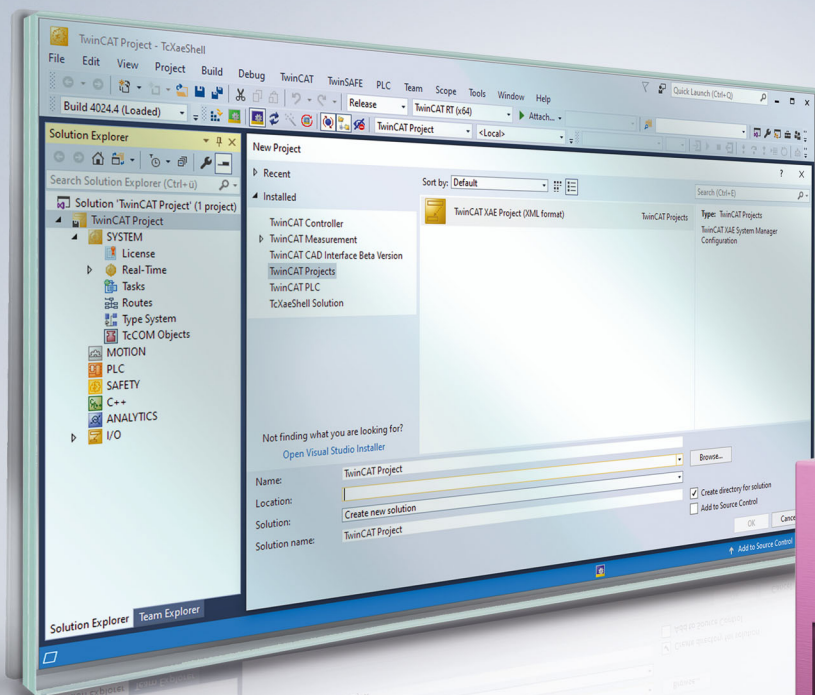


# BECKHOFF New Automation Technology

Handbuch | DE

# TF6350

TwinCAT 3 | SMS/SMTP





# Inhaltsverzeichnis

<b>1</b>	<b>Vorwort</b>	<b>5</b>
1.1	Hinweise zur Dokumentation	5
1.2	Sicherheitshinweise	6
1.3	Hinweise zur Informationssicherheit	7
<b>2</b>	<b>Übersicht</b>	<b>8</b>
<b>3</b>	<b>Installation</b>	<b>9</b>
3.1	Systemvoraussetzungen	9
3.2	Installation	9
3.3	Installation Windows CE	12
3.4	Lizenzierung	14
<b>4</b>	<b>TwinCAT SMTP</b>	<b>17</b>
4.1	Konfiguration	17
4.2	SPS API	18
4.2.1	Funktionsbausteine	19
4.3	Beispiele	32
4.3.1	Mailversand	32
4.3.2	SMTP Server Beispiele für die SPS	32
4.4	Anhang	33
4.4.1	Trouble-Shooting	33
4.4.2	Fehlercodes	34
4.4.3	Windows Socket Fehler Codes	36
<b>5</b>	<b>TwinCAT SMS</b>	<b>41</b>
5.1	SPS API	41
5.1.1	Funktionsbausteine	41
5.1.2	Funktionen	43
5.1.3	Globale Konstanten	43
5.2	Beispiele	43
5.2.1	Verschicken einer SMS über den Beckhoff 4G-Stick	43
5.3	Anhang	45
5.3.1	Fehlersuche	45
<b>6</b>	<b>Anhang</b>	<b>47</b>
6.1	ADS Return Codes	47
6.2	Support und Service	51



# 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**



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>.

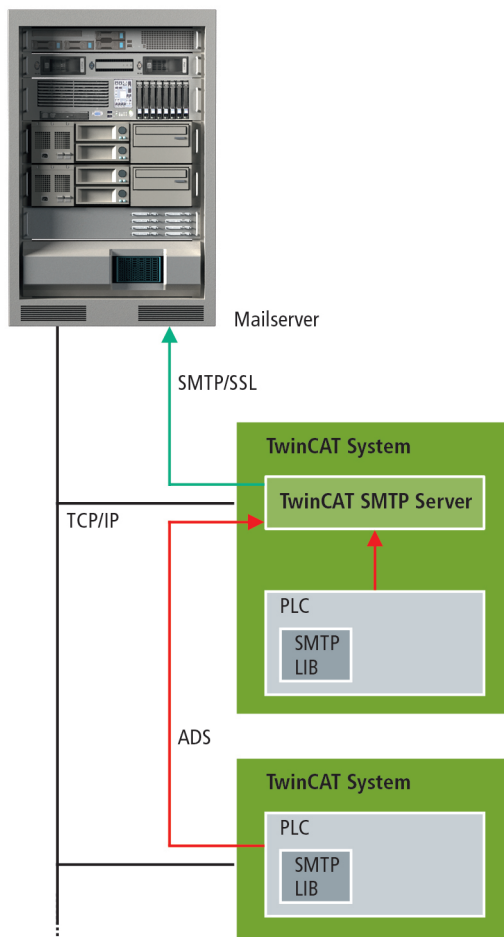
## 2 Übersicht

Diese Funktionalität ermöglicht das Versenden von Alarm- und Statusmeldungen direkt aus TwinCAT. Mit dieser Funktion können nach selbst definierten Bedingungen E-Mails oder SMS verschickt werden.

### SMTP Server:

Der TwinCAT SMTP Server dient zum Verschicken von E-Mail Nachrichten mit TwinCAT über ADS.

[TwinCAT SMTP Overview \[► 17\]](#)



### SMS Bibliothek:

Die TwinCAT SMS Bibliothek dient zum Verschicken von SMS-Nachrichten aus TwinCAT via GSM modem.

[TwinCAT SMS Overview \[► 41\]](#)



### 3 Installation

#### 3.1 Systemvoraussetzungen

Technische Daten	TF6350 TwinCAT3 SMS/SMTP
Zielsystem	WinXP, WES, Win7, WES7 IPC or CX, (x86)
Min. TwinCAT-Version	3.1.4000
Min. TwinCAT-Level	TC1200   TwinCAT 3 PLC

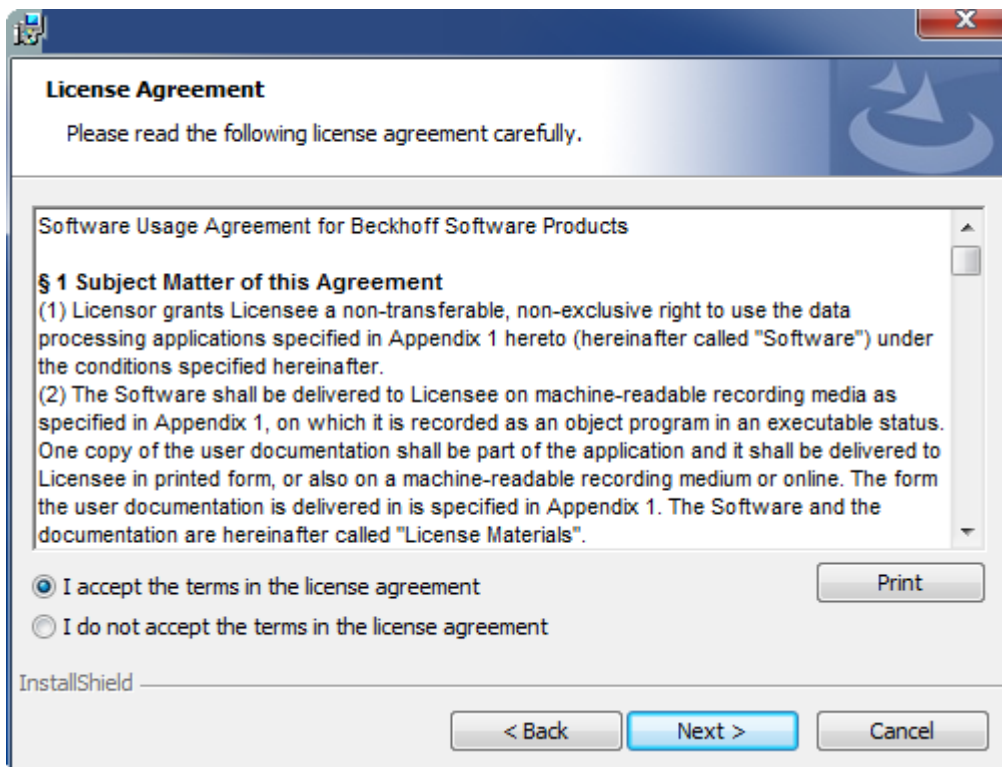
#### Voraussetzungen

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.1.4000	PC or CX (x86)	Tc2_Smtp, Tc2_Sms

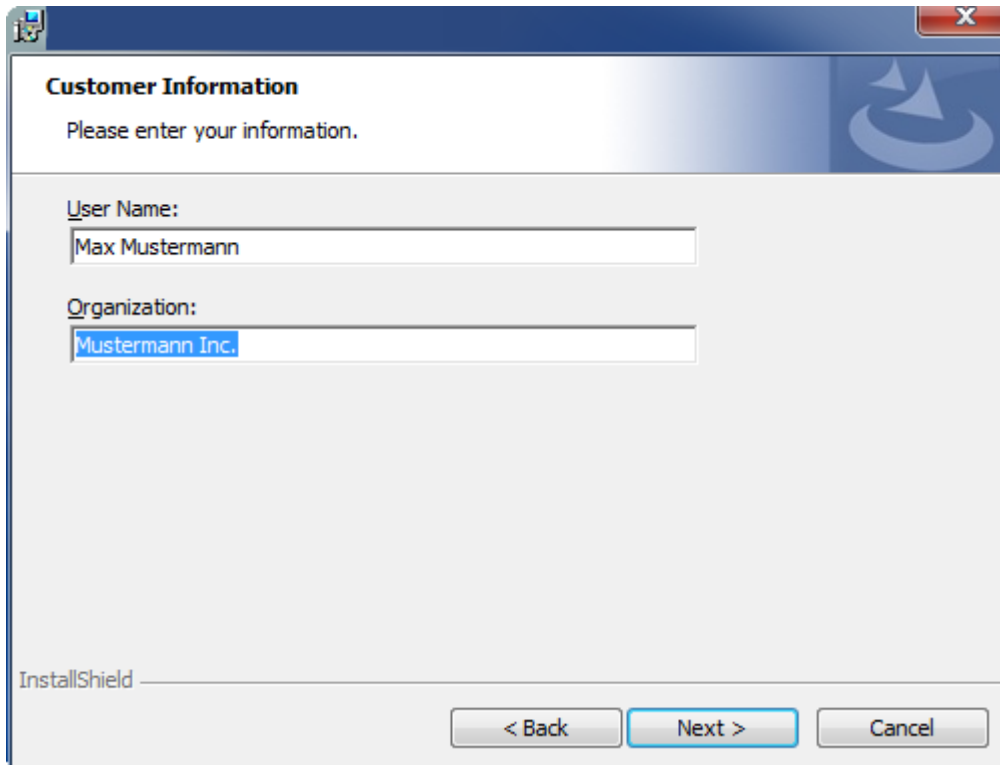
#### 3.2 Installation

Nachfolgend wird beschrieben, wie die TwinCAT 3 Function für Windows-basierte Betriebssysteme installiert wird.

- ✓ Die Setup-Datei der TwinCAT 3 Function wurde von der Beckhoff-Homepage heruntergeladen.
- 1. Führen Sie die Setup-Datei als Administrator aus. Wählen Sie dazu im Kontextmenü der Datei den Befehl **Als Administrator ausführen**.
  - ⇒ Der Installationsdialog öffnet sich.
- 2. Akzeptieren Sie die Endbenutzerbedingungen und klicken Sie auf **Next**.



3. Geben Sie Ihre Benutzerdaten ein.



**Customer Information**  
Please enter your information.

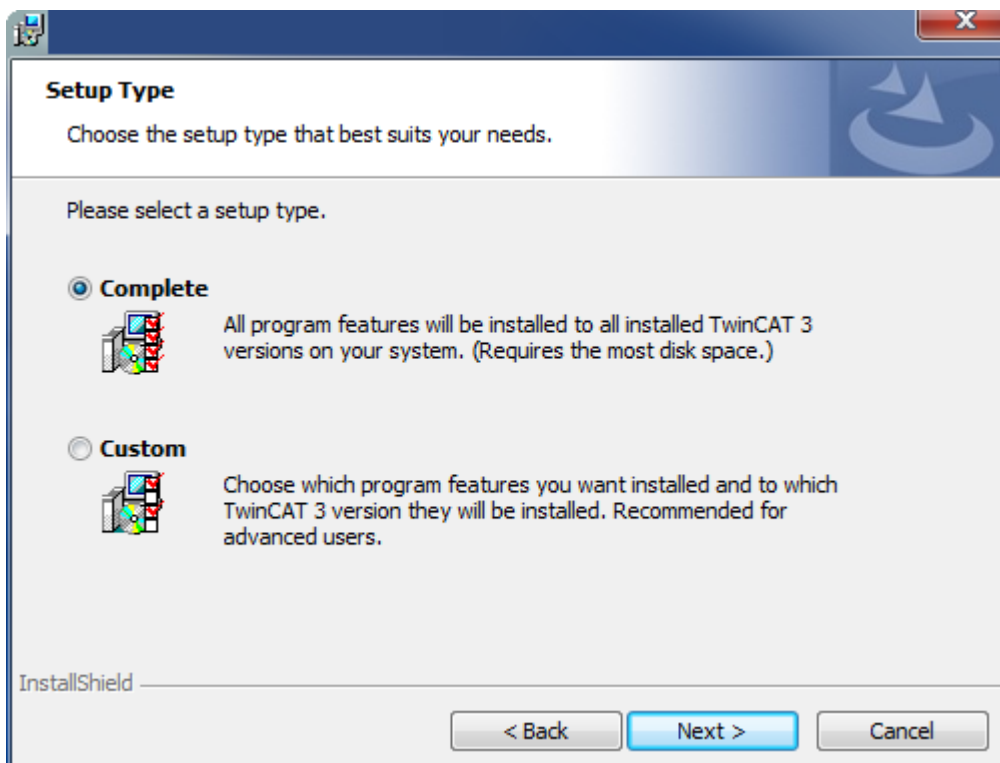
User Name:  
Max Mustermann

Organization:  
Mustermann Inc.

InstallShield


< Back   Next >   Cancel


4. Wenn Sie die TwinCAT 3 Function vollständig installieren möchten, wählen Sie **Complete** als Installationstyp. Wenn Sie die Komponenten der TwinCAT 3 Function separat installieren möchten, wählen Sie **Custom**.



**Setup Type**  
Choose the setup type that best suits your needs.

Please select a setup type.

