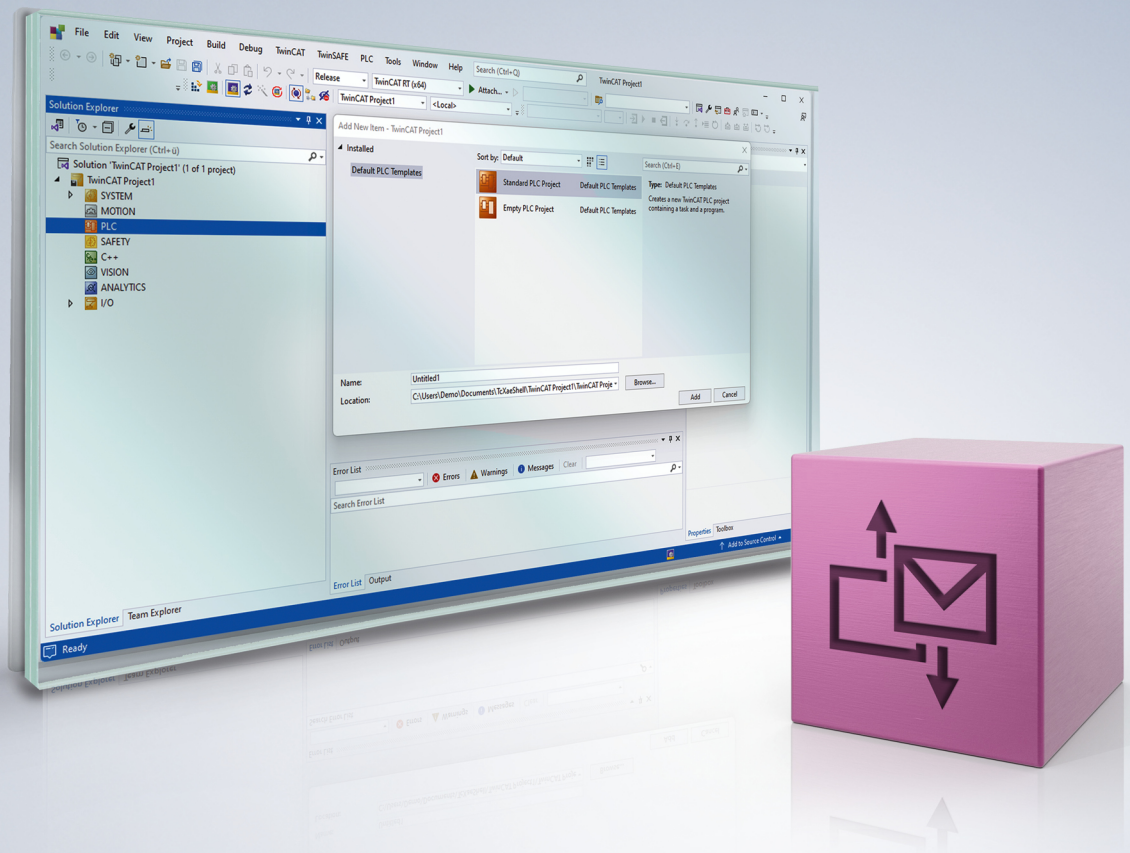
BECKHOFF New Automation Technology

Manual | EN

TF6350

TwinCAT 3 | SMS/SMTP



2025-09-18 | Version: 1.5.2

Table of contents

1 Foreword 5
1.1 Notes on the documentation 5
1.2 For your safety 5
1.3 Notes on information security 7
2 Overview 8
3 Installation 9
3.1 System requirements 9
3.2 Installation 9
3.3 Installation Windows CE 12
3.4 Licensing 14
4 TwinCAT SMTP 17
4.1 Configuration 17
4.2 PLC API 19
4.2.1 Function blocks 20
4.3 Samples 33
4.3.1 Mail dispatch 33
4.3.2 SmtpFull sample with features 33
4.4 Appendix 34
4.4.1 Trouble-Shooting 34
4.4.2 Error Codes 35
4.4.3 Windows Socket Error Codes 36
5 TwinCAT SMS 41
5.1 PLC API 42
5.1.1 Function blocks 42
5.1.2 Functions 43
5.1.3 Global constants 44
5.2 Samples 44
5.2.1 Sending an SMS via the Beckhoff 4G stick 44
5.3 Appendix 45
5.3.1 Fault Finding 45
6 Appendix 47
6.1 ADS Return Codes 47
6.2 Support and Service 51

1 Foreword

1.1 Notes on the documentation

This description is intended exclusively for trained specialists in control and automation technology who are familiar with the applicable national standards.
The documentation and the following notes and explanations must be complied with when installing and commissioning the components.
The trained specialists must always use the current valid documentation.

The trained specialists must ensure that the application and use of the products described is in line with all safety requirements, including all relevant laws, regulations, guidelines, and standards.

Disclaimer

The documentation has been compiled with care. The products described are, however, constantly under development.
We reserve the right to revise and change the documentation at any time and without notice.
Claims to modify products that have already been supplied may not be made on the basis of the data, diagrams, and descriptions in this documentation.

Trademarks

Beckhoff®, ATRO®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, MX-System®, Safety over EtherCAT®, TC/BSD®, TwinCAT®, TwinCAT/BSD®, TwinSAFE®, XFC®, XPlanar®, and XTS® are registered and licensed trademarks of Beckhoff Automation GmbH.
If third parties make use of the designations or trademarks contained in this publication for their own purposes, this could infringe upon the rights of the owners of the said designations.



EtherCAT® is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.

Copyright

© Beckhoff Automation GmbH & Co. KG, Germany.
The distribution and reproduction of this document, as well as the use and communication of its contents without express authorization, are prohibited.
Offenders will be held liable for the payment of damages. All rights reserved in the event that a patent, utility model, or design are registered.

Third-party trademarks

Trademarks of third parties may be used in this documentation. You can find the trademark notices here: https://www.beckhoff.com/trademarks.

