindungen sollten dann von dem Remote-Partner automatisch geschlossen werden. Ein Bus-Controller als Remote-Partner schließt solche offenen Verbindungen automatisch.

VAR_INPUT

```
VAR_INPUT
    bStart           : BOOL;
    sRemoteIPAddr   : STRING(15);
    iMyPortNo       : WORD;
END_VAR
```

bStart: Durch eine steigende Flanke an diesem Eingang wird der Funktionsbaustein aktiviert.

sRemotelPAddr: Die Variable ist ein String mit der Netzwerkadresse (IP-Adresse) des Remote-Partners dessen Verbindung geschlossen werden soll.

iMyPortNo: Hier wird die zur Verbindung zugehörige Portnummer als Parameter übergeben. Die Portnummer wird bei jedem Verbindungsaufbau mit dem [FB_ModbusTcpOpen \[▶ 9\]](#)-Funktionsbaustein jedes Mal neu (dynamisch) erzeugt.

VAR_OUTPUT

```
VAR_OUTPUT
  bBusy      : BOOL;
  bError     : BOOL;
  iErrorId   : WORD;
END_VAR
```

bBusy: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

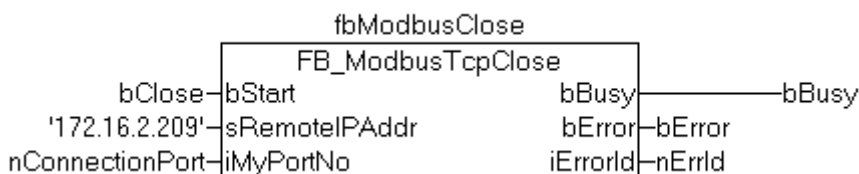
bError: Sollte ein Fehler bei der Ausführung des Befehls auftreten, dann wird dieser Ausgang gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde.

iErrorId: Liefert bei einem gesetzten bError-Ausgang den Befehlsspezifischen Fehlercode (Tabelle) oder den gerätespezifischen Fehlercode [[▶ 14](#)].

Fehlercodes	Beschreibung
0x8400	Connection is already closed.
0x8401	Connection close timeout.
0x8402	Connection close resource error.

Beispiel für den Aufruf des Bausteins in FUP:

```
PROGRAM MAIN
VAR
  fbModbusClose : FB_ModbusTcpClose;
  bOpen        : BOOL;
  bBusy        : BOOL;
  bError       : BOOL;
  nErrId       : WORD;
  nConnectionPort: WORD;
END_VAR
```

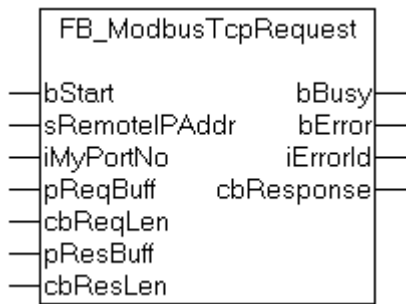


Im Beispiel wird bei einer steigenden Flanke am *bClose*-Eingang die TCP/IP-Verbindung zu dem Remote-Teilnehmer mit der Netzwerkadresse: '172.16.2.209' geschlossen.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.7.0 und abwärts	BC9xxx (165) firmware version >= 0xB6	TcModbusTCPBC.Lb6

3.3 FB_ModbusTcpRequest



Mit dem FB_ModbusTcpRequest-Funktionsbaustein kann ein MODBUS-Request an einen Remote-Partner gesendet werden. Die Verbindung muss dafür zuerst mit dem [FB_ModbusTcpOpen \[► 9\]](#)-Funktionsbaustein geöffnet worden sein. Die bei dem Verbindungsaufbau zurückgelieferte, dynamische Portnummer, wird als Eingangsparameter an die *iMyPortNo*-Variable übergeben. Die nicht benutzten Verbindungen werden nach ca. 10 Sekunden automatisch geschlossen oder können auch von dem Remote-Partner geschlossen worden sein. Bei einer fehlenden Verbindung wird z.B. an dem *iErrorId*-Ausgang der Fehlercode 0x8305 zurückgegeben. Mit dem *FB_ModbusTcpRequest*-Funktionsbaustein wird der gesamte MODBUS-Frame beginnend mit den UNIT ID übertragen. Der TCP-Anteil wird von dem BC9xxx-Bus-Controller vorangestellt.

VAR_INPUT

```
VAR_INPUT
  bStart      : BOOL;
  sRemotelIPAddr: STRING(15);
  iMyPortNo   : WORD;
  pReqBuff    : DWORD;
  cbReqLen    : WORD;
  pResBuff    : DWORD;
  cbResLen    : WORD;
END_VAR
```

bStart: Durch eine steigende Flanke an diesem Eingang wird der Funktionsbaustein aktiviert.

sRemotelIPAddr: Die Variable ist ein String mit der Netzwerkadresse (IP-Adresse) des Remote-Partners.

iMyPortNo: Hier wird die zur Verbindung zugehörige Portnummer als Parameter übergeben. Die Portnummer wird bei jedem Verbindungsaufbau mit dem [FB_ModbusTcpOpen \[► 9\]](#)-Funktionsbaustein jedes Mal neu (dynamisch) erzeugt.