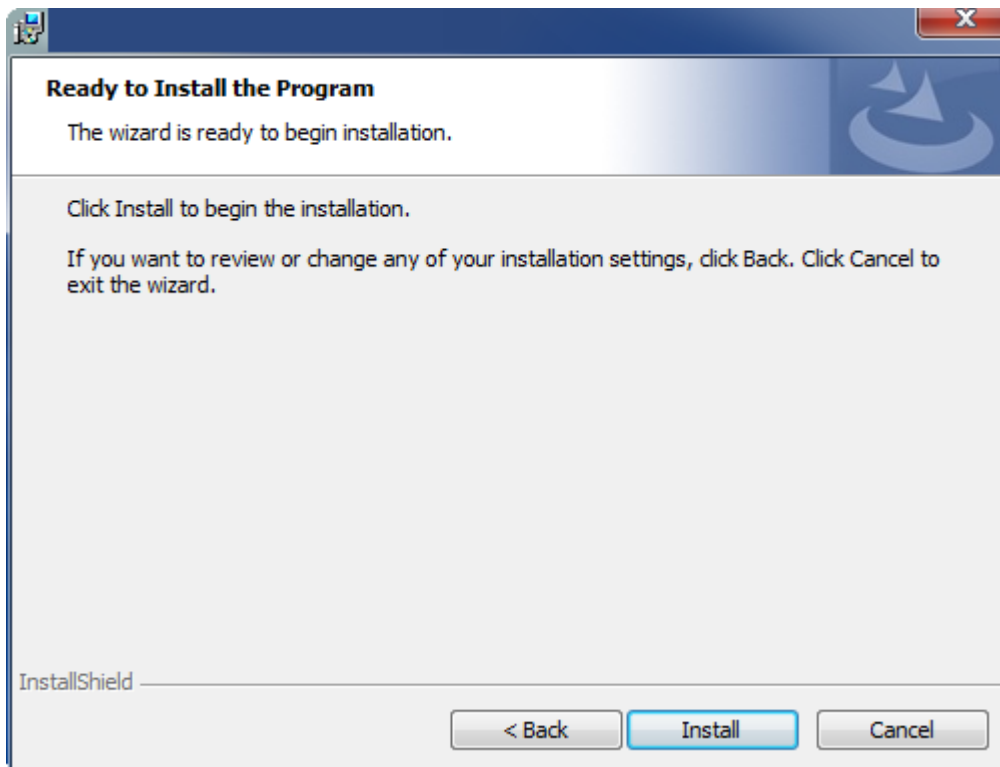
**Complete**  
 All program features will be installed to all installed TwinCAT 3 versions on your system. (Requires the most disk space.)

**Custom**  
 Choose which program features you want installed and to which TwinCAT 3 version they will be installed. Recommended for advanced users.

InstallShield

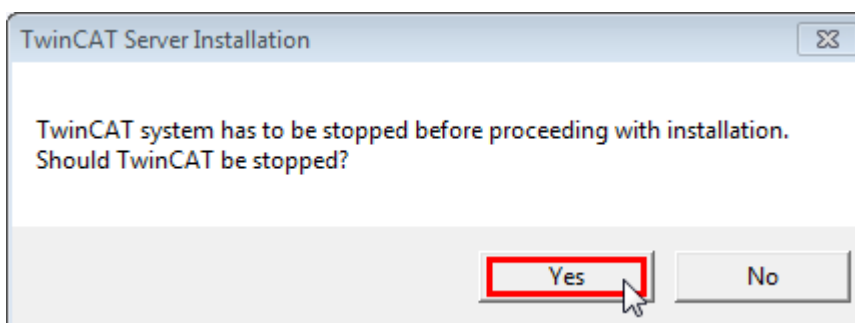
< Back   Next >   Cancel

5. Wählen Sie **Next** und anschließend **Install**, um die Installation zu beginnen.

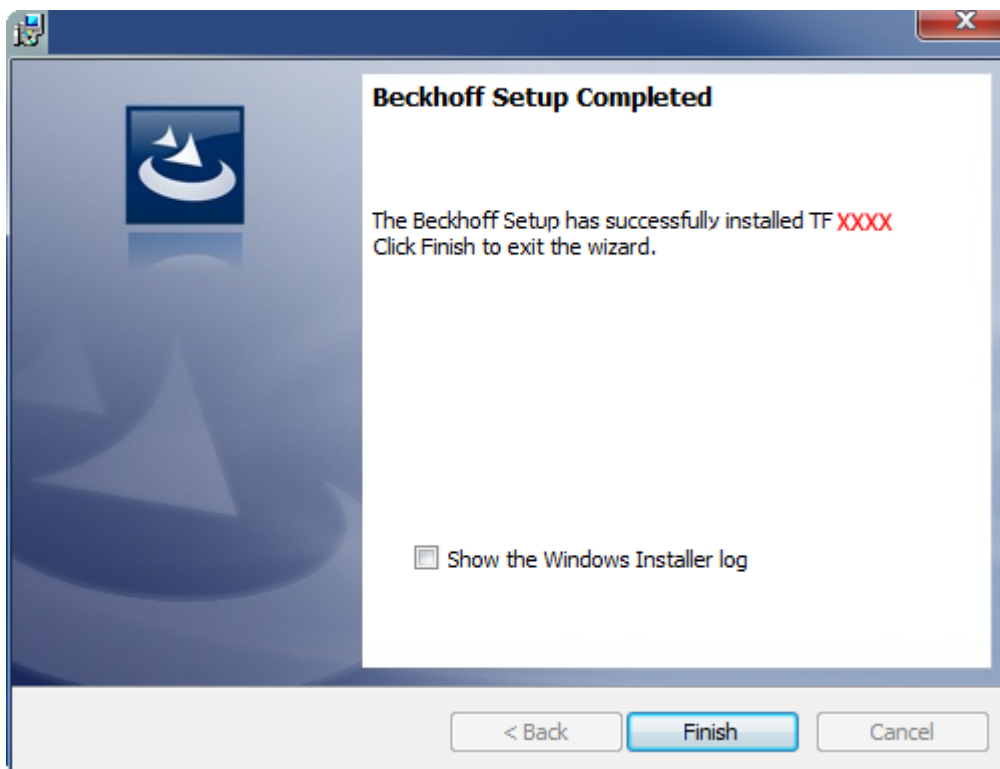


⇒ Ein Dialog weist Sie darauf hin, dass das TwinCAT-System für die weitere Installation gestoppt werden muss.

6. Bestätigen Sie den Dialog mit **Yes**.



7. Wählen Sie **Finish**, um das Setup zu beenden.



⇒ Die TwinCAT 3 Function wurde erfolgreich installiert und kann lizenziert werden (siehe [Lizenzierung](#) [► 14]).

### 3.3 Installation Windows CE

Dieser Teil der Dokumentation beschreibt, wie die TwinCAT 3 Function TF6350 SMS/SMTP auf einem Beckhoff Embedded Controller mit Windows CE installiert werden kann.

Der Setup Prozess besteht aus 4 Schritten:

- Download der Setup Datei
- Installation auf einem Host Computer
- Übertragung der ausführbaren Datei auf das Windows CE Gerät
- Installation der Software
- Upgrade der Software

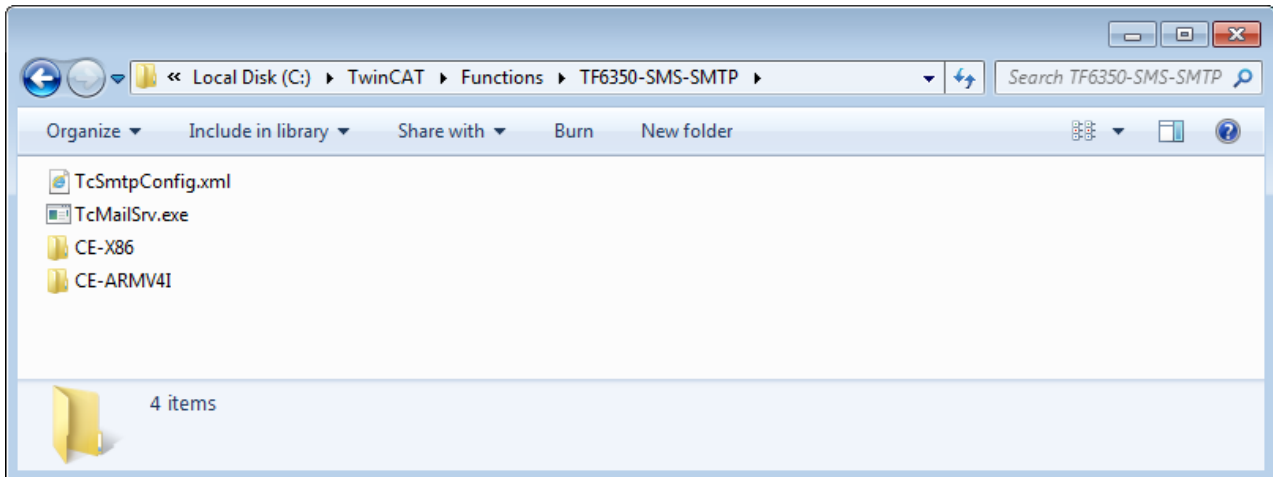
#### Download der Setup Datei

Die ausführbare Datei für Windows CE ist Teil des TF6350 SMS/SMTP Setups. Daher müssen Sie nur das entsprechende Setup von [www.beckhoff.com](http://www.beckhoff.com) beziehen, welches automatisch alle Versionen für Windows XP, Windows 7 und Windows CE (x86 und ARM) enthält.

#### Installation auf einem Host Computer

Nach der Installation enthält der Installationsordner drei Verzeichnisse - jeweils ein Verzeichnis pro Hardwareplattform:

- **CE-ARM:** ARM-basierte Embedded Controller, welche unter Windows CE laufen, z.B. CX8090, CX9020
- **CE-X86:** X86-basierte Embedded Controller, welche unter Windows CE laufen, z.B. CX50xx, CX20x0
- **Win32:** Embedded Controller, welche unter Windows XP, Windows 7 oder Windows Embedded Standard laufen



Die Verzeichnisse CE-ARM und CE-X86 enthalten die CAB-Dateien Dateien der TF6350 Function für Windows CE - in Bezug auf die jeweilige Hardwareplattform Ihres Windows CE Geräts. Diese ausführbaren Dateien müssen auf das Windows CE Gerät kopiert werden. Wie dies funktioniert erfahren Sie im nächsten Kapitel.

### Übertragung der ausführbaren Datei auf das Windows CE Gerät

Übertragen Sie die ausführbare Datei auf Ihr Windows CE Gerät. Zur Dateiübertragung stehen Ihnen mehrere Wege offen:

- über Netzwerkfreigaben
- über den integrierten FTP-Server
- über ActiveSync
- über CF/SD Karten

Für weitere Informationen konsultieren Sie bitte den Windows CE Bereich in unserem Information System.

### Installation der Software

Nachdem die Datei auf das Windows CE Gerät übertragen wurde, müssen Sie die Datei dort ausführen. Den darauf folgenden Installationsdialog können Sie mit OK bestätigen. Nachdem die Installation beendet wurde, starten Sie das CE Gerät neu.

Nachdem das Gerät neu gestartet wurde, werden die ausführbaren Dateien (Client und Server) automatisch im Hintergrund geladen und ist nun verfügbar.

Die Software wird in dem folgenden Verzeichnis auf dem CE-Gerät installiert: **Hard Disk\TwinCAT\Functions\ TF6350-SMS-SMTP**

### Upgrade der Software

Falls Sie schon eine ältere TF6350 SMS/SMTP Version auf dem Windows CE Gerät installiert haben, so müssen Sie die folgenden Schritte auf dem Windows CE Gerät durchführen um auf eine neuere Version zu upgraden:

- Öffnen Sie den CE Explorer, indem Sie auf Start --> Run klicken und "explorer" eingeben
- Navigieren Sie nach \Hard Disk\TwinCAT\Functions\TF6350-SMS-SMTP\Server
- Benennen Sie die Datei TcMail.Srv.exe um
- Starten Sie das Windows CE Gerät neu
- Übertragen Sie die neue CAB-Datei auf das Windows CE Gerät
- Installieren Sie die neue CAB-Datei
- Löschen Sie die alten (umbenannten) Dateien
- Starten Sie das Windows CE Gerät neu

Nachdem der Neustart durchgeführt wurde ist die neue Version aktiv.

Der nächste Schritt nach einer erfolgreichen Installation ist die Lizenzierung der TC3 Function.

## 3.4 Lizenzierung

Die TwinCAT 3 Function ist als Vollversion oder als 7-Tage-Testversion freischaltbar. Beide Lizenztypen sind über die TwinCAT-3-Entwicklungsumgebung (XAE) aktivierbar.

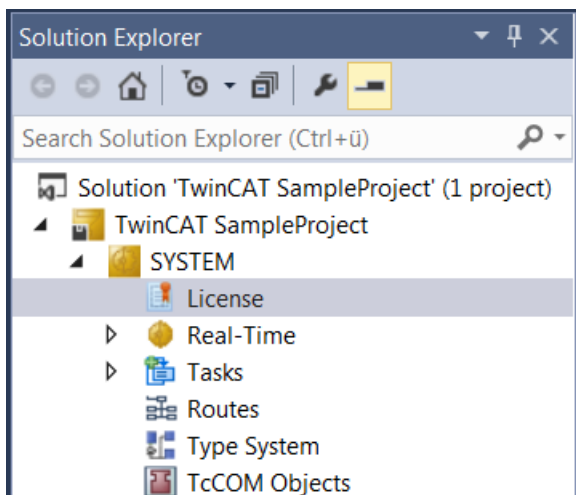
### Lizenzierung der Vollversion einer TwinCAT 3 Function

Die Beschreibung der Lizenzierung einer Vollversion finden Sie im Beckhoff Information System in der Dokumentation „[TwinCAT 3 Lizenzierung](#)“.

### Lizenzierung der 7-Tage-Testversion einer TwinCAT 3 Function

**i** Eine 7-Tage-Testversion kann nicht für einen TwinCAT 3 Lizenzdongle freigeschaltet werden.

1. Starten Sie die TwinCAT-3-Entwicklungsumgebung (XAE).
2. Öffnen Sie ein bestehendes TwinCAT-3-Projekt oder legen Sie ein neues Projekt an.
3. Wenn Sie die Lizenz für ein Remote-Gerät aktivieren wollen, stellen Sie das gewünschte Zielsystem ein. Wählen Sie dazu in der Symbolleiste in der Drop-down-Liste **Choose Target System** das Zielsystem aus.
  - ⇒ Die Lizenzierungseinstellungen beziehen sich immer auf das eingestellte Zielsystem. Mit der Aktivierung des Projekts auf dem Zielsystem werden automatisch auch die zugehörigen TwinCAT-3-Lizenzen auf dieses System kopiert.
4. Klicken Sie im **Solution Explorer** im Teilbaum **SYSTEM** doppelt auf **License**.



⇒ Der TwinCAT-3-Lizenzmanager öffnet sich.

- Öffnen Sie die Registerkarte **Manage Licenses**. Aktivieren Sie in der Spalte **Add License** das Auswahlkästchen für die Lizenz, die Sie Ihrem Projekt hinzufügen möchten (z. B. „TF4100 TC3 Controller Toolbox“).