1.2 For your safety

Safety regulations

Read the following explanations for your safety.
Always observe and follow product-specific safety instructions, which you may find at the appropriate places in this document.

Exclusion of liability

All the components are supplied in particular hardware and software configurations which are appropriate for the application. Modifications to hardware or software configurations other than those described in the documentation are not permitted, and nullify the liability of Beckhoff Automation GmbH & Co. KG.

Personnel qualification

This description is only intended for trained specialists in control, automation, and drive technology who are familiar with the applicable national standards.

Signal words

The signal words used in the documentation are classified below. In order to prevent injury and damage to persons and property, read and follow the safety and warning notices.

Personal injury warnings

⚠ DANGER

Hazard with high risk of death or serious injury.

⚠ WARNING

Hazard with medium risk of death or serious injury.

⚠ CAUTION

There is a low-risk hazard that could result in medium or minor injury.

Warning of damage to property or environment

NOTICE

The environment, equipment, or data may be damaged.

Information on handling the product

This information includes, for example:
recommendations for action, assistance or further information on the product.

1.3 Notes on information security

The products of Beckhoff Automation GmbH & Co. KG (Beckhoff), insofar as they can be accessed online, are equipped with security functions that support the secure operation of plants, systems, machines and networks. Despite the security functions, the creation, implementation and constant updating of a holistic security concept for the operation are necessary to protect the respective plant, system, machine and networks against cyber threats. The products sold by Beckhoff are only part of the overall security concept. The customer is responsible for preventing unauthorized access by third parties to its equipment, systems, machines and networks. The latter should be connected to the corporate network or the Internet only if appropriate protective measures have been set up.

In addition, the recommendations from Beckhoff regarding appropriate protective measures should be observed. Further information regarding information security and industrial security can be found in our https://www.beckhoff.com/secguide.

Beckhoff products and solutions undergo continuous further development. This also applies to security functions. In light of this continuous further development, Beckhoff expressly recommends that the products are kept up to date at all times and that updates are installed for the products once they have been made available. Using outdated or unsupported product versions can increase the risk of cyber threats.

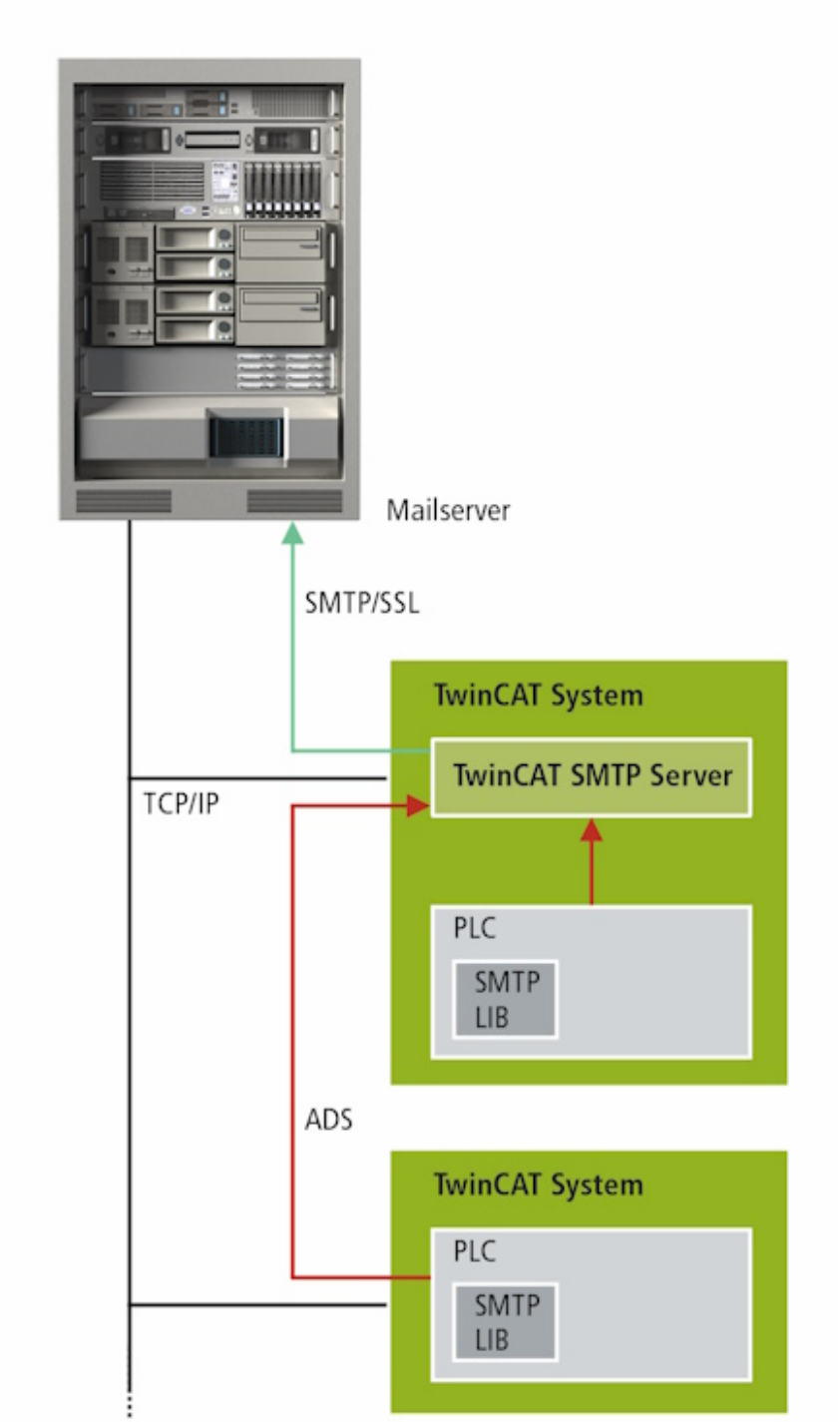
To stay informed about information security for Beckhoff products, subscribe to the RSS feed at https://www.beckhoff.com/secinfo.