pReqBuff : Enthält die Adresse des Puffers mit den MODBUS-Request-Daten. Der Programmierer ist selbst dafür verantwortlich, den Puffer in der Größe so zu dimensionieren, dass *cbReqLen*- Bytes daraus entnommen werden können. Der Puffer kann eine Einzelvariable, ein Array oder eine Struktur sein, dessen Adresse man mit dem ADR - Operator ermitteln kann.

cbReqLen : Bytelänge der MODBUS-Request Daten im *pReqBuff*-Puffer.

pResBuff : Enthält die Adresse des Puffers für die MODBUS-Response-Daten. Der Programmierer ist selbst dafür verantwortlich den Puffer in der Größe so zu dimensionieren, dass er *cbResLen*- Bytes aufnehmen kann. Der Puffer kann eine Einzelvariable, ein Array oder eine Struktur sein, dessen Adresse man mit dem ADR - Operator ermitteln kann.

cbResLen : Bytelänge des MODBUS-Response Puffers. Der Programmierer ist selbst dafür verantwortlich den Puffer in der Größe so zu dimensionieren, dass alle Response-Daten aufgenommen werden können.

VAR_OUTPUT

```
VAR_OUTPUT
  bBusy       : BOOL;
  bError      : BOOL;
  iErrorId    : WORD;
  cbResponse  : WORD;
END_VAR
```

bBusy: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

bError: Sollte ein Fehler bei der Ausführung des Befehls auftreten, dann wird dieser Ausgang gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde.

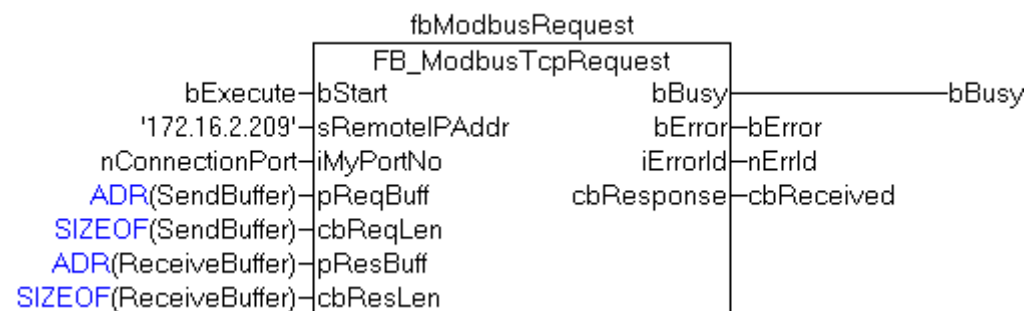
iErrorId: Liefert bei einem gesetzten bError-Ausgang den Befehlsspezifischen Fehlercode (Tabelle) oder den gerätespezifischen Fehlercode [► 14].

cbResponse : Die Bytelänge der tatsächlich im Response-Puffer zurückgelieferten Response-Daten.

Fehlercodes	Beschreibung
0x8300	Tx error. Timeout error.
0x8301	Tx error. Remote shutdown.
0x8302	Tx error. Internal error.
0x8303	Tx error. Resource error.
0x8305	Tx error. No remote connection.
0x8100	Rx error. Wrong transaction ID.
0x8101	Rx error. Wrong protocol ID.
0x8102	Rx error. Wrong length.
0x8103	Rx error. Exceeding length.
0x8104	Rx error. Wrong unit ID.

Beispiel für den Aufruf des Bausteins in FUP:

```
PROGRAM MAIN
VAR
    fbModbusRequest : FB_ModbusTcpRequest;
    bExecute        : BOOL;
    nConnectionPort: nConnectionPort;
    SendBuffer      : ARRAY[0..5] OF BYTE := 1, 6, 64, 64, 0, 0;
    ReceiveBuffer   : ARRAY[0..255] OF BYTE;
    bBusy           : BOOL;
    bError          : BOOL;
    nErrId          : WORD;
    cbReceived      : WORD;
END_VAR
```



Im Beispiel wird bei einer steigenden Flanke am *bExecute*-Eingang ein Request zu dem Remote-Teilnehmer mit der Netzwerkadresse: '172.16.2.209' gesendet.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.7.0 und abwärts	BC9xxx (165) firmware version >= 0xB6	TcModbusTCPBC.Lb6

4 Anhang

4.1 BC9xxx return codes

Wert (hex)	Beschreibung
0x8800	No valid network configuration.
0x8E00	Connection closed by target.
0x8F00	Resource error.

Wert (hex)	Beschreibung Receive Error
0x8100	Receive wrong transaction ID
0x8101	Receive wrong protocol ID
0x8102	Receive wrong lenght
0x8103	Receive exceeding lenght
0x8104	Receive wrong unit ID

Wert (hex)	Beschreibung Transmit Error
0x8300	Transmit timeout
0x8301	Transmit Shutdown
0x8302	Transmit internal error
0x8303	Transmit resource error
0x8305	Transmit remote not connected

Wert (hex)	Beschreibung Close Error
0x8400	Receive wrong transaction ID
0x8401	Receive wrong protocol ID
0x8402	Receive wrong lenght

Wert (hex)	Beschreibung Open Error
0x8500	Open ressoucre error
0x8501	Open ressoucre error_1
0x8502	Open connection error
0x8503	Open connection already open

Mehr Informationen:
www.beckhoff.de/tx1200

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