Order No	License	Add License
TF3601	TC3 Condition Monitoring Level 2	<input type="checkbox"/> cpu license
TF3650	TC3 Power Monitoring	<input type="checkbox"/> cpu license
TF3680	TC3 Filter	<input type="checkbox"/> cpu license
TF3800	TC3 Machine Learning Inference Engine	<input type="checkbox"/> cpu license
TF3810	TC3 Neural Network Inference Engine	<input type="checkbox"/> cpu license
TF3900	TC3 Solar-Position-Algorithm	<input type="checkbox"/> cpu license
TF4100	TC3 Controller Toolbox	<input checked="" type="checkbox"/> cpu license
TF4110	TC3 Temperature-Controller	<input type="checkbox"/> cpu license
TF4500	TC3 Speech	<input type="checkbox"/> cpu license

- Öffnen Sie die Registerkarte **Order Information (Runtime)**.  
 ⇒ In der tabellarischen Übersicht der Lizenzen wird die zuvor ausgewählte Lizenz mit dem Status „missing“ angezeigt.
- Klicken Sie auf **7 Days Trial License...**, um die 7-Tage-Testlizenz zu aktivieren.

- Es öffnet sich ein Dialog, der Sie auffordert, den im Dialog angezeigten Sicherheitscode einzugeben.

- Geben Sie den Code genauso ein, wie er angezeigt wird, und bestätigen Sie ihn.
- Bestätigen Sie den nachfolgenden Dialog, der Sie auf die erfolgreiche Aktivierung hinweist.

⇒ In der tabellarischen Übersicht der Lizenzen gibt der Lizenzstatus nun das Ablaufdatum der Lizenz an.

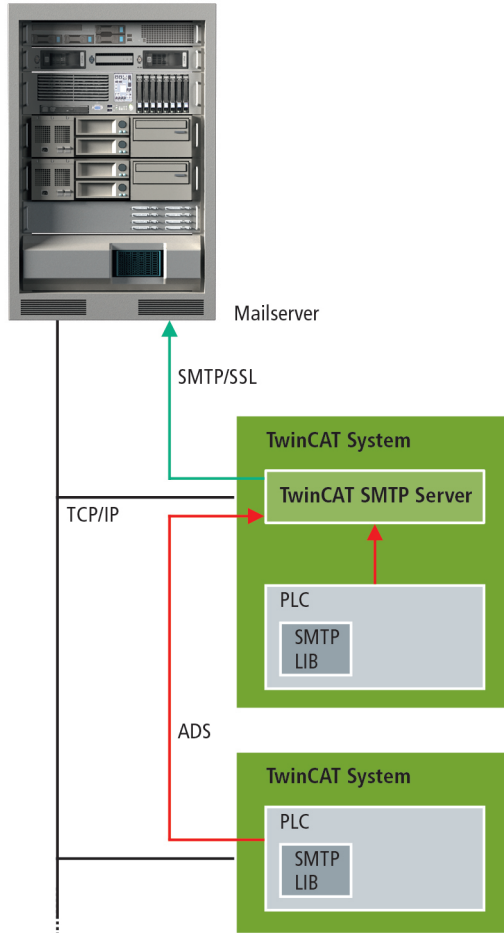
10. Starten Sie das TwinCAT-System neu.

⇒ Die 7-Tage-Testversion ist freigeschaltet.



## 4 TwinCAT SMTP

Mit dem TwinCAT SMTP Server können E-Mails direkt aus der SPS versendet werden. Dazu wird mit abgeschlossener Installation der ADS-Server zusammen mit TwinCAT gestartet. Der Server wird über ADS aus der SPS heraus angesprochen. [Mehrere Bausteine \[18\]](#) stehen für den Versand von E-Mails in der SPS zur Verfügung:



Der TwinCAT SMTP Server kann mit einem lokalen Mailserver oder externen Mail Providern kommunizieren. Er unterstützt die Verschlüsselungsmethoden TLS und STARTTLS.

### 4.1 Konfiguration

Der TC SMTP Server verwendet eine XML-basierende Konfiguration. Die **TcSmtplibConfig.xml** befindet sich im Installationsverzeichnis des Supplements.

Default-Konfiguration:

```
<TcSmtplibConfig>
  <!-- EnableLogFile: 0 (Disabled), 1 (Enabled) -->
  <EnableLogFile>0</EnableLogFile>
  <!-- LogSize: in Byte, 0 = use Default -->
  <LogSize>0</LogSize>
  <!-- Authentication: 0 (NONE), 1 (AUTO), 2 (LOGIN), 3 (NTLM), 4 (PLAIN) -->
  <Authentication>1</Authentication>
  <!-- Port: 0 (use default ports) -->
  <Port>0</Port>
  <!-- ContentEncoding: 0 (7BIT), 1 (8BIT), 2 (BINARY), 3 (BASE64), 4 (QUOTED_PRINTABLE) -->
  <ContentEncoding>0</ContentEncoding>
  <!-- Timeout for the socket connection -->
  <Timeout>8000</Timeout>
</TcSmtplibConfig>
```

```
<!-- Charset for the message content -->
<Charset>iso-8859-1</Charset>
<Reconnects>5</Reconnects>
</TcSmtplibConfig>
```

### Hinweis zur XML Konfigurationdatei

**EnableLogFile:** Es wird eine Logdatei im Installationsverzeichnis erstellt.

**Authentication:** Auswahl der Authentifizierungsmethode. Verwenden Sie die Option 1 als Default.

**Port:** Option 0 verwendet die Standard-Ports für den Mailversand.

**ContentEncoding:** Definiert das Entschlüsseln des Inhaltes.

**Timeout:** Timeout zum Mailversand in ms.

**Charset:** Definiert die Zeichenkodierung.

**Reconnects:** Anzahl der Wiederholungen.

### Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Smtp

## 4.2 SPS API

Die TwinCAT Bibliothek stellt Funktionsblöcke zum Versenden von E-Mails zum TwinCAT SMTP Server zur Verfügung.

Name	Beschreibung
<a href="#">FB_SmtpV3</a> [ <a href="#">▶ 19</a> ]	Funktionsblock zum Versenden von E-Mails zum TwinCAT SMTP Server
<a href="#">FB_SmtpV3_Full</a> [ <a href="#">▶ 21</a> ]	Funktionsblock mit weiteren Eigenschaften (z. B. Anhänge versenden, Prioritäten setzen, HTML-Meldungen)

### ● Auf aktuelle Version achten

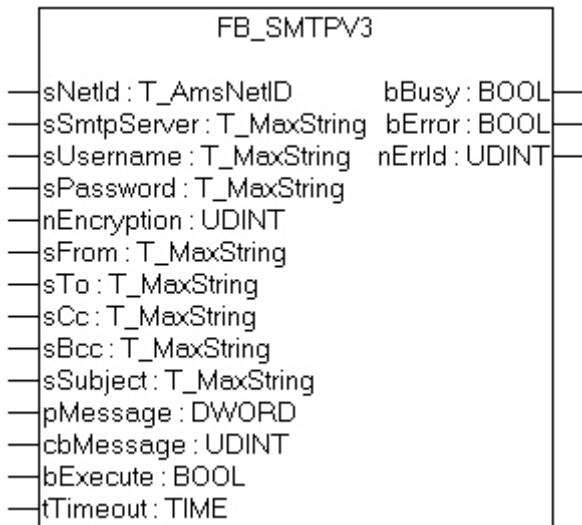
**i** Die Funktionsblöcke im als veraltet/abgekündigt gekennzeichneten Ordner sind überholt. Bitte verwenden Sie [FB\\_SmtpV3](#) und [FB\\_SmtpV3\\_Full](#), welche die gleichen Funktionalitäten bieten.

### Voraussetzungen

Entwicklungsumgebung	Zielsystem	Einzubindende SPS-Bibliotheken
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Smtp

## 4.2.1 Funktionsbausteine

### 4.2.1.1 FB\_SmtpV3



Der Block sendet einen Bytestream zu einem ADS Gerät. Der TwinCAT SMTP Server muss auf dem Zielsystem laufen, damit der Stream entgegengenommen und als E-Mail verschickt wird. Sobald das Bytestream versendet wurde, wird die E-Mail verschickt.

#### VAR\_INPUT

```

VAR_INPUT
  sNetId      : T_AmsNetID; (* AmsNetID *)
  sSmtpServer : T_MaxString; (* Smtip Server address ( IP or Name) *)
  sUsername   : T_MaxString; (* Smtip Username *)
  sPassword   : T_MaxString; (* Smtip Password *)
  nEncryption : UDINT;      (* 0=NONE, 1=STARTTLS, 2=TLS *)
  sFrom       : T_MaxString; (* Sender string *)
  sTo         : T_MaxString; (* To recipient string *)
  sCc         : T_MaxString; (* Cc recipient string *)
  sBcc        : T_MaxString; (* Bcc recipient string *)
  sSubject    : T_MaxString; (* Subject string *)
  pMessage    : POINTER TO STRING; (* Pointer to the message *)
  cbMessage   : UDINT;      (* Message lenght in byte to send *)
  bExecute    : BOOL;
  tTimeout    : TIME := T#20s;
END_VAR
    
```

**sNetId:** AmsNetID, auf dem der TwinCAT SMTP Server läuft.

**sSmtpServer:** Name oder IP des zu erreichenden SMTP Servers.

**sUsername:** Username des zu erreichenden SMTP Servers.

**sPassword:** Password des zu erreichenden SMTP Servers.

**nEncryption:** SMTP Verschlüsselungstyp:

0 = NONE

1 = STARTTLS

2 = TLS

**sFrom:** Ein String, der die E-Mail-Adresse oder den Namen des Absenders enthält. Der Absender muss festgelegt werden. Der String ist auf 255 Zeichen limitiert.

**sTo:** Ein String, der die E-Mail-Adresse des Empfängers enthält. Mindestens ein Empfänger muss eingetragen werden. Es ist aber auch möglich mehrere Adressen einzutragen. Diese müssen durch ein Semikolon getrennt werden. Der String ist auf 255 Zeichen limitiert.

**sCc:** Ein String, der die E-Mail-Adresse von weiteren Empfängern enthält (cc=carbon copy). Es ist möglich mehrere Adressen von Empfängern einzutragen, diese müssen dann durch ein Semikolon getrennt werden. Der String kann aber auch leer bleiben. Dem/den Empfänger(n) wird eine Kopie der E-Mail zugeschickt. Die E-Mail-Adresse des Empfängers ist für andere Empfänger **sichtbar**. Der String ist auf 255 Zeichen limitiert.

**sBcc:** Ein String der die E-Mail-Adressen von weiteren Empfängern enthält (Bcc = blind carbon copy). Es ist möglich mehrere Adressen von Empfängern einzutragen, diese müssen dann durch ein Semikolon getrennt werden. Der String kann aber auch leer bleiben. Dem/den Empfänger(n) wird eine Kopie der E-Mail zugeschickt. Die E-Mail-Adresse der Empfänger ist für andere Empfänger **nicht sichtbar**. Der String ist auf 255 Zeichen limitiert.

**sSubject:** Dieser String enthält den Betreff der E-Mail. Falls die E-Mail ohne Betreff gesendet wird, wird automatisch der Computername des Absenders in die Betreffzeile geschrieben (z.B. "E-Mail gesendet von: CX-00762C"). Der String der Betreffzeile ist auf 255 Zeichen limitiert.

**pMessage:** Dieser Parameter gibt die Adresse des Strings, welcher den Nachrichtentext enthält, an. Falls die E-Mail ohne Text gesendet wird, wird automatisch das Datum und die Uhrzeit eingesetzt (z.B. "Mail send at Thu, 23 Mar 2006 02:31:44 -0800"). Die Adresse des Strings kann mit dem ADR Operator festgelegt werden.

**cbMessage:** Länge des E-Mail-Textes. Die Länge kann durch den LEN Operator festgelegt werden.

**bExecute:** Der Funktionsbaustein wird durch eine steigende Flanke an dieser Eingangsvariablen aktiviert.

**tTimeout:** Die maximale erlaubte Zeit, um einen Befehl auszuführen.

## VAR\_OUTPUT

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

**bBusy:** Die Ausgangsvariable bleibt TRUE, bis der Block einen Befehl ausgeführt hat, allerdings nur bis zum Ablauf von tTimeOut.

**bError :** Die Ausgangsvariable wird auf TRUE umgeschaltet, sobald ein Fehler bei der Ausführung des Befehls auftritt. Der Befehls-spezifische Fehler ist in nErrId enthalten.

**nErrId:** Beinhaltet den befehlspezifischen Fehler Code des zuletzt ausgeführten Befehls



Stellen Sie sicher, dass Sie \0 nicht innerhalb der Byte-Anordnung verwenden, ansonsten wird die Nachricht gestoppt.

Die maximale Anzahl von Zeichen in einer Nachricht beträgt 510.725 - insgesamt haben Sie 1275 Zeichen für From, To, Cc, Bcc und Subject.

## Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Smtp

## 4.2.1.2 FB\_SmtpV3\_Full



Dieser Funktionsbaustein kommuniziert über ADS mit dem TwinCAT SMTP Server. Er bietet sehr umfangreiche E-Mail-Funktionalitäten, wie zum Beispiel die Priorisierung von E-Mails aus der SPS heraus. Im Detail werden die einzelnen Parameter in dieser Dokumentation erläutert.

## VAR\_INPUT

```

VAR_INPUT
  sNetId      : T_AmsNetID;      (* AmsNetID *)
  sSmtpServer : T_MaxString;     (* Smtip Server address ( IP or Name) *)
  sUsername   : T_MaxString;     (* Smtip Username *)
  sPassword   : T_MaxString;     (* Smtip Password *)
  nEncryption : UDINT;          (* 0=NONE, 1=STARTTLS, 2=TLS*)
  sFrom       : T_MaxString;     (* Sender stzring *)
  sTo         : T_MaxString;     (* To recipient string *)
  sCc         : T_MaxString;     (* Cc recipient string *)
  sBcc        : T_MaxString;     (* Bcc recipient string *)
  sDispositionNotification: T_MaxString; (* Disposition notification recipient string *)
  sReturnReceipt : T_MaxString; (* Return recipient string *)
  nPriority     : UDINT;         (* Priority value *)
  nSensitivity  : UDINT;         (* Sensitivity value *)
  nPort        : UDINT;         (* Communication port *)
  nContentType  : UDINT;         (* Content type *)
  sSubject     : T_MaxString;    (* Subject string *)
  pMessage     : POINTER TO STRING; (* Pointer to the message *)
  cbMessage    : UDINT;         (* Message lenght in byte to send *)
  sAttachments : ARRAY [0..32] OF STRING; (* Different attachments *)
  bExecute     : BOOL;          (* Trigger flag *)
  tTimeout     : TIME := T#20s; (* Communication timeout *)END_VAR

```

**sNetId:** AmsNetID, auf dem der TwinCAT SMTP Server läuft.

**sSmtpServer:** Name oder IP des zu erreichenden SMTP Servers.

**sUsername:** Benutzername des zu erreichenden SMTP Servers.

**sPassword:** Passwort für des zu erreichenden SMTP Servers.

**nEncryption:** SMTP Verschlüsselungstyp:

0 = NONE

1 = STARTTLS

2 = TLS

**sFrom:** Ein String, der die E-Mail-Adresse des Absenders enthält. Der Absender muss festgelegt werden. Der String ist auf 255 Zeichen limitiert.

**sTo:** Ein String, der die E-Mail-Adresse des Empfängers enthält. Mindestens ein Empfänger muss eingetragen werden. Es ist aber auch möglich mehrere Adressen einzutragen. Diese müssen per Semikolon getrennt werden. Der String ist auf 255 Zeichen limitiert.