2 Overview

This functionality enables alarm and status messages to be sent directly from TwinCAT. With this function, emails or SMS can be sent according to self-defined conditions.

SMTP Server:
The TwinCAT SMTP Server is used to send emails with TwinCAT via ADS.

TwinCAT SMTP Overview [▶ 17]



SMS library:

The TwinCAT SMS library is used to send SMS messages from TwinCAT via GSM modem.

TwinCAT SMS Overview [▶ 41]

3 Installation

3.1 System requirements

Technical Data	TF6350 TwinCAT 3 SMS/SMTP
Target system	Windows CE, Windows 10, Windows 11 IPC or CX, (x86)
Min. TwinCAT version	3.1.4000
Min. TwinCAT level	TC1200 | TwinCAT 3 PLC

Requirements

Development Environment	Target system	PLC libraries to include
TwinCAT v3.1.4000	PC or CX (x86)	Tc2_Smtp, Tc2_Sms

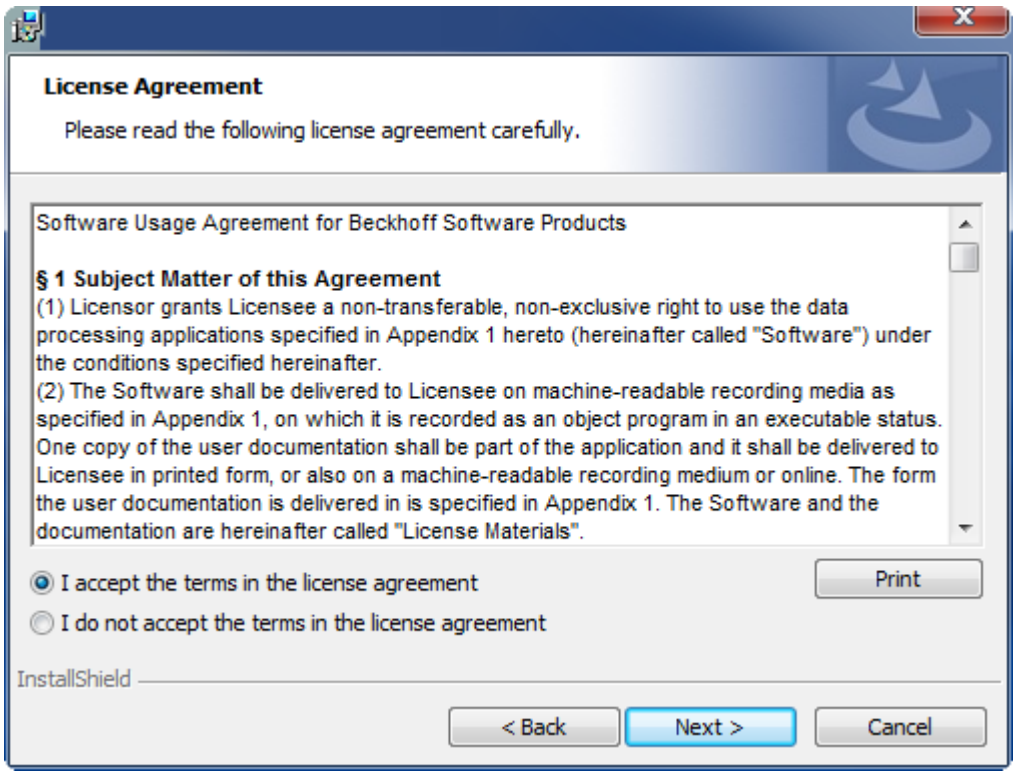
3.2 Installation

Setup installation (TwinCAT 3.1 Build 4024)

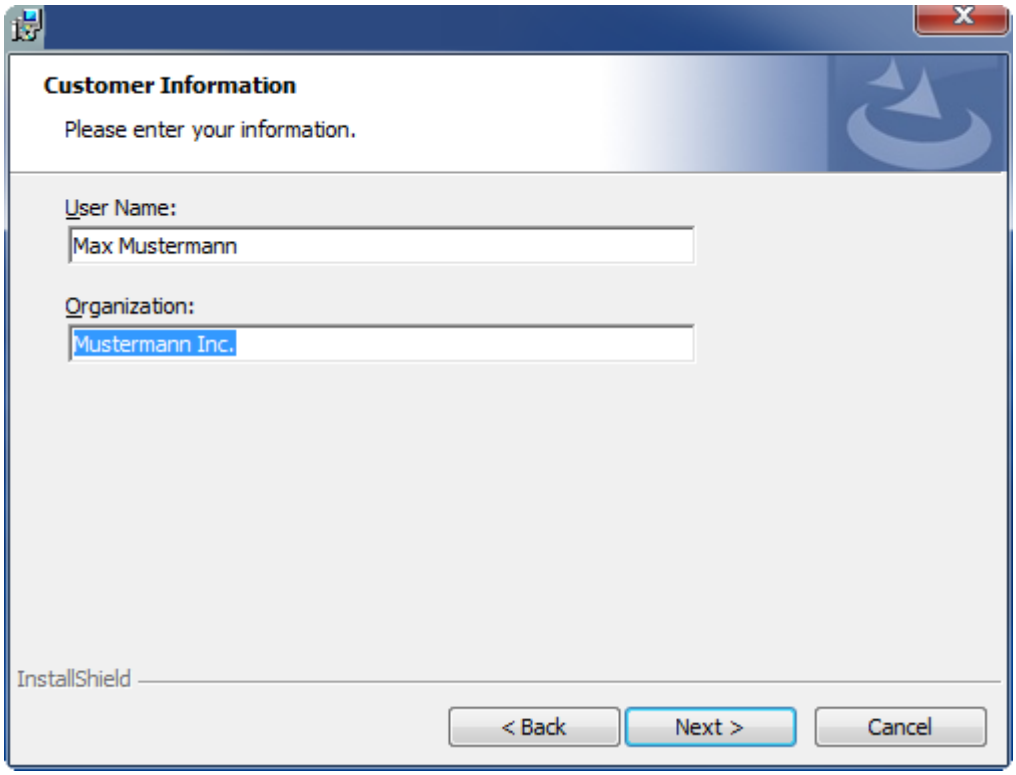
The following section describes how to install the TwinCAT 3 function for Windows-based operating systems.

✓ The TwinCAT 3 function setup file was downloaded from the Beckhoff website.

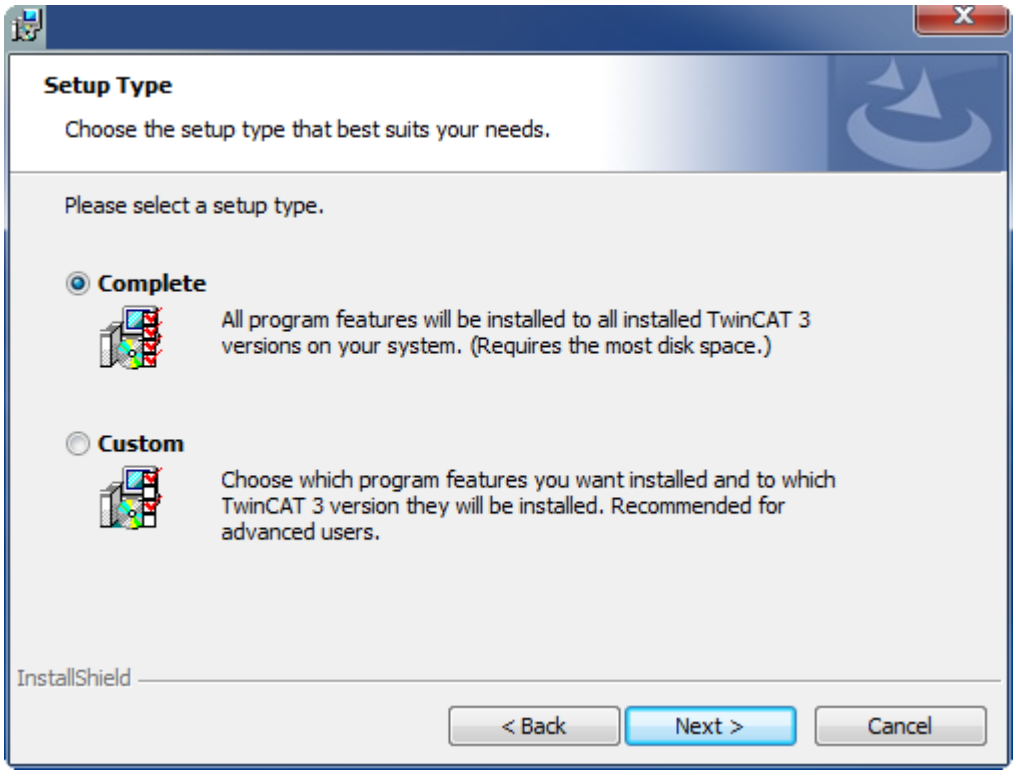
1. Run the setup file as administrator. To do this, select the **Run As Admin** command in the context menu of the file.
   ⇨ The installation dialog opens.
2. Accept the end user licensing agreement and click **Next**.



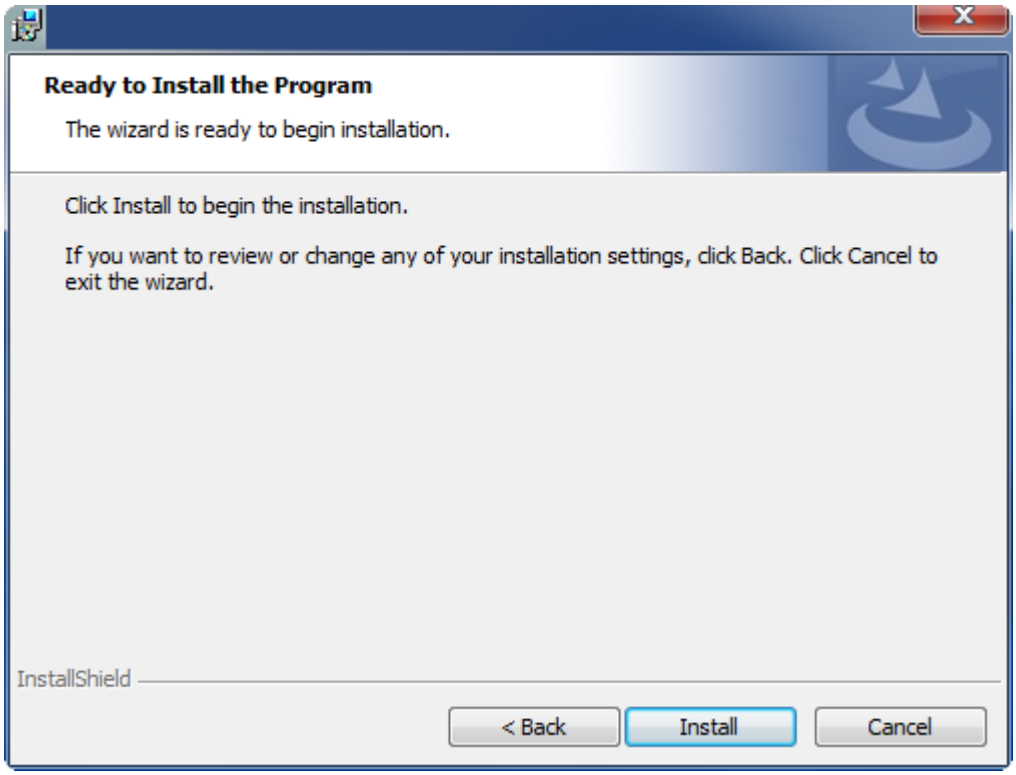
3. Enter your user data.



4. If you want to install the full version of the TwinCAT 3 function, select **Complete** as the installation type. If you want to install the TwinCAT 3 function components separately, select **Custom**.