**sCc:** Ein String, der die E-Mail-Adresse von weiteren Empfängern enthält (cc=carbon copy). Es ist möglich mehrere Adressen von Empfängern einzutragen, diese müssen dann durch ein Semikolon getrennt werden. Der String kann aber auch leer bleiben. Dem/den Empfänger(n) wird eine Kopie der E-Mail zugeschickt. Die E-Mail-Adresse des Empfängers ist für andere Empfänger **sichtbar**. Der String ist auf 255 Zeichen limitiert.

**sBcc:** Ein String, der die E-Mail-Adressen von weiteren Empfängern enthält (Bcc = blind carbon copy). Es ist möglich mehrere Adressen von Empfängern einzutragen, diese müssen dann durch ein Semikolon getrennt werden. Der String kann aber auch leer bleiben. Dem/den Empfänger(n) wird eine Kopie der E-Mail zugeschickt. Die E-Mail-Adresse der Empfänger ist für andere Empfänger **nicht sichtbar**. Der String ist auf 255 Zeichen limitiert.

**sDispositionNotification:** Die hier angegebene E-Mail-Adresse, erhält eine Lesebestätigung der Empfänger von sTo und sCc. Voraussetzung dafür ist jedoch, dass diese auch von den Empfängern gesendet wird.

**sReturnReceipt:** An die hier angegebene E-Mail-Adresse wird eine Übertragungsbestätigung der gesendeten E-Mail geschickt.

**nPriority:** Mit diesem Parameter können Sie die Priorität der E-Mail einstellen:

1 = Highest

2 = not used

3 = Normal

4 = not used

5 = Lowest

**nSensitivity:** Mit diesem Parameter können Sie die Vertraulichkeit der Nachricht einstellen:

0 = Private

1 = Personal

2 = Normal

3 = Confidential

**nPort:** Hier können Sie den Kommunikations-Port auswählen. Sollten Sie keinen eigenen Port eintragen, so wird auf den Default-Port 25 zurückgegriffen.

**nContentType:** Durch diesen Parameter ist es möglich beispielsweise HTML-Code, welcher per Pointer (pMessage) und Größe (cbMessage) einer Stringvariablen dem Baustein übergeben wird, in der E-Mail lesbar zu machen.

**sSubject:** Dieser String enthält den Betreff der E-Mail. Falls die E-Mail ohne Betreff gesendet wird, wird automatisch der Computername des Absenders in die Betreffzeile geschrieben (z.B. "E-Mail gesendet von: CX-00762C"). Der String der Betreffzeile ist auf 255 Zeichen limitiert.

**pMessage:** Dieser Parameter gibt die Adresse des Strings, welcher den Nachrichtentext enthält, an. Falls die E-Mail ohne Text gesendet wird, wird automatisch das Datum und die Uhrzeit eingesetzt (z.B. "Mail send at Thu, 23 Mar 2006 02:31:44 -0800"). Die Adresse des Strings kann mit dem ADR Operator ermittelt werden.

**cbMessage:** Länge des E-Mail-Textes. Die Länge kann durch den LEN Operator festgelegt werden.

**bExecute:** Der Funktionsbaustein wird durch eine steigende Flanke an dieser Eingangsvariablen aktiviert.

**sAttachments:** Auflistung von Anhängen (Pfad und Dateinamen) die verschickt werden sollen.

**tTimeout:** Die erlaubte, maximale Zeit, um einen Befehl auszuführen.

**VAR\_OUTPUT**

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

**bBusy:** Der Output bleibt TRUE, bis der Block einen Befehl ausgeführt hat, allerdings nur bis zum Ablauf von tTimeOut.

**bError :** Der Output wird auf TRUE umgeschaltet, sobald ein Fehler bei der Ausführung des Befehls auftaucht. Der Befehls-spezifische Fehler ist in nErrId enthalten.

**nErrId:** Beinhaltet den befehlspezifischen Fehler Code des zuletzt ausgeführten Befehls (siehe Tabelle [▶ 34]).



Stellen Sie sicher, dass Sie \0 nicht innerhalb der Byte-Anordnung verwenden, ansonsten wird die Nachricht gestoppt.

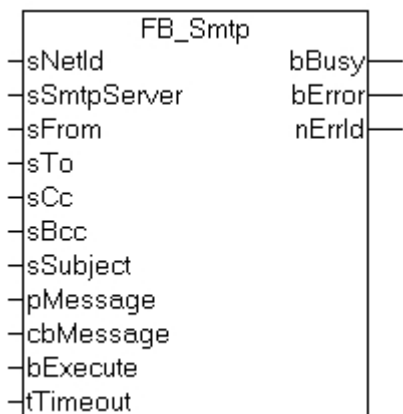
Die maximale Anzahl von Zeichen in einer Nachricht beträgt 510.725 - insgesamt haben Sie 1275 Zeichen für From, To, Cc, Bcc und Subject.

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Smtp

**4.2.1.3 [Abgekündigte Funktionen]**

**4.2.1.3.1 FB\_Smtp**



Der Baustein schickt per ADS einen Bytestream an ein ADS-Remotegerät. Auf dem ADS-Remotegerät muss der TwinCAT ADS SMTP Service laufen, um den Bytestream entgegenzunehmen und zu einer E-Mail zu verarbeiten. Nach der Bytesteam-Verarbeitung wird dann die E-Mail versendet.

Zu beachten ist, dass der SMTP-Server keine Passwort-Überprüfung unterstützt, da der TwinCAT ADS SMTP Service sich nicht per Passwort-Überprüfung am Server anmeldet.

**VAR\_INPUT**

```
VAR_INPUT
  sNetId      : T_AmsNetID; (* AmsNetID *)
  sSmtpServer : T_MaxString; (* Smtp-Server address (IP or Name) *)
  sFrom       : T_MaxString; (* Sender string *)
  sTo         : T_MaxString; (* To recipient string *)
  sCc         : T_MaxString; (* Cc recipient string *)
  sBcc        : T_MaxString; (* Bcc recipient string *)
  sSubject    : T_MaxString; (* Subject string *)
  pMessage    : DWORD;      (* Pointer to the message *)
```

```

cbMessage : UDINT;      (* Messagelenght to send *)
bExecute  : BOOL;
tTimeout  : TIME := T#20s;
END_VAR

```

**sNetId:** AmsNetID worauf der TwinCAT Mailserver läuft.

**sSmtpServer:** Name oder IP des SMTP-Servers.

**sFrom:** Ein String der die E-Mail-Adresse des Absenders enthält. Es muss ein Absender angegeben werden. Der String ist auf 255 Zeichen begrenzt.

**sTo:** Ein String der die E-Mail-Adressen der Empfänger beinhaltet. Es können mehrere Adressen angegeben werden, diese müssen dann mit einem Semikolon getrennt werden. Es muss mindestens ein Empfänger angegeben werden. Der String ist auf 255 Zeichen begrenzt.

**sCc:** Ein String mit einer E-Mail-Adresse eines weiteren Empfängers (Cc=Carbon Copy). Es kann auch ein Leerstring angegeben werden. Eine Kopie der E-Mail wird an diesen Empfänger gesendet. Die E-Mail-Adresse dieses Empfängers wird bei anderen Empfängern **sichtbar**. Es dürfen auch mehrere Empfängeradressen getrennt durch Semikolons angegeben werden. Der String ist auf 255 Zeichen begrenzt.

**sBcc:** Ein String mit einer E-Mail-Adresse eines weiteren Empfängers (Bcc=Blind Carbon Copy). Es kann auch ein Leerstring angegeben werden. Eine Kopie der E-Mail wird an diese Empfänger gesendet. Die E-Mail-Adresse dieses Empfängers wird bei anderen Empfängern nicht sichtbar. Es dürfen auch mehrere Empfängeradressen getrennt durch Semikolons angegeben werden. Der String ist auf 255 Zeichen begrenzt.

**sSubject:** Ein String mit dem Betreff der E-Mail. Die E-Mail kann auch ohne Betreff versendet werden. Sie bekommt dann automatisch den Namen vom versendenden Computer in den Betreff eingetragen (z.B. "Mail send from: CX\_00762C"). Der String für den Betreff ist auf 255 Zeichen begrenzt.

**pMessage:** Die Adresse (Pointer) eines nullterminierten Strings mit dem E-Mail-Text. Die E-Mail kann auch ohne Body-Text versendet werden, sie bekommt dann automatisch einen kleinen Satz mit dem Sendedatum und Uhrzeit eingetragen (z.B. "Mail send at: Thu, 23 Mar 2006 02:31:44 -0800"). Die Adresse des Strings kann mit dem ADR-Operator ermittelt werden.

**cbMessage:** Die Länge des E-Mail-Textes. Die Länge kann durch den LEN-Operator ermittelt werden.

**bExecute:** Eine steigende Flanke an diesem Eingang aktiviert den Funktionsbaustein.

**tTimeout:** Maximale Zeit, die bei der Ausführung des Befehls nicht überschritten werden darf.

## VAR\_OUTPUT

```

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

```

**bBusy:** Dieser Ausgang bleibt so lange auf TRUE, bis der Baustein eine Befehlsanforderung ausführt, längstens aber für die Dauer der an dem tTimeOut-Eingang angelegten Zeit.

**bError:** Dieser Ausgang wird auf TRUE geschaltet, wenn bei der Ausführung eines Befehls ein Fehler aufgetreten ist. Der befehlspezifische Fehlercode ist in iErrorId enthalten.

**nErrId:** Enthält den befehlspezifischen Fehlercode des zuletzt ausgeführten Befehls (siehe Tabelle [▶ 341](#)).



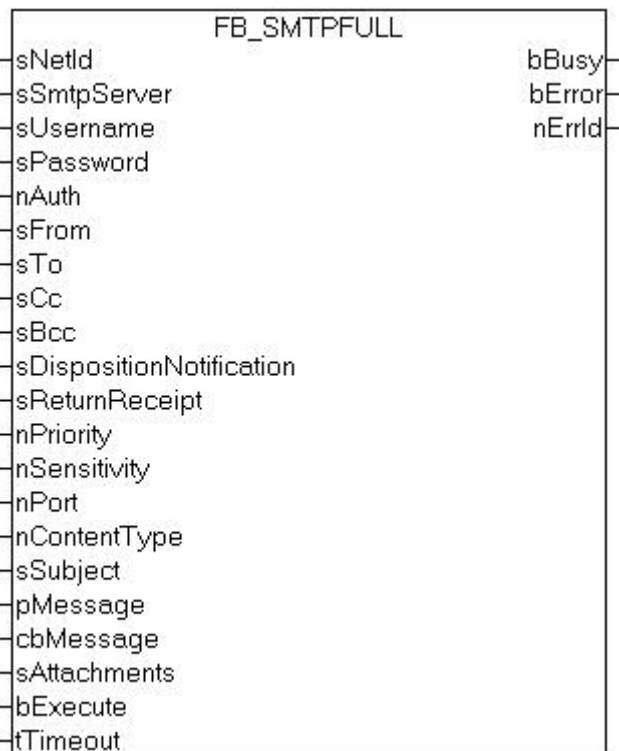


Stellen Sie sicher, dass Sie \0 nicht innerhalb der Byte-Anordnung verwenden, ansonsten wird die Nachricht gestoppt.  
 Die maximale Anzahl von Zeichen in einer Nachricht beträgt 510.725 - insgesamt haben Sie 1275 Zeichen für From, To, Cc, Bcc und Subject.

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Smtp

**4.2.1.3.2 FB\_SmtpFull**



Dieser Funktionsbaustein kommuniziert über ADS mit dem TwinCAT SMTP Server. Er bietet sehr umfangreiche E-Mail-Funktionalitäten, wie zum Beispiel die Priorisierung von E-Mails aus der SPS heraus. Im Detail werden die einzelnen Parameter in dieser Dokumentation erläutert.

**VAR\_INPUT**

```

VAR_INPUT
  sNetId      : T_AmsNetID;      (* AmsNetID *)
  sSmtpServer : T_MaxString;    (* Smtip Server addres ( IP or Name) *)
  sUsername   : T_MaxString;    (* Smtip Username *)
  sPassword   : T_MaxString;    (* Smtip Password *)
  nAuth       : UDINT;          (* Smtip Auth Type*)
  sFrom       : T_MaxString;    (* Sender stzring *)
  sTo         : T_MaxString;    (* To recipient string *)
  sCc         : T_MaxString;    (* Cc recipient string *)
  sBcc        : T_MaxString;    (* Bcc recipient string *)
  sDispositionNotification : T_MaxString; (* Disposition notification recipient string *)
  sReturnReceipt : T_MaxString; (* Return recipient string *)
  nPriority    : UDINT;          (* Priority value *)
  nSensitivity : UDINT;          (* Sensitivity value *)
  nPort       : UDINT;          (* Communication port *)
  nContentType : UDINT;          (* Content type *)
  sSubject    : T_MaxString;    (* Subject string *)
  pMessage    : DWORD;          (* Pointer to the message *)
  cbMessage   : UDINT;          (* Messagelenght in byte to send *)
  sAttachments : ARRAY [0..32] OF STRING; (* Different attachments *)
    
```

```
bExecute      : BOOL;          (* Trigger flag *)
tTimeout     : TIME := T#20s;  (* Communication timeout *)END_VAR
```

**sNetId:** AmsNetID, auf dem der TwinCAT SMTP Server läuft.

**sSmtpServer:** Name oder IP des SMTP Servers

**sUsername:** Benutzername des SMTP Servers

**sPassword:** Passwort für den SMTP Server.

**nAuth:** Smtplib Auth Type:

0 = AUTH NONE

1 = RESERVED

2 = AUTH LOGIN

3 = AUTH NTLM

4 = AUTH PLAIN

**sFrom:** Ein String, der die E-Mail-Adresse des Absenders enthält. Der Absender muss festgelegt werden. Der String ist auf 255 Zeichen limitiert.

**sTo:** Ein String, der die E-Mail-Adresse des Empfängers enthält. Mindestens ein Empfänger muss eingetragen werden. Es ist aber auch möglich mehrere Adressen einzutragen. Diese müssen per Semikolon getrennt werden. Der String ist auf 255 Zeichen limitiert.

**sCc:** Ein String, der die E-Mail-Adresse von weiteren Empfängern enthält (cc=carbon copy). Es ist möglich mehrere Adressen von Empfängern einzutragen, diese müssen dann durch ein Semikolon getrennt werden. Der String kann aber auch leer bleiben. Dem/den Empfänger(n) wird eine Kopie der E-Mail-Adresse zugeschickt. Die E-Mail-Adresse des Empfängers ist für andere Empfänger **sichtbar**. Der String ist auf 255 Zeichen limitiert.

**sBcc:** Ein String, der die E-Mail-Adressen von weiteren Empfängern enthält (Bcc = blind carbon copy). Es ist möglich mehrere Adressen von Empfängern einzutragen, diese müssen dann durch ein Semikolon getrennt werden. Der String kann aber auch leer bleiben. Dem/den Empfänger(n) wird eine Kopie der E-Mail zugeschickt. Die E-Mail-Adresse der Empfänger ist für andere Empfänger **nicht sichtbar**. Der String ist auf 255 Zeichen limitiert.

**sDispositionNotification:** Die hier angegebene E-Mail-Adresse, erhält eine Lesebestätigung der Empfänger von sTo und sCc. Voraussetzung dafür ist jedoch, dass diese auch von den Empfängern gesendet wird.