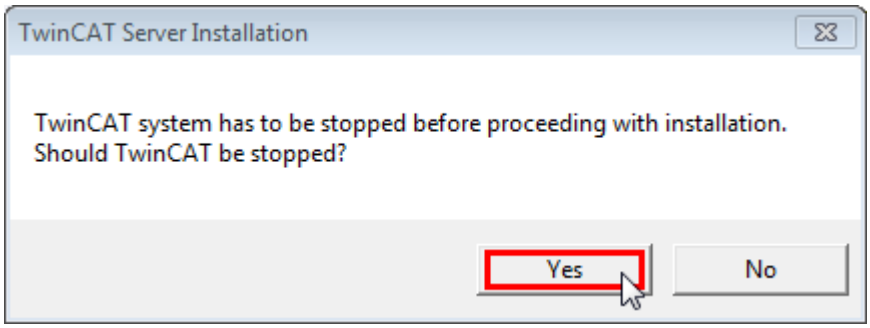


5. Click **Next**, then **Install** to start the installation.

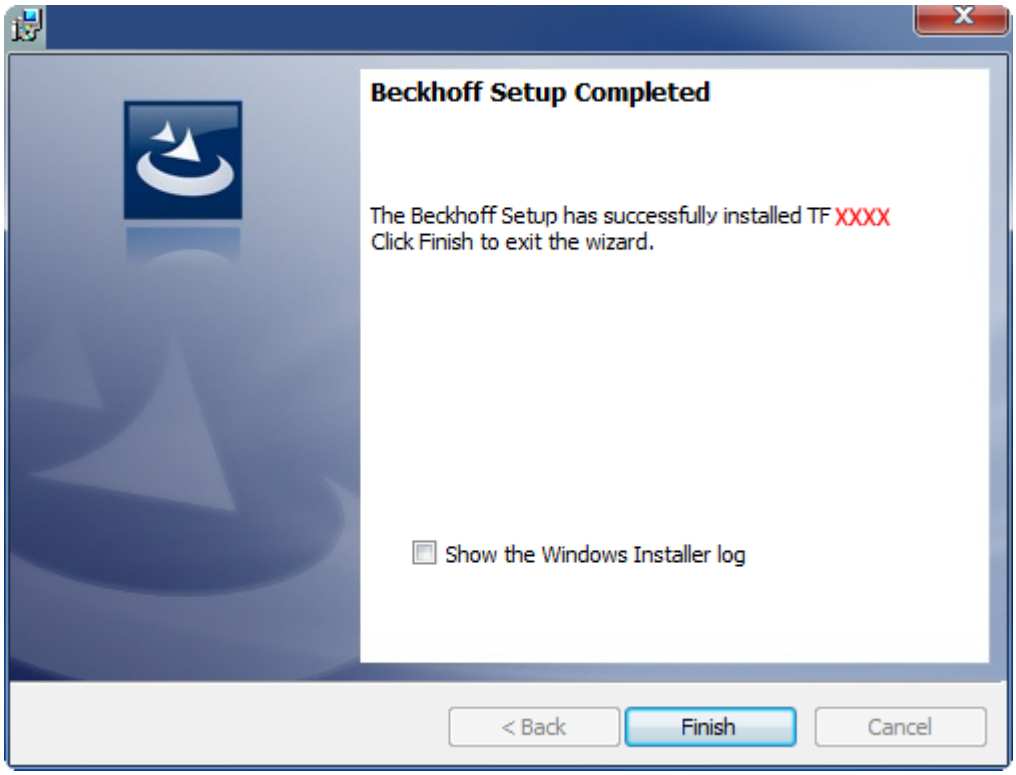


⇨ A dialog box informs you that the TwinCAT system must be stopped to proceed with the installation.

6. Confirm the dialog with **Yes.**



7. Click **Finish** to exit the setup.



⇨ The TwinCAT 3 function has been installed successfully.

3.3 Installation Windows CE

This part of the documentation describes how the TwinCAT 3 Function TF6350 SMS/SMTP can be installed on a Beckhoff Embedded Controller with Windows CE.

The setup process consists of the following steps:

- Downloading the setup file
- Installation on a host computer
- Transfer of the executable file to the Windows CE device
- Installing the software
- Software upgrade

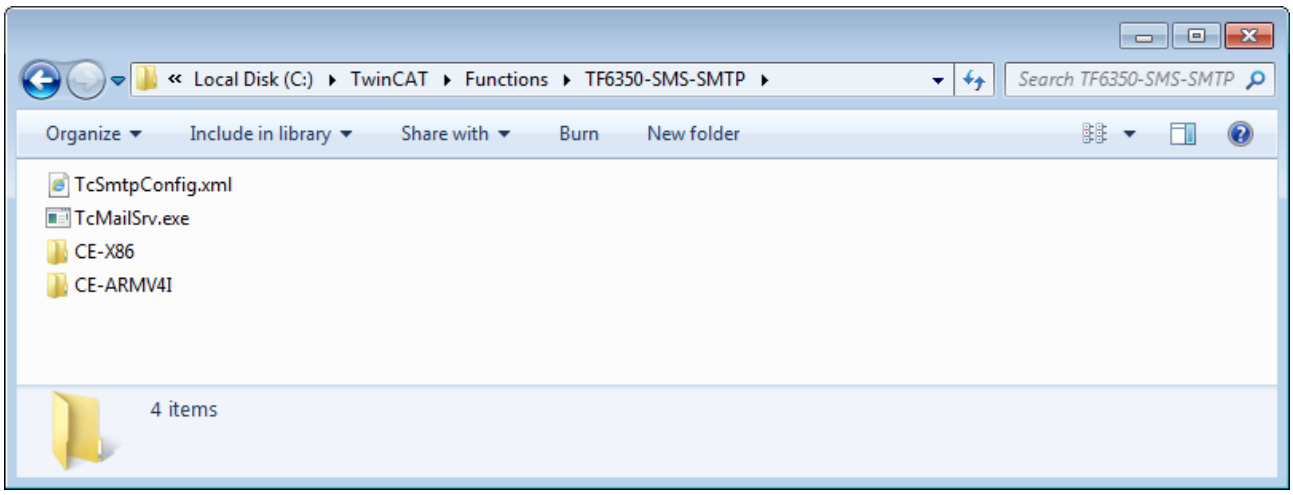
Downloading the setup file

The executable file for Windows CE is part of the TF6350 SMS/SMTP setup. Therefore, you only need to obtain the corresponding setup from www.beckhoff.com, which automatically contains all versions for Windows CE (x86 and Arm®).