**sReturnReceipt:** An die hier angegebene E-Mail-Adresse wird eine Übertragungsbestätigung der gesendeten Mail geschickt.

**nPriority:** Mit diesem Parameter können Sie die Priorität der E-Mail einstellen:

1 = Highest

2 = not used

3 = Normal

4 = not used

5 = Lowest

**nSensitivity:** Mit diesem Parameter können Sie die Vertraulichkeit der Nachricht einstellen:

0 = Private

1 = Personal

2 = Normal

3 = Confidential

**nPort:** Hier können Sie den Kommunikations-Port auswählen. Sollten Sie keinen eigenen Port eintragen, so wird auf den Default-Port 25 zurückgegriffen.

**nContentType:** Durch diesen Parameter ist es möglich beispielsweise HTML-Code, welcher per Pointer (pMessage) und Größe (cbMessage) einer Stringvariablen dem Baustein übergeben wird, in der E-Mail lesbar zu machen.

**sSubject:** Dieser String enthält den Betreff der E-Mail. Falls die E-Mail ohne Betreff gesendet wird, wird automatisch der Computername des Absenders in die Betreffzeile geschrieben (z.B. "E-Mail gesendet von: CX\_00762C"). Der String der Betreffzeile ist auf 255 Zeichen limitiert.

**pMessage:** Dieser Parameter gibt die Adresse des Strings, welcher den Nachrichtentext enthält, an. Falls die E-Mail ohne Text gesendet wird, wird automatisch das Datum und die Uhrzeit eingesetzt (z.B. "Mail send at Thu, 23 Mar 2006 02:31:44 -0800"). Die Adresse des Strings kann mit dem ADR Operator ermittelt werden.

**cbMessage:** Länge des E-Mail-Textes. Die Länge kann durch den LEN Operator festgelegt werden.

**bExecute:** Der Funktionsbaustein wird durch eine steigende Flanke an dieser Eingangsvariablen aktiviert.

**sAttachments:** Auflistung von Dateinamen

**tTimeout:** Die erlaubte, maximale Zeit, um einen Befehl auszuführen.

**VAR\_OUTPUT**

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

**bBusy:** Der Output bleibt TRUE, bis der Block einen Befehl ausgeführt hat, allerdings nur bis zum Ablauf von tTimeOut.

**bError :** Der Output wird auf TRUE umgeschaltet, sobald ein Fehler bei der Ausführung des Befehls auftaucht. Der befehlspezifische Fehler ist in iErrorId enthalten.

**nErrId:** Beinhaltet den befehlspezifischen Fehlercode des zuletzt ausgeführten Befehls ([siehe Tabelle \[▶\\_34\]](#)).



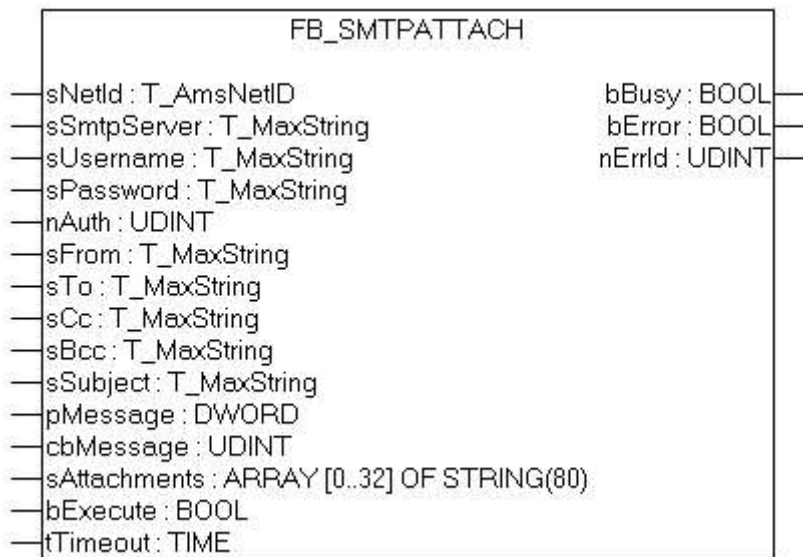
Stellen Sie sicher, dass Sie \0 nicht innerhalb der Byte-Anordnung verwenden, ansonsten wird die Nachricht gestoppt.

Die maximale Anzahl von Zeichen in einer Nachricht beträgt 510.725 - insgesamt haben Sie 1275 Zeichen für From, To, Cc, Bcc und Subject.

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Smtp

### 4.2.1.3.3 FB\_SmtpAttach



Der Baustein sendet einen Datenstrom zu einem entfernten ADS Gerät via ADS. Der TwinCAT ADS SMTP Service muss auf dem entfernten ADS Gerät laufen, so dass der Datenstrom empfangen und in eine E-Mail verarbeitet werden kann. Sobald der Datenstrom verarbeitet wurde, wird die E-Mail versendet.

#### VAR\_INPUT

```

VAR_INPUT
  sNetId      : T_AmsNetID;      (* AmsNetID *)
  sSmtpServer : T_MaxString;     (* Smtip Server address ( IP or Name) *)
  sUsername   : T_MaxString;     (* Smtip Username *)
  sPassword   : T_MaxString;     (* Smtip Password *)
  nAuth       : UDINT;          (* Smtip Auth Type*)
  sFrom       : T_MaxString;     (* Sender stzring *)
  sTo         : T_MaxString;     (* To recipient string *)
  sCc         : T_MaxString;     (* Cc recipient string *)
  sBcc        : T_MaxString;     (* Bcc recipient string *)
  sSubject    : T_MaxString;     (* Subject string *)
  pMessage    : DWORD;          (* Pointer to the message *)
  cbMessage   : UDINT;          (* Messagelenght in byte to send *)
  sAttachments : ARRAY [0..32] OF STRING;
  bExecute    : BOOL;
  tTimeout    : TIME := T#20s;
END_VAR

```

**sNetId:** AmsNetID auf dem der TwinCAT SMS Server läuft.

**sSmtpServer:** Name oder IP des SMTP Servers.

**sUsername:** Benutzername des SMTP Servers.

**sPassword:** Passwort für den SMTP Server.

**nAuth:** Smtip Auth Type:

- 0 = AUTH NONE
- 1 = RESERVED
- 2 = AUTH LOGIN
- 3 = AUTH NTLM
- 4 = AUTH PLAIN

**sFrom:** Ein String, der die E-Mail-Adresse des Absenders enthält. Der Absender muss festgelegt werden. Der String ist auf 255 Zeichen limitiert.

**sTo:** Ein String, der die E-Mail-Adresse des Empfängers enthält. Mindestens ein Empfänger muss eingetragen werden. Es ist aber auch möglich mehrere Adressen einzutragen. Diese müssen per Semikolon getrennt werden. Der String ist auf 255 Zeichen limitiert.

**sCc:** Ein String, der die E-Mail-Adresse von weiteren Empfängern enthält (cc=carbon copy). Es ist möglich mehrere Adressen von Empfängern einzutragen, diese müssen dann durch ein Semikolon getrennt werden. Der String kann aber auch leer bleiben. Dem/den Empfänger(n) wird eine Kopie der E-Mail zugeschickt. Die E-Mail-Adresse des Empfängers ist für andere Empfänger **sichtbar**. Der String ist auf 255 Zeichen limitiert.

**sBcc:** Ein String, der die E-Mail-Adressen von weiteren Empfängern enthält (Bcc = blind carbon copy). Es ist möglich mehrere Adressen von Empfängern einzutragen, diese müssen dann durch ein Semikolon getrennt werden. Der String kann aber auch leer bleiben. Dem/den Empfänger(n) wird eine Kopie der E-Mail zugeschickt. Die E-Mail-Adresse der Empfänger ist für andere Empfänger **nicht sichtbar**. Der String ist auf 255 Zeichen limitiert.

**sSubject:** Dieser String enthält den Betreff der E-Mail. Falls die E-Mail ohne Betreff gesendet wird, wird automatisch der Computername des Absenders in die Betreffzeile geschrieben (z.B. "E-Mail gesendet von: CX\_00762C"). Der String der Betreffzeile ist auf 255 Zeichen limitiert.

**pMessage:** Dieser Parameter gibt die Adresse des Strings, welcher den Nachrichtentext enthält, an. Falls die E-Mail ohne Text gesendet wird, wird automatisch das Datum und die Uhrzeit eingesetzt (z.B. "Mail send at Thu, 23 Mar 2006 02:31:44 -0800"). Die Adresse des Strings kann mit dem ADR Operator festgelegt werden.

**cbMessage:** Länge des E-Mail-Textes. Die Länge kann durch den LEN Operator festgelegt werden.

**bExecute:** Der Funktionsbaustein wird durch eine steigende Flanke an dieser Eingangsvariablen aktiviert.

**sAttachments:** Auflistung von Dateinamen

**tTimeout:** Die maximale, erlaubte Zeit, um einen Befehl auszuführen.

**VAR\_OUTPUT**

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

**bBusy:** Die Ausgangsvariable bleibt TRUE, bis der Block einen Befehl ausgeführt hat, allerdings nur bis zum Ablauf von tTimeOut.

**bError :** Die Ausgangsvariable wird auf TRUE umgeschaltet, sobald ein Fehler bei der Ausführung des Befehls auftaucht. Der befehlspezifische Fehler ist in iErrorId enthalten.

**nErrId:** Beinhaltet den befehlspezifischen Fehlercode des zuletzt ausgeführten Befehls ([siehe Tabelle 34](#)).



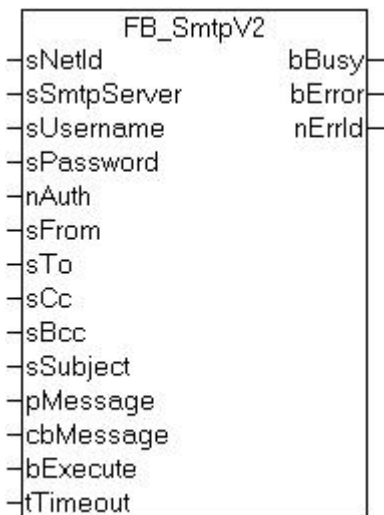
Stellen Sie sicher, dass Sie \0 nicht innerhalb der Byte-Anordnung verwenden, ansonsten wird die Nachricht gestoppt.

Die maximale Anzahl von Zeichen in einer Nachricht beträgt 510.725 - insgesamt haben Sie 1275 Zeichen für From, To, Cc, Bcc und Subject.

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Smtp

#### 4.2.1.3.4 FB\_SmtpV2



Der Baustein sendet einen Datenstrom zu einem entfernten ADS Gerät via ADS. Der TwinCAT ADS SMTP Service muss auf dem entfernten ADS Gerät laufen, so dass der Datenstrom empfangen und in eine E-Mail verarbeitet werden kann. Sobald der Datenstrom verarbeitet wurde, wird die E-Mail versendet.

#### VAR\_INPUT

```
VAR_INPUT
sNetId      : T_AmsNetID; (* AmsNetID *)
sSmtpServer : T_MaxString; (* Smtip Server address ( IP or Name) *)
sUsername   : T_MaxString; (* Smtip Username *)
sPassword   : T_MaxString; (* Smtip Password *)
nAuth       : UDINT;      (* Smtip Auth Type *)
sFrom       : T_MaxString; (* Sender stzring *)
sTo         : T_MaxString; (* To recipient string *)
sCc         : T_MaxString; (* Cc recipient string *)
sBcc        : T_MaxString; (* Bcc recipient string *)
sSubject    : T_MaxString; (* Subject string *)
pMessage    : DWORD;      (* Pointer to the message *)
cbMessage   : UDINT;      (* Message lenght in byte to send *)
bExecute    : BOOL;
tTimeout    : TIME := T#20s;
END_VAR
```

**sNetId:** AmsNetID auf dem der TwinCAT SMS Server läuft.

**sSmtpServer:** Name oder IP des SMTP Servers.

**sUsername:** Benutzername des SMTP Servers.

**sPassword:** Passwort für den SMTP Server.

**nAuth:** Smtip Auth Type:

0 = AUTH NONE  
 1 = RESERVED  
 2 = AUTH LOGIN  
 3 = AUTH NTLM  
 4 = AUTH PLAIN

**sFrom:** Ein String, der die E-Mail-Adresse des Absenders enthält. Der Absender muss festgelegt werden. Der String ist auf 255 Zeichen limitiert.

**sTo:** Ein String, der die E-Mail-Adresse des Empfängers enthält. Mindestens ein Empfänger muss eingetragen werden. Es ist aber auch möglich mehrere Adressen einzutragen. Diese müssen per Semikolon getrennt werden. Der String ist auf 255 Zeichen limitiert.

**sCc:** Ein String, der die E-Mail-Adresse von weiteren Empfängern enthält (cc=carbon copy). Es ist möglich

mehrere Adressen von Empfängern einzutragen, diese müssen dann durch ein Semikolon getrennt werden. Der String kann aber auch leer bleiben. Dem/den Empfänger(n) wird eine Kopie der E-Mail zugeschickt. Die E-Mail-Adresse des Empfängers ist für andere Empfänger **sichtbar**. Der String ist auf 255 Zeichen limitiert.

**sBcc:** Ein String, der die E-Mail Adressen von weiteren Empfängern enthält (Bcc = blind carbon copy). Es ist möglich mehrere Adressen von Empfängern einzutragen, diese müssen dann durch ein Semikolon getrennt werden. Der String kann aber auch leer bleiben. Dem/den Empfänger(n) wird eine Kopie der E-Mail zugeschickt. Die E-Mail-Adresse der Empfänger ist für andere Empfänger **nicht sichtbar**. Der String ist auf 255 Zeichen limitiert.

**sSubject:** Dieser String enthält den Betreff der E-Mail. Falls die E-Mail ohne Betreff gesendet wird, wird automatisch der Computername des Absenders in die Betreffzeile geschrieben (z.B. "E-Mail gesendet von: CX\_00762C"). Der String der Betreffzeile ist auf 255 Zeichen limitiert.

**pMessage:** Dieser Parameter gibt die Adresse des Strings, welcher den Nachrichtentext enthält, an. Falls die E-Mail ohne Text gesendet wird, wird automatisch das Datum und die Uhrzeit eingesetzt (z.B. "Mail send at Thu, 23 Mar 2006 02:31:44 -0800"). Die Adresse des Strings kann mit dem ADR Operator festgelegt werden.

**cbMessage:** Länge des E-Mail-Textes. Die Länge kann durch den LEN Operator festgelegt werden.

**bExecute:** Der Funktionsbaustein wird durch eine steigende Flanke an dieser Eingangsvariablen aktiviert.

**tTimeout:** Die maximale, erlaubte Zeit, um einen Befehl auszuführen.

**VAR\_OUTPUT**

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

**bBusy:** Die Ausgangsvariable bleibt TRUE, bis der Block einen Befehl ausgeführt hat, allerdings nur bis zum Ablauf von tTimeOut.

**bError :** Die Ausgangsvariable wird auf TRUE umgeschaltet, sobald ein Fehler bei der Ausführung des Befehls auftaucht. Der Befehls-spezifische Fehler ist in iErrorId enthalten.

**nErrId:** Beinhaltet den befehlspezifischen Fehlercode des zuletzt ausgeführten Befehls ([siehe Tabelle \[▶ 34\]](#)).



Stellen Sie sicher, dass Sie \0 nicht innerhalb der Byte-Anordnung verwenden, ansonsten wird die Nachricht gestoppt.