Installation on a host computer

After the installation, the installation folder contains three directories - one directory per hardware platform:

- **CE ARM:** Arm®-based embedded controllers running under Windows CE, e.g. CX8090, CX9020.
- **CE-X86:** X86-based embedded controllers that run under Windows CE, e.g. CX50xx. CX20x0.
- **Win32:** Embedded controllers running under Windows Embedded Standard, for example.



The CE ARM and CE X86 directories contain the CAB files of the TF6350 function for Windows CE in relation to the respective hardware platform of your Windows CE device. These executable files must be copied to the Windows CE device. You will find out how this works in the next chapter.

Transfer of the executable file to the Windows CE device

Transfer of the executable file to the Windows CE device.

There are several ways to transfer files, via:

- network shares
- the integrated FTP server
- ActiveSync
- CF/SD cards

For further information, please refer to the Windows CE section in the Beckhoff Information System.

Installing the software

After the file has been transferred to the Windows CE device, you must execute the file there. You can confirm the installation dialog with **OK**. After the installation has been completed, restart the Windows CE device.

After the device has been restarted, the executable files (client and server) are automatically loaded in the background and are now available.

The software is installed in the following directory on the CE device: **\Hard Disk\TwinCAT\Functions\ TF6350-SMS-SMTP**

Software upgrade

If you already have an older TF6350 SMS/SMTP version installed on your Windows CE device, you must perform the following steps on the Windows CE device to upgrade to a newer version:

- Open the CE Explorer by clicking Start > Run and entering "explorer".
- Navigate to \Hard Disk\TwinCAT\Functions\TF6350-SMS-SMTP\Server.
- Rename the file "TcMail.Srv.exe".
- Restart the Windows CE device.
- Transfer the new CAB file to the Windows CE device.
- Install the new CAB file.
- Delete the old (re-named) files.
- Restart the Windows CE device.

Once the restart has been carried out, the new version is active.

The next step after a successful installation is the licensing of the TwinCAT 3 function.

3.4 Licensing

The TwinCAT 3 function can be activated as a full version or as a 7-day test version. Both license types can be activated via the TwinCAT 3 development environment (XAE).

Licensing the full version of a TwinCAT 3 Function

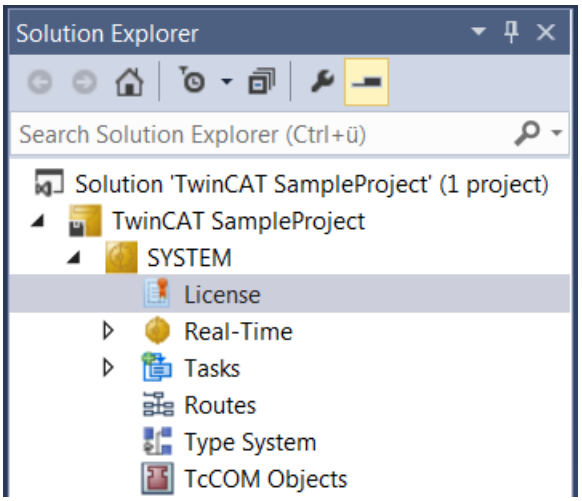
A description of the procedure to license a full version can be found in the Beckhoff Information System in the documentation "TwinCAT 3 Licensing".

Licensing the 7-day test version of a TwinCAT 3 Function



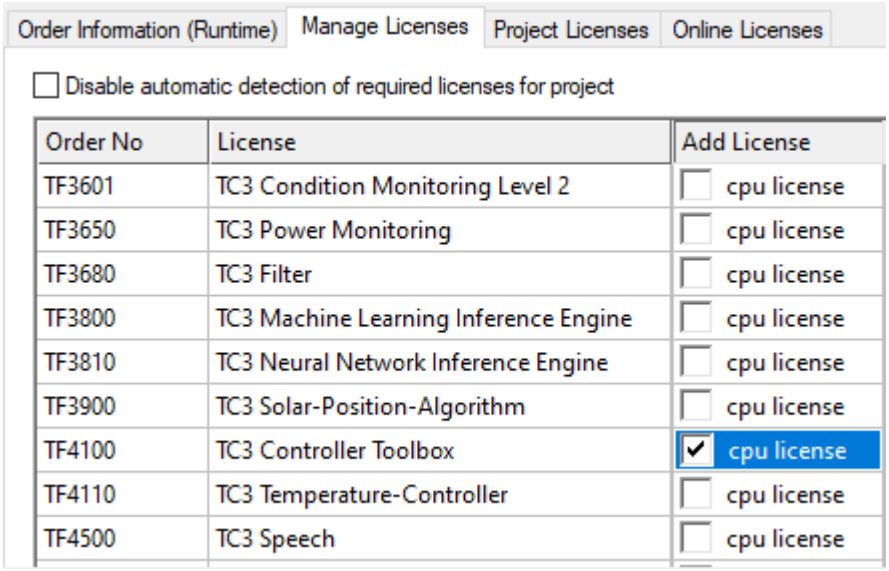
A 7-day test version cannot be enabled for a TwinCAT 3 license dongle.

1. Start the TwinCAT 3 development environment (XAE).
2. Open an existing TwinCAT 3 project or create a new project.
3. If you want to activate the license for a remote device, set the desired target system. To do this, select the target system from the **Choose Target System** drop-down list in the toolbar.
   ⇨ The licensing settings always refer to the selected target system. When the project is activated on the target system, the corresponding TwinCAT 3 licenses are automatically copied to this system.
4. In the **Solution Explorer**, double-click **License** in the **SYSTEM** subtree.

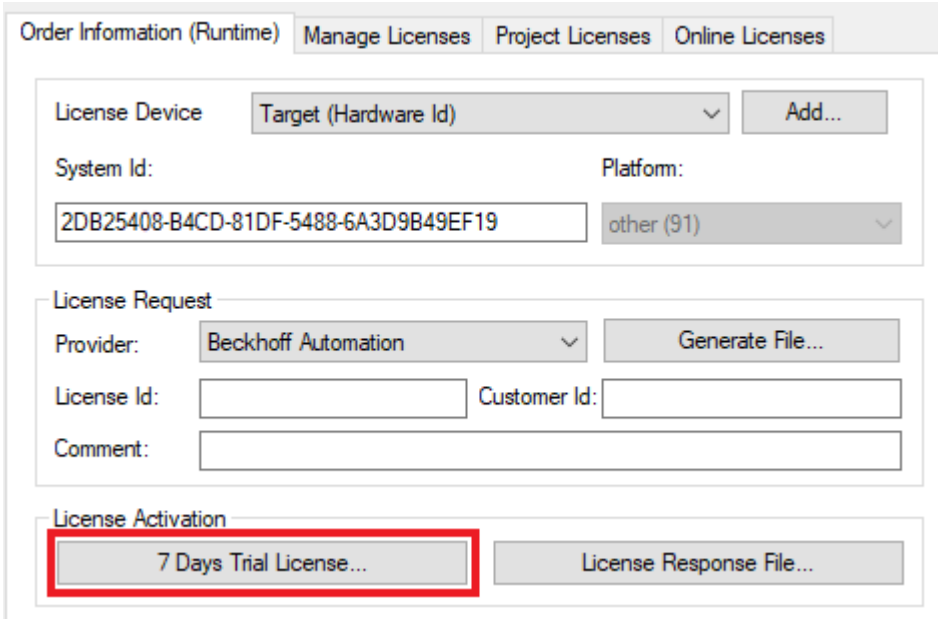


⇨ The TwinCAT 3 license manager opens.

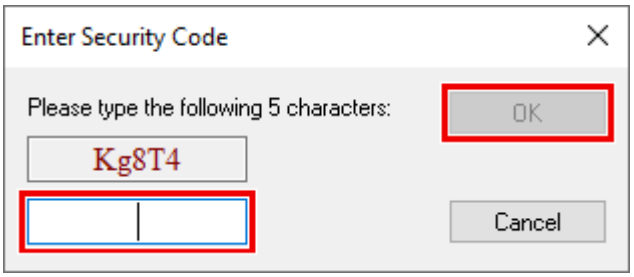
5. Open the **Manage Licenses** tab. In the **Add License** column, check the check box for the license you want to add to your project (e.g. "TF4100 TC3 Controller Toolbox").



6. Open the **Order Information (Runtime)** tab.
   ⇨ In the tabular overview of licenses, the previously selected license is displayed with the status “missing”**.**
7. Click **7-Day Trial License...** to activate the 7-day trial license.



   ⇨ A dialog box opens, prompting you to enter the security code displayed in the dialog.



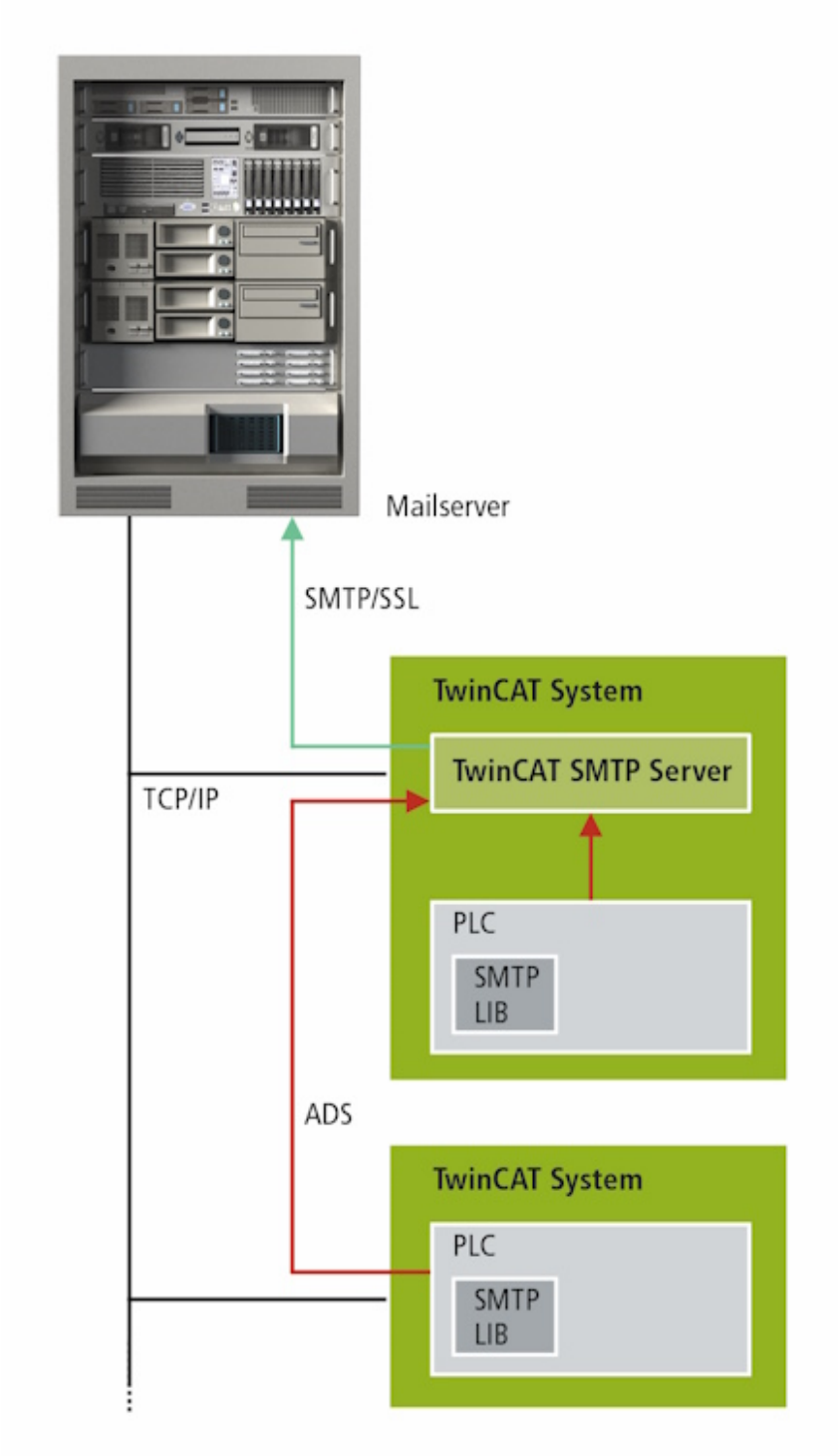
8. Enter the code exactly as it is displayed and confirm the entry.
9. Confirm the subsequent dialog, which indicates the successful activation.
   ⇨ In the tabular overview of licenses, the license status now indicates the expiry date of the license.