Die maximale Anzahl von Zeichen in einer Nachricht beträgt 510.725 - insgesamt haben Sie 1275 Zeichen für From, To, Cc, Bcc und Subject.

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Smtp

## 4.3 Beispiele

### 4.3.1 Mailversand

Mit dem Funktionsblock FB\_SmtpV3 ist der verschlüsselte Mailversand via TLS oder STARTTLs verfügbar.

#### Beschreibung

Eine E-Mail wird nach dem Triggern der Variable bStart versendet.

**ToDo:** Konfiguriere die Mailserver Adresse und die Zugangsdaten.

[Download Sample1 TC3 project](#)

#### Programmvariablen

```
PROGRAM MAIN
VAR
fbSendMail: FB_SmtpV3;
sMessage: STRING := 'Hello Beckhoff';
R_Edge: R_TRIG;
bStart: BOOL;
bBusy: BOOL;
bError: BOOL;
nErrID: UDINT;
bSend: BOOL;
nErr: UDINT;
nMailCounter: UDINT;
END_VAR
```

#### Programm-code

```
fbSendMail(
sNetId:= '',
sSmtpServer:= 'mail.company.com',
sUsername:= 'TestUser',
sPassword:= 'TestPwd',
sFrom:= 'TestUser@company.com',
sTo:= 'service@company.com',
sSubject:= 'Email from your Beckhoff PLC',
pMessage:= ADR(sMessage),
cbMessage:= SIZEOF(sMessage),
bExecute:= bStart,
bError=> bError,
bBusy=> bBusy,
nErrId=> nErrId);
IF NOT bBusy AND NOT bError AND bStart THEN
bStart := FALSE;
END_IF
```

#### Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Smtp

### 4.3.2 SMTP Server Beispiele für die SPS

Mit dem FB\_SmtpFull stehen sehr umfangreiche E-Mail-Funktionalitäten für die SPS zur Verfügung.

#### Beschreibung

In dem hier gezeigten Beispiel werden diese Möglichkeiten aufgezeigt:

- Es wird unter anderem der E-Mail-Text in HTML Code übergeben, was ganz neue Möglichkeiten der Formatierung bietet.



- Des Weiteren kann nun die Priorität einer E-Mail festgelegt oder Lesebestätigungen angefordert werden.
- Mehrere Dateien können als Anhang versendet werden.

Eine E-Mail wird nach dem Triggern der Variable bStart versendet.

**ToDo:** Konfiguriere die Mailserveradresse, die Zugangsdaten und die Pfade zu den Dateianhängen.

[Download Sample2 TC3 project](#)

**Programm-Variablen**

```
VAR
fbSmtplibFull      : FB_SmtplibV3_Full;
sMessage_HTML     : STRING(500) := '<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd"><html><head><title>HTML-Test</title><body><h3>Dear SMTP-
User,</h3>this e-mail was sent in HTML from <u>your PLC!</u><br/><br/>Best regards,<br/>your
Beckhoff-Team</body></html>';
nPriority          : UDINT;
R_Edge            : R_TRIG;
bStart            : BOOL;
bError            : BOOL;
udErrId           : UDINT;
nMailCounter      : UDINT := 0;
sFiles            : ARRAY [0..32] OF STRING(80) := ['c:\Temperatures.txt','c:\Log.txt']; (* TODO: Adjust
file paths*)END_VAR
```

**Programm-Code**

```
fbSmtplibFull(
sNetId:= '',
sSmtplibServer:= 'mail.company.com',
sUsername:= 'TestUser',
sPassword:= 'TestPwd',
sFrom:= 'TestUser@company.com',
sTo:= 'service@company.com',
sSubject:= 'Email from your Beckhoff PLC',
nContentType:= 2, (* 2 = HTML *)
nPriority:= 1, (* 1 = HIGH *)
sAttachments:= sFiles,
pMessage:= ADR(sMessage_HTML),
cbMessage:= SIZEOF(sMessage_HTML),
bExecute:= bStart,
bError=> bError,
bBusy=>bBusy,
nErrId=> nErrId);

IF NOT bBusy AND NOT bError AND bStart THEN
nState := 0;
bStart := FALSE;
END_IF
```

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Smtplib

**4.4 Anhang**

**4.4.1 Trouble-Shooting**

The following list provides basic help in case of errors and should be read *before* contacting our support department.

**1. Check if one of the PLC function blocks returns an error code:**

Please consult the list of error codes:

[SMTP Error codes](#) |▶ 34|

ADS Return codes

[Win Socket Error codes \[▶ 36\]](#)

## 2. Activate the logging option in the TcSmtplib.xml

The TcSmtplib.xml is located in \TwinCAT\Functions\TF6350-SMS-SMTP

Enable verbose logging by setting the EnableLogFile = 1 and restart TwinCAT.

```
<TcSmtplib> <!-- EnableLogFile: 0 (Disabled), 1 (Enabled) --> <EnableLogFile>1</
EnableLogFile> ... <Reconnects>5</Reconnects> </TcSmtplib>
```

The log file will be generate in \TwinCAT\Functions\TF6350-SMS-SMTP .

If the troubleshooting checklist does not help, please contact our support department and provide the following information:

### General system information

- What kind of hardware is being used on the computer running TF6350 SMS/SMTP?
  - Beckhoff IPC or Embedded PC: Which product number does the PC have?
  - Which Operating System image version is currently installed on that computer?

### Product-related system information

- Which version of TF6350 SMS/SMTP is being used?
- Which function blocks of the Tc2\_SMTP library are being used in the PLC program?
- Which SMTP Server is being used?
- Microsoft Exchange Server
- Axigen
- PostFix
- WebMail Provider (e.g. GMAIL, Hotmail, GMX)
- Please provide the SMTP logfile (see 2. of the troubleshooting list)
- Please provide an exact description of the environment in which the product TF6350 SMS/SMTP is being used
  - Where is the computer running TF6350 SMS/SMTP located?
  - Where is the SMTP-Server located? (Local network, Internet)
  - Which encryption is in use? (NON, TLS, SSL)
  - What are the IP settings of the Mail-Server and the computer running TF6350 SMS/SMTP? (IP address, subnet mask, ports)
  - In case that there is a firewall between both computers:

What kind of Firewall system is being used (please provide vendor information)?

## 4.4.2 Fehlercodes

Diese Liste gibt mögliche Fehlercodes für das Supplementprodukt TwinCAT SMTP Server an. Sollten Sie einen Fehlercode bekommen, welchen Sie nicht in der Liste finden, dann schauen Sie bitte in den [ADS Return Codes \[▶ 47\]](#) oder in der Liste der [WinSockErrorCodes \[▶ 36\]](#) nach.

Fehlercode (hex)	Fehlercode (dec)	Beschreibung
< 0x8000	< 32778	ADS return code
0x800A	32778	Not connected
0x800B	32779	Sender expected
0x800C	32780	Recipients expected
0x800D	32781	Send FROM command failed
0x800E	32782	Send DATA command failed
0x800F	32783	Send mail header failed
0x8010	32784	Send mail body failed
0x8011	32785	Send "end of mail indicator" failed
0x8012	32786	Send "RCPT" command failed
0x8013	32787	Server Response got no username request
0x8014	32788	Server Response got no password request
0x8015	32789	Unable to create socket connection
0x8016	32790	Authentication type not supported by smtp server
0x8017	32791	Wrong username or password
0x8018	32792	Not supported
0x8019	32793	Invalid hostname
0x801A	32794	Unable to send attachment
0x801B	32795	File not found
0x801C	32796	Invalid Version (New SMTP Server with old SMTP PLC library)
0x801D	32797	Unable to connect (Connection error => sometimes wrong port or wrong server)
0x801E	32798	Unable to create socket
0x801F	32799	WSA startup failed
0x8020	32800	Invalid hostname
0x8021	32801	Unexpected response from server
0x8022	32802	Error while receiving data
0x8023	32803	No supported authentication methods found
0x8024	32804	Invalid parameter
0x80A0	32928	Security interface not found
0x80A1	32929	Unable to call security interface
0x80A2	32930	Security initialization failed
0x80A4	32932	Unable to create credentials
0x80A5	32933	SSL-handshake failed
0x80A6	32934	Invalid server credentials
0x80A7	32935	Unable to verify server
0x80A8	32936	Unable to encrypt message
0x80A9	32937	Unable to decrypt message

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4000	PC or CX (x86)	Tc2_Smtp

### 4.4.3 Windows Socket Fehler Codes

Die folgende Liste beschreibt die möglichen Fehler Codes, die durch die WSAGetLastError Funktion zurückkommen könnten. Die Fehler sind in alphabetischer Reihenfolge, nach Fehler-Makros sortiert, aufgelistet. Einige Fehler Codes, die in Winsock2.h definiert sind, sind nicht zurückgekommen - diese sind nicht in der Liste enthalten.

<b>Rückgabewert</b>	<b>Beschreibung</b>
WSAEINTR10004	Interrupted function call.blocking operation was interrupted by a call to WSACancelBlockingCall.
WSAEACCES 10013	Permission denied.An attempt was made to access a socket in a way forbidden by its access permissions. An example is using a broadcast address for sendto without broadcast permission being set using setsockopt(SO_BROADCAST). Another possible reason for the WSAEACCES error is that when the bind function is called (on Windows NT 4 SP4 or later), another application, service, or kernel mode driver is bound to the same address with exclusive access. Such exclusive access is a new feature of Windows NT 4 SP4 and later, and is implemented by using the SO_EXCLUSIVEADDRUSE option.
WSAEFAULT 10014	Bad address.The system detected an invalid pointer address in attempting to use a pointer argument of a call. This error occurs if an application passes an invalid pointer value, or if the length of the buffer is too small. For instance, if the length of an argument, which is a sockaddr structure, is smaller than the sizeof(sockaddr).
WSAEINVAL 10022	Invalid argument.Some invalid argument was supplied (for example, specifying an invalid level to the setsockopt function). In some instances, it also refers to the current state of the socket—for instance, calling accept on a socket that is not listening.
WSAEMFILE 10024	Too many open files.Too many open sockets. Each implementation may have a maximum number of socket handles available, either globally, per process, or per thread.
WSAEWOULDBLOC K 10035	Resource temporarily unavailable.This error is returned from operations on nonblocking sockets that cannot be completed immediately, for example recv when no data is queued to be read from the socket. It is a nonfatal error, and the operation should be retried later. It is normal for WSAEWOLDBLOCK to be reported as the result from calling connect on a nonblocking SOCK_STREAM socket, since some time must elapse for the connection to be established.
WSAEINPROGRESS 10036	Operation now in progress.A blocking operation is currently executing. Windows Sockets only allows a single blocking operation—per- task or thread—to be outstanding, and if any other function call is made (whether or not it references that or any other socket) the function fails with the WSAEINPROGRESS error.
WSAEALREADY 10037	Operation already in progress.An operation was attempted on a nonblocking socket with an operation already in progress—that is, calling connect a second time on a nonblocking socket that is already connecting, or canceling an asynchronous request (WSAAsyncGetXbyY) that has already been canceled or completed.
WSAENOTSOCK 10038	Socket operation on nonsocket.An operation was attempted on something that is not a socket. Either the socket handle parameter did not reference a valid socket, or for select, a member of an fd_set was not valid.
WSAEDESTADDRRE Q 10039	Destination address required.A required address was omitted from an operation on a socket. For example, this error is returned if sendto is called with the remote address of ADDR_ANY.
WSAEMSGSIZE 10040	Message too long.A message sent on a datagram socket was larger than the internal message buffer or some other network limit, or the buffer used to receive a datagram was smaller than the datagram itself.
WSAEPROTOTYPE 10041	Protocol wrong type for socket.A protocol was specified in the socket function call that does not support the semantics of the socket type requested. For example, the ARPA Internet UDP protocol cannot be specified with a socket type of SOCK_STREAM.
WSAENOPROTOOP T 10042	Bad protocol option.An unknown, invalid or unsupported option or level was specified in a getsockopt or setsockopt call.
WSAEPROTONOSU PPORT 10043	Protocol not supported.The requested protocol has not been configured into the system, or no implementation for it exists. For example, a socket call requests a SOCK_DGRAM socket, but specifies a stream protocol.
WSAESOCKTNOSUP PORT 10044	Socket type not supported.The support for the specified socket type does not exist in this address family. For example, the optional type SOCK_RAW might be selected in a socket call, and the implementation does not support SOCK_RAW sockets at all.

Rückgabewert	Beschreibung
WSAEOPNOTSUPP 10045	Operation not supported. The attempted operation is not supported for the type of object referenced. Usually this occurs when a socket descriptor to a socket that cannot support this operation is trying to accept a connection on a datagram socket.
WSAEPFNOSUPPORT 10046	Protocol family not supported. The protocol family has not been configured into the system or no implementation for it exists. This message has a slightly different meaning from WSAEAFNOSUPPORT. However, it is interchangeable in most cases, and all Windows Sockets functions that return one of these messages also specify WSAEAFNOSUPPORT.
WSAEAFNOSUPPORT 10047	Address family not supported by protocol family. An address incompatible with the requested protocol was used. All sockets are created with an associated address family (that is, AF_INET for Internet Protocols) and a generic protocol type (that is, SOCK_STREAM). This error is returned if an incorrect protocol is explicitly requested in the socket call, or if an address of the wrong family is used for a socket, for example, in sendto.
WSAEADDRINUSE 10048	Address already in use. Typically, only one usage of each socket address (protocol/IP address/port) is permitted. This error occurs if an application attempts to bind a socket to an IP address/port that has already been used for an existing socket, or a socket that was not closed properly, or one that is still in the process of closing. For server applications that need to bind multiple sockets to the same port number, consider using setsockopt (SO_REUSEADDR). Client applications usually need not call bind at all— connect chooses an unused port automatically. When bind is called with a wildcard address (involving ADDR_ANY), a WSAEADDRINUSE error could be delayed until the specific address is committed. This could happen with a call to another function later, including connect, listen, WSAConnect, or WSAJoinLeaf.
WSAEADDRNOTAVAIL 10049	Cannot assign requested address. The requested address is not valid in its context. This normally results from an attempt to bind to an address that is not valid for the local computer. This can also result from connect, sendto, WSAConnect, WSAJoinLeaf, or WSASendTo when the remote address or port is not valid for a remote computer (for example, address or port 0).
WSAENETDOWN 10050	Network is down. A socket operation encountered a dead network. This could indicate a serious failure of the network system (that is, the protocol stack that the Windows Sockets DLL runs over), the network interface, or the local network itself.
WSAENETUNREACH 10051	Network is unreachable. A socket operation was attempted to an unreachable network. This usually means the local software knows no route to reach the remote host.
WSAENETRESET 10052	Network dropped connection on reset. The connection has been broken due to keep-alive activity detecting a failure while the operation was in progress. It can also be returned by setsockopt if an attempt is made to set SO_KEEPALIVE on a connection that has already failed.
WSAECONNABORTED 10053	Software caused connection abort. An established connection was aborted by the software in your host computer, possibly due to a data transmission time-out or protocol error.
WSAECONNRESET 10054	Connection reset by peer. An existing connection was forcibly closed by the remote host. This normally results if the peer application on the remote host is suddenly stopped, the host is rebooted, the host or remote network interface is disabled, or the remote host uses a hard close (see setsockopt for more information on the SO_LINGER option on the remote socket). This error may also result if a connection was broken due to keep-alive activity detecting a failure while one or more operations are in progress. Operations that were in progress fail with WSAENETRESET. Subsequent operations fail with WSAECONNRESET.
WSAENOBUFS 10055	No buffer space available. An operation on a socket could not be performed because the system lacked sufficient buffer space or because a queue was full.
WSAEISCONN 10056	Socket is already connected. A connect request was made on an already-connected socket. Some implementations also return this error if sendto is called on a connected SOCK_DGRAM socket (for SOCK_STREAM sockets, the to parameter in sendto is ignored) although other implementations treat this as a legal occurrence.



<b>Rückgabewert</b>	<b>Beschreibung</b>
WSAENOTCONN 10057	Socket is not connected.A request to send or receive data was disallowed because the socket is not connected and (when sending on a datagram socket using sendto) no address was supplied. Any other type of operation might also return this error—for example, setsockopt setting SO_KEEPALIVE if the connection has been reset.
WSAESHUTDOWN 10058	Cannot send after socket shutdown.A request to send or receive data was disallowed because the socket had already been shut down in that direction with a previous shutdown call. By calling shutdown a partial close of a socket is requested, which is a signal that sending or receiving, or both have been discontinued.
WSAETIMEDOUT 10060	Connection timed out.A connection attempt failed because the connected party did not properly respond after a period of time, or the established connection failed because the connected host has failed to respond.
WSAECONNREFUSED 10061	Connection refused.No connection could be made because the target computer actively refused it. This usually results from trying to connect to a service that is inactive on the foreign host—that is, one with no server application running.
WSAEHOSTDOWN 10064	Host is down.A socket operation failed because the destination host is down. A socket operation encountered a dead host. Networking activity on the local host has not been initiated. These conditions are more likely to be indicated by the error WSAETIMEDOUT.
WSAEHOSTUNREACH 10065	No route to host.A socket operation was attempted to an unreachable host. See WSAENETUNREACH.
WSAEPROCLIM 10067	Too many processes.A Windows Sockets implementation may have a limit on the number of applications that can use it simultaneously.WSASStartup may fail with this error if the limit has been reached.
WSASYSNOTREADY 10091	Network subsystem is unavailable.This error is returned by WSASStartup if the Windows Sockets implementation cannot function at this time because the underlying system it uses to provide network services is currently unavailable. Users should check: <ul style="list-style-type: none"> <li>• That the appropriate Windows Sockets DLL file is in the current path.</li> <li>• That they are not trying to use more than one Windows Sockets implementation simultaneously. If there is more than one Winsock DLL on your system, be sure the first one in the path is appropriate for the network subsystem currently loaded.</li> <li>• The Windows Sockets implementation documentation to be sure all necessary components are currently installed and configured correctly.</li> </ul>
WSAVERNOTSUPPORTED 10092	Winsock.dll version out of range.The current Windows Sockets implementation does not support the Windows Sockets specification version requested by the application. Check that no old Windows Sockets DLL files are being accessed.
WSANOTINITIALIZED 10093	Successful WSASStartup not yet performed.Either the application has not called WSASStartup or WSASStartup failed. The application may be accessing a socket that the current active task does not own (that is, trying to share a socket between tasks), or WSACleanup has been called too many times.
WSAEDISCON 10101	Graceful shutdown in progress.Returned by WSARcv and WSARcvFrom to indicate that the remote party has initiated a graceful shutdown sequence.
WSATYPE_NOT_FOUND 10109	Class type not found.The specified class was not found.
WSAHOST_NOT_FOUND 11001	Host not found.No such host is known. The name is not an official host name or alias, or it cannot be found in the database(s) being queried. This error may also be returned for protocol and service queries, and means that the specified name could not be found in the relevant database.
WSATRY_AGAIN 11002	Nonauthoritative host not found.This is usually a temporary error during host name resolution and means that the local server did not receive a response from an authoritative server. A retry at some time later may be successful.

Rückgabewert	Beschreibung
WSANO_RECOVERY 11003	This is a nonrecoverable error. This indicates that some sort of nonrecoverable error occurred during a database lookup. This may be because the database files (for example, BSD-compatible HOSTS, SERVICES, or PROTOCOLS files) could not be found, or a DNS request was returned by the server with a severe error.
WSANO_DATA 11004	Valid name, no data record of requested type. The requested name is valid and was found in the database, but it does not have the correct associated data being resolved for. The usual example for this is a host name-to-address translation attempt (using gethostbyname or WSAAsyncGetHostByName) which uses the DNS (Domain Name Server). An MX record is returned but no A record—indicating the host itself exists, but is not directly reachable.
WSA_INVALID_HANDLE OS dependent	Specified event object handle is invalid. An application attempts to use an event object, but the specified handle is not valid.
WSA_INVALID_PARAMETER OS dependent	One or more parameters are invalid. An application used a Windows Sockets function which directly maps to a Windows function. The Windows function is indicating a problem with one or more parameters.
WSA_IO_INCOMPLETE OS dependent	Overlapped I/O event object not in signaled state. The application has tried to determine the status of an overlapped operation which is not yet completed. Applications that use WSAGetOverlappedResult (with the fWait flag set to FALSE) in a polling mode to determine when an overlapped operation has completed, get this error code until the operation is complete.
WSA_IO_PENDING OS dependent	Overlapped operations will complete later. The application has initiated an overlapped operation that cannot be completed immediately. A completion indication will be given later when the operation has been completed.
WSA_NOT_ENOUGH_MEMORY OS dependent	Insufficient memory available. An application used a Windows Sockets function that directly maps to a Windows function. The Windows function is indicating a lack of required memory resources.
WSA_OPERATION_ABORTED OS dependent	Overlapped operation aborted. An overlapped operation was canceled due to the closure of the socket, or the execution of the SIO_FLUSH command in WSALocI.
WSA_INVALID_PROCEDURE OS dependent	Invalid procedure table from service provider. A service provider returned a bogus procedure table to Ws2_32.dll. (This is usually caused by one or more of the function pointers being null.)
WSA_INVALID_PROVIDER OS dependent	Invalid service provider version number. A service provider returned a version number other than 2.0.
WSA_PROVIDER_FAILED_INIT OS dependent	Unable to initialize a service provider. Either a service provider's DLL could not be loaded (LoadLibrary failed) or the provider's WSPStartup/NSPStartup function failed.
WSASYSCALL_FAILED OS dependent	System call failure. Generic error code, returned under various conditions. Returned when a system call that should never fail does fail. For example, if a call to WaitForMultipleEvents fails or one of the registry functions fails trying to manipulate the protocol/namespace catalogs. Returned when a provider does not return SUCCESS and does not provide an extended error code. Can indicate a service provider implementation error.

### Voraussetzungen

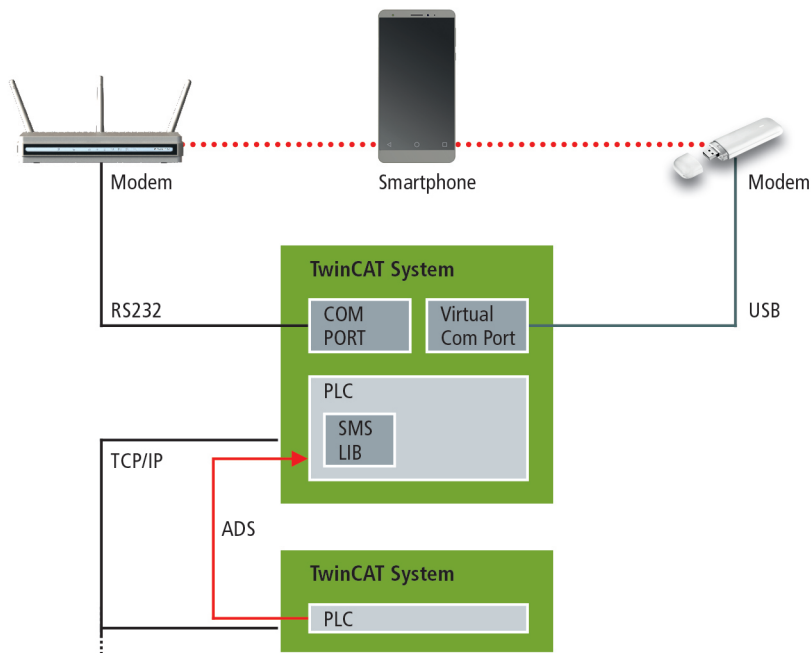
Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Smtp



## 5 TwinCAT SMS

Die TwinCAT SMS-Bibliotheken enthalten einen Baustein zum Verschicken von SMS-Nachrichten direkt aus der SPS. Die SMS-Bibliothek basiert auf der 'Serial Communication'-Bibliothek (die ebenfalls installiert wird). Damit ist es möglich, die serielle Schnittstelle des PCs und die serielle Klemme (ELxxxx, KL6xxx) anzusprechen.

Zusätzlich können auch über den ADS-Server von TF6340 TC3 Serial Communication USB-Geräte über virtuelle serielle Schnittstellen angesprochen werden. Wir empfehlen den LTE-USB-Stick [CU8210](#).

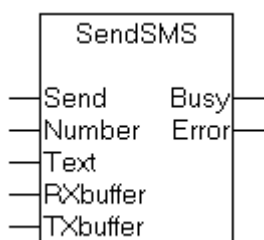


Des Weiteren ist es möglich, Alarm- oder Statusmeldungen von einem anderen TwinCAT System, dass über ein Netzwerk verbunden ist, auszugeben.

### 5.1 SPS API

#### 5.1.1 Funktionsbausteine

##### 5.1.1.1 FUNCTION\_BLOCK SendSMS



Mit dem Funktionsbaustein **SendSMS** wird eine SMS über ein angeschlossenes GSM Modem verschickt. Der Funktionsbaustein basiert auf der 'Serial Communication' Bibliothek.

Da der Baustein nur über die **ComBuffer** Struktur der 'Serial Communication' Bibliothek kommuniziert ist er instanzierbar und auf jede Art von serieller Schnittstelle anwendbar.

**VAR\_INPUT**

```
Send      : BOOL;
Number    : String;
Text      : String(160);
```

**Send:** Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

**Number:** Anzuwählende Telefonnummer im nationalen Format (z.B.: 0170123456)

**Text:** Die zu verschickende SMS Nachricht

**VAR\_OUTPUT**

```
Busy      : BOOL;
Error     : INT;
```

**Busy:** Bei der steigenden Flanke des Send Eingangs wird dieser Ausgang gesetzt und bleibt gesetzt, bis die SMS an das Modem übermittelt wurde oder ein Fehler aufgetreten ist.

**Error:** Sollte ein Fehler bei der Übertragung der SMS auftreten, wird der Busy Ausgang zurückgesetzt und am Error Ausgang liegt ein Fehlercode an.

Fehlercode	Bedeutung	Ursache
0	Kein Fehler	Die SMS wurde erfolgreich übertragen.
1	Keine Kommunikation zum Modem möglich.	Ist die Klemme korrekt konfiguriert? Wird die passende ComLib Bibliothek verwendet?
2	Modem meldet Fehler beim Konfigurieren.	Ist ein kompatibles GSM Modem angeschlossen?
3	Modem kann SMS nicht verschicken.	Ist die SIM Karte in Ordnung? Kann die Karte ohne Eingabe der PIN verwendet werden? Hat das Modem eine Verbindung zum Netz? Ist ein kompatibles Modem angeschlossen?
4	Kommunikationsfehler.	Ist die richtige Übertragungsgeschwindigkeit eingestellt?

**VAR\_IN\_OUT**

```
RXbuffer  : ComBuffer;
TXbuffer  : ComBuffer;
```

**RXbuffer:** Struktur für die Kommunikation mit der seriellen Schnittstelle. Ein schnittstellenspezifischer Baustein der 'Serial Communcation' Bibliothek füllt diesen Puffer mit den Daten der Schnittstelle.

**TXbuffer:** Struktur für die Kommunikation mit der seriellen Schnittstelle. Ein schnittstellenspezifischer Baustein der 'Serial Communcation' Bibliothek überträgt die Daten dieses Puffers an die Schnittstelle.

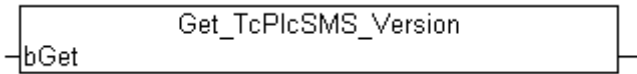
Diese Strukturen und ihre Verwendung sind in der Dokumentation der 'Serial Communcation' Bibliothek näher beschrieben. Der SendSMS Baustein wird dabei wie ein SendString oder ReceiveString Baustein angeschlossen.

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Sms

## 5.1.2 Funktionen

### 5.1.2.1 FUNCTION Get\_TcPlcSMS\_Version



Mit dieser Funktion können Versionsinformationen der SPS-Bibliothek ausgelesen werden.

**FUNCTION Get\_TcPlcSMS\_Version: STRING(20)**

#### VAR\_INPUT

bGet : BOOL;

**bGet:** Dieser Parameter hat für die Funktionsausführung keine Bedeutung und kann beliebig gesetzt werden..

#### Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Sms

## 5.1.3 Globale Konstanten

### 5.1.3.1 Bibliotheksversion

Alle Bibliotheken haben eine bestimmte Version. Diese Version ist u. a. im SPS-Bibliotheks-Repository zu sehen. Eine globale Konstante enthält die Information über die Bibliotheksversion:

Global\_Version

```
VAR_GLOBAL CONSTANT
    stLibVersion_Tc2_SMS_SMTP : ST_LibVersion;
END_VAR
```

ST\_LibVersion

Um zu sehen ob die Version die Sie haben auch die Version ist die Sie brauchen, benutzen Sie die Funktion F\_CmpLibVersion (definiert in Tc2\_System library).



#### Version vergleichen

Alle anderen Möglichkeiten Bibliotheksversionen zu vergleichen, die Sie von TwinCAT 2 kennen, sind veraltet!

## 5.2 Beispiele

### 5.2.1 Verschicken einer SMS über den Beckhoff 4G-Stick

Der installierte 4G-Stick wird als virtuelle serielle Schnittstelle verbunden, worüber sich die SPS per ADS verbinden kann. Durch den Aufruf des Funktionsbausteins „SendSMS“ wird die SMS an einen Empfänger versandt.

1. Tragen Sie in der Datenstruktur stSerialCfg ein.
2. Tragen Sie in der Stringvariable sNumber des Empfängers ein.
3. Versenden Sie eine SMS über eine steigende Flanke an bConnect.

### Programmvariablen

```
(*
COM_VCOM_SMS
Sends a SMS over a connected GSM Device especially over a virtual serial com port
*)
PROGRAM MAIN
VAR
(* function-block for sending a SMS *)
fbSendSMS : SendSMS;
sText : STRING := 'Please check machine #5, threshold is reached';
sSend : BOOL;
sNumber : STRING := '';
sBusy : BOOL;
sError : INT;

(* communication buffer between application and SerialLineControl *)
RxBuffer : ComBuffer;
TxBuffer : ComBuffer;

(* serial line control *)
fbLineCtrlAds : SerialLineControlADS;
bAdsError : BOOL;
nAdsErrorID : UDINT;
bConnect : BOOL;
sNetId : T_AmsNetId;
stSerialCfg : ComSerialConfig;
END_VAR
```

### Programm-Code

```
(*~~~~~ cyclic call of serial background comm. ~~~~~*)
fbLineCtrlAds(
Connect := bConnect, //bConnect has to be set to true, after the
stSerialCfg is checked
SerialCfg := stSerialCfg, //stSerialCfg has to be filled with your
Com-Settings (Com-Port, Baudrate, etc.)
NetId := sNetId,
Timeout := ,
TxBuffer := TxBuffer,
RxBuffer := RxBuffer,
Busy => ,
Error => bAdsError,
ErrorID => nAdsErrorID,
PortOpened =>
);
fbSendSMS(
Send:= sSend, //sSend has to be set to true, when the message is ready
Number:= sNumber, //sNumber is the number of the recipient
Text:= sText, //sText is the text you would like to send to the recipient
Busy=> sBusy,
Error=> sError,
RXbuffer:= RxBuffer,
TXbuffer:= TxBuffer);
```

### Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Sms, Tc2_SerialCom

## 5.3 Anhang

### 5.3.1 Fehlersuche

Falls das Verschicken einer SMS mit dem SendSMS Funktionsbaustein oder über den SMS COM Server nicht funktioniert, kann das verschiedene Gründe haben:

- keine Verbindung zum GSM Modem
- Verwendung eines nicht unterstützten GSM Modems
- falsche Rufnummer
- falsches GSM-Netz
- PIN erforderlich (die SIM Karte darf nicht mit einer PIN geschützt sein)
- nicht initialisierte serielle Klemme (KL6Init aufrufen)
- Fehlerhafter Aufruf des ADS Dienstes

Um diese Fehler zu finden gibt es verschiedene Hilfsmittel:

#### Log Datei verwenden

Mit dem SMS COM Server Konfigurator können Sie das Protokollieren in eine Log-Datei aktivieren. Danach werden alle verschickten Nachrichten und Fehler in die Datei TcSmsSrvCfg.xml geschrieben.

#### NT Event Log überprüfen

Fehler bei der Versendung von Nachrichten werden zusätzlich immer im NT Event Log protokolliert. Das Event Log erreichen Sie über das TwinCAT Icon in der Taskleiste.

#### ADS Fehlermeldungen auswerten

Falls der Aufruf einer ADS Funktion fehlschlägt, wird der Fehler im Rückgabewert der Funktion codiert. Eine Liste dieser Fehlercodes finden Sie unter [ADS Return Codes](#) [► 47].

#### Konfiguration der Klemme

Die serielle Klemme kann auf verschiedene Arten konfiguriert werden. Unterschiedlich konfigurierte Klemmen werden zum Teil auch unterschiedlich im Prozessabbild abgebildet (3 Byte / 5 Byte Klemme, Advanced / Standard). Dabei ist zu beachten, dass die 'Serial Communication' Bibliothek zu der Konfiguration der Klemme passen muss. Siehe auch Dokumentation zur KL6xxx und Dokumentation der ComLib:

Wichtig ist auch die Übertragungsgeschwindigkeit der Klemme auf das verwendete Modem abzustimmen.

#### Verschicken einer Test SMS

Um herauszufinden, ob der Fehler im ADS Aufruf oder in der Konfiguration des SMS COM Servers liegt, können Sie mit dem Visual Basic Beispiel Programm einfach eine Test SMS verschicken.

#### Verschicken einer Test SMS mit einem Handy

Nummer einzugeben.

#### Netzauswahl beim Westermo GS-01

Für die verschiedenen Netze in Europa und den USA gibt es verschiedene Varianten des GS-01. Die Lampe an der Vorderseite des Modems zeigt an, ob ein Netz verfügbar ist. Wenn die Lampe blinkt, hat das Modem Verbindung zu einem Netz. Wenn die Lampe dauernd leuchtet, sollte das Westermo Handbuch zur Fehlersuche verwendet werden.

**Voraussetzungen**

<b>Entwicklungsumgebung</b>	<b>Zielplattform</b>	<b>Einzubindende SPS-Bibliotheken</b>
TwinCAT v3.0.0	PC or CX (x86)	Tc2_

# 6 Anhang

## 6.1 ADS Return Codes

Gruppierung der Fehlercodes:

Globale Fehlercodes: 0x0000 [[▶ 47](#)]... (0x9811\_0000 ...)

Router Fehlercodes: 0x0500 [[▶ 47](#)]... (0x9811\_0500 ...)

Allgemeine ADS Fehler: 0x0700 [[▶ 48](#)]... (0x9811\_0700 ...)

RTime Fehlercodes: 0x1000 [[▶ 50](#)]... (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_INVALIDAMSLENGTH	Ungültige AMS-Länge.
0xF	15	0x9811000F	ERR_INVALIDAMSNETID	Ungültige AMS Net ID.
0x10	16	0x98110010	ERR_LOWINSTLEVEL	Installations-Level ist zu niedrig –TwinCAT 2 Lizenzfehler.
0x11	17	0x98110011	ERR_NODEBUGINTAVAILABLE	Kein Debugging verfügbar.
0x12	18	0x98110012	ERR_PORTDISABLED	Port deaktiviert – TwinCAT System Service nicht gestartet.
0x13	19	0x98110013	ERR_PORTALREADYCONNECTED	Port bereits verbunden.
0x14	20	0x98110014	ERR_AMSSYNC_W32ERROR	AMS Sync Win32 Fehler.
0x15	21	0x98110015	ERR_AMSSYNC_TIMEOUT	AMS Sync Timeout.
0x16	22	0x98110016	ERR_AMSSYNC_AMSERROR	AMS Sync Fehler.
0x17	23	0x98110017	ERR_AMSSYNC_NOINDEXINMAP	Keine Index-Map für AMS Sync vorhanden.
0x18	24	0x98110018	ERR_INVALIDAMSSPORT	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	Lockierter 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.
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.
0x728	1832	0x98110728	ADSERR_DEVICE_LICENSESYSTEMID	Lizenzproblem: System-ID ist ungültig.
0x729	1833	0x98110729	ADSERR_DEVICE_LICENSENOTIMELIMIT	Lizenz nicht zeitlich begrenzt.
0x72A	1834	0x9811072A	ADSERR_DEVICE_LICENSEFUTUREISSUE	Lizenzproblem: Zeitpunkt in der Zukunft.
0x72B	1835	0x9811072B	ADSERR_DEVICE_LICENSETIMETOLONG	Lizenz-Zeitraum zu lang.
0x72C	1836	0x9811072C	ADSERR_DEVICE_EXCEPTION	Exception beim Systemstart.
0x72D	1837	0x9811072D	ADSERR_DEVICE_LICENSEDUPLICATED	Lizenz-Datei zweimal gelesen.
0x72E	1838	0x9811072E	ADSERR_DEVICE_SIGNATUREINVALID	Ungültige Signatur.
0x72F	1839	0x9811072F	ADSERR_DEVICE_CERTIFICATEINVALID	Zertifikat ungültig.
0x730	1840	0x98110730	ADSERR_DEVICE_LICENSEOEMNOTFOUND	Public Key vom OEM nicht bekannt.
0x731	1841	0x98110731	ADSERR_DEVICE_LICENSERESTRICTED	Lizenz nicht gültig für diese System.ID.
0x732	1842	0x98110732	ADSERR_DEVICE_LICENSEDEMODENIED	Demo-Lizenz untersagt.
0x733	1843	0x98110733	ADSERR_DEVICE_INVALIDFNCID	Funktions-ID ungültig.
0x734	1844	0x98110734	ADSERR_DEVICE_OUTOFRANGE	Außerhalb des gültigen Bereiches.
0x735	1845	0x98110735	ADSERR_DEVICE_INVALIDALIGNMENT	Ungültiges Alignment.

Hex	Dec	HRESULT	Name	Beschreibung
0x736	1846	0x98110736	ADSERR_DEVICE_LICENSEPLATFORM	Ungültiger Plattform Level.
0x737	1847	0x98110737	ADSERR_DEVICE_FORWARD_PL	Kontext – Weiterleitung zum Passiv-Level.
0x738	1848	0x98110738	ADSERR_DEVICE_FORWARD_DL	Kontext – Weiterleitung zum Dispatch-Level.
0x739	1849	0x98110739	ADSERR_DEVICE_FORWARD_RT	Kontext – Weiterleitung zur Echtzeit.
0x740	1856	0x98110740	ADSERR_CLIENT_ERROR	Clientfehler.
0x741	1857	0x98110741	ADSERR_CLIENT_INVALIDPARM	Dienst enthält einen ungültigen Parameter.
0x742	1858	0x98110742	ADSERR_CLIENT_LISTEMPTY	Polling-Liste ist leer.
0x743	1859	0x98110743	ADSERR_CLIENT_VARUSED	Var-Verbindung bereits im Einsatz.
0x744	1860	0x98110744	ADSERR_CLIENT_DUPLINVOKEID	Die aufgerufene ID ist bereits in Benutzung.
0x745	1861	0x98110745	ADSERR_CLIENT_SYNCTIMEOUT	Timeout ist aufgetreten – Die Gegenstelle antwortet nicht im vorgegebenen ADS Timeout. Die Routeneinstellung der Gegenstelle kann falsch konfiguriert sein.
0x746	1862	0x98110746	ADSERR_CLIENT_W32ERROR	Fehler im Win32 Subsystem.
0x747	1863	0x98110747	ADSERR_CLIENT_TIMEOUTINVALID	Ungültiger Client Timeout-Wert.
0x748	1864	0x98110748	ADSERR_CLIENT_PORTNOTOPEN	Port nicht geöffnet.
0x749	1865	0x98110749	ADSERR_CLIENT_NOAMSADDR	Keine AMS Adresse.
0x750	1872	0x98110750	ADSERR_CLIENT_SYNCINTERNAL	Interner Fehler in Ads-Sync.
0x751	1873	0x98110751	ADSERR_CLIENT_ADDHASH	Überlauf der Hash-Tabelle.
0x752	1874	0x98110752	ADSERR_CLIENT_REMOVEHASH	Schlüssel in der Tabelle nicht gefunden.
0x753	1875	0x98110753	ADSERR_CLIENT_NOMORESVM	Keine Symbole im Cache.
0x754	1876	0x98110754	ADSERR_CLIENT_SYNCRESINVALID	Ungültige Antwort erhalten.
0x755	1877	0x98110755	ADSERR_CLIENT_SYNCPORTLOCKED	Sync Port ist verriegelt.

### 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_IRQLNOTLESSOREQUAL	Aufruf einer Service-Funktion im falschen Kontext
0x1017	4119	0x98111017	RTERR_VMXNOTSUPPORTED	Intel VT-x Erweiterung wird nicht unterstützt.
0x1018	4120	0x98111018	RTERR_VMXDISABLED	Intel VT-x Erweiterung ist nicht aktiviert im BIOS.
0x1019	4121	0x98111019	RTERR_VMXCONTROLSMISSING	Fehlende Funktion in Intel VT-x Erweiterung.
0x101A	4122	0x9811101A	RTERR_VMXENABLEFAILS	Aktivieren von Intel VT-x schlägt fehl.

### Spezifische positive HRESULT Return Codes:

HRESULT	Name	Beschreibung
0x0000_0000	S_OK	Kein Fehler.
0x0000_0001	S_FALSE	Kein Fehler. Bsp.: erfolgreiche Abarbeitung, bei der jedoch ein negatives oder unvollständiges Ergebnis erzielt wurde.
0x0000_0203	S_PENDING	Kein Fehler. Bsp.: erfolgreiche Abarbeitung, bei der jedoch noch kein Ergebnis vorliegt.
0x0000_0256	S_WATCHDOG_TIMEOUT	Kein Fehler. Bsp.: erfolgreiche Abarbeitung, bei der jedoch eine Zeitüberschreitung eintrat.

**TCP Winsock-Fehlercodes**

Hex	Dec	Name	Beschreibung
0x274C	10060	WSAETIMEDOUT	Verbindungs Timeout aufgetreten - Fehler beim Herstellen der Verbindung, da die Gegenstelle nach einer bestimmten Zeitspanne nicht ordnungsgemäß reagiert hat, oder die hergestellte Verbindung konnte nicht aufrecht erhalten werden, da der verbundene Host nicht reagiert hat.
0x274D	10061	WSAECONNREFUSED	Verbindung abgelehnt - Es konnte keine Verbindung hergestellt werden, da der Zielcomputer dies explizit abgelehnt hat. Dieser Fehler resultiert normalerweise aus dem Versuch, eine Verbindung mit einem Dienst herzustellen, der auf dem fremden Host inaktiv ist—das heißt, einem Dienst, für den keine Serveranwendung ausgeführt wird.
0x2751	10065	WSAEHOSTUNREACH	Keine Route zum Host - Ein Socketvorgang bezog sich auf einen nicht verfügbaren Host.
Weitere Winsock-Fehlercodes: Win32-Fehlercodes			

## 6.2 Support und Service

Beckhoff und seine weltweiten Partnerfirmen bieten einen umfassenden Support und Service, der eine schnelle und kompetente Unterstützung bei allen Fragen zu Beckhoff Produkten und Systemlösungen zur Verfügung stellt.

### Beckhoff Niederlassungen und Vertretungen

Wenden Sie sich bitte an Ihre Beckhoff Niederlassung oder Ihre Vertretung für den lokalen Support und Service zu Beckhoff Produkten!

Die Adressen der weltweiten Beckhoff Niederlassungen und Vertretungen entnehmen Sie bitte unseren Internetseiten: <https://www.beckhoff.de>

Dort finden Sie auch weitere Dokumentationen zu Beckhoff Komponenten.

### Beckhoff Support

Der Support bietet Ihnen einen umfangreichen technischen Support, der Sie nicht nur bei dem Einsatz einzelner Beckhoff Produkte, sondern auch bei weiteren umfassenden Dienstleistungen unterstützt:

- Support
- Planung, Programmierung und Inbetriebnahme komplexer Automatisierungssysteme
- umfangreiches Schulungsprogramm für Beckhoff Systemkomponenten

Hotline: +49(0)5246 963 157  
 Fax: +49(0)5246 963 9157  
 E-Mail: [support@beckhoff.com](mailto:support@beckhoff.com)

### Beckhoff Service

Das Beckhoff Service-Center unterstützt Sie rund um den After-Sales-Service:

- Vor-Ort-Service
- Reparaturservice
- Ersatzteilservice
- Hotline-Service

Hotline: +49(0)5246 963 460  
 Fax: +49(0)5246 963 479  
 E-Mail: [service@beckhoff.com](mailto:service@beckhoff.com)

### Beckhoff Firmenzentrale

Beckhoff Automation GmbH & Co. KG

Hülshorstweg 20  
33415 Verl  
Deutschland

Telefon: +49(0)5246 963 0  
Fax: +49(0)5246 963 198  
E-Mail: [info@beckhoff.com](mailto:info@beckhoff.com)  
Internet: <https://www.beckhoff.de>



Mehr Informationen:  
**[www.beckhoff.de/tf6350](http://www.beckhoff.de/tf6350)**

Beckhoff Automation GmbH & Co. KG  
Hülshorstweg 20  
33415 Verl  
Deutschland  
Telefon: +49 5246 9630  
[info@beckhoff.de](mailto:info@beckhoff.de)  
[www.beckhoff.de](http://www.beckhoff.de)