10. Restart the TwinCAT system.

⇨ The 7-day trial version is enabled.

4 TwinCAT SMTP

The TwinCAT SMTP Server can be used to send emails directly from the PLC. For this purpose, the ADS server is started together with TwinCAT when the installation is complete. The server is addressed via ADS from the PLC. Several function blocks [▶ 19] are available for sending emails in the PLC:



The TwinCAT SMTP Server can communicate with a local mail server or external mail providers. It supports TLS and STARTTLS encryption methods.

4.1 Configuration

The TC SMTP server uses an XML-based configuration. The **TcSmtpConfig.xml** is located in the installation directory of the Function.

Default configuration:

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

Explanation of the XML configuration file

EnableLogFile: a log file is created in the installation directory.

Authentication: selection of the authentication method. Use option 1 as default.

Port: option 0 uses the default ports for sending mail.

ContentEncoding: defines the decoding of the content.

Timeout: timeout for sending mail in ms

Charset: defines the character encoding.

Reconnects: number of repetitions

Requirements

Development Environment	Target platform	PLC library to include
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Smtp

4.2 PLC API

The TwinCAT library provides function blocks for sending emails to the TwinCAT SMTP Server.

Name	Description
FB_SmtpV3 [▶ 20]	Function block for sending emails to the TwinCAT SMTP Server
FB_SmtpV3_Full [▶ 22]	Function block with additional properties (e.g. send attachments, set priorities, HTML messages)

Ensure you have the latest version

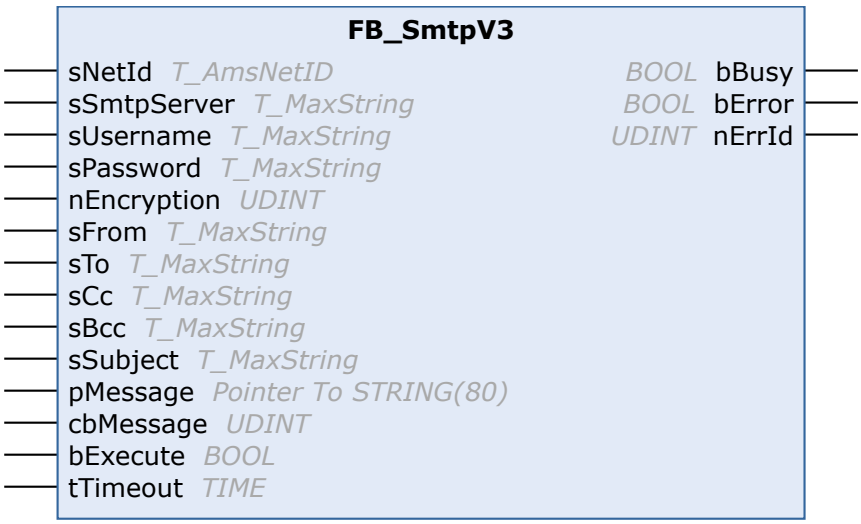
The function blocks in the folder marked as obsolete/discontinued are obsolete. Please use FB_SmtpV3 and FB_SmtpV3_Full, which offer the same functionalities.

Requirements

Development environment	Target system	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Smtp

4.2.1 Function blocks

4.2.1.1 FB_SmtpV3



The function block FB_SmtpV3 sends a byte stream to an ADS device. The TwinCAT SMTP Server must be running on the target system in order to receive the stream and send it as an email. Once the byte stream is sent, the email is sent.

Inputs

```
VAR_INPUT
    sNetId      : T_AmsNetID; (* AmsNetID *)
    sSmtpServer : T_MaxString; (* Smtp Server address ( IP or Name) *)
    sUsername   : T_MaxString; (* Smtp Username *)
    sPassword   : T_MaxString; (* Smtp 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
```

Name	Type	Description
sNetId	T_AmsNetID	AmsNetID on which the TwinCAT SMTP Server is running.
sSmtpServer	T_MaxString	Name or IP of the SMTP server to be reached
sUsername	T_MaxString	User name of the SMTP server to be reached
sPassword	T_MaxString	Password of the SMTP server to be reached
nEncryption	UDINT	SMTP encryption type: 0 = NONE 1 = STARTTLS 2 = TLS
sFrom	T_MaxString	A string containing the email address or name of the sender. The sender must be specified. The string is limited to 255 characters.
sTo	T_MaxString	A string containing the recipient's email address. At least one recipient must be entered. However, it is also possible to enter multiple addresses. These must be separated by a semicolon. The string is limited to 255 characters.

Name	Type	Description
sCc	T_MaxString	A string containing the email addresses of additional recipients (cc=carbon copy). It is possible to enter multiple recipient addresses. These must then be separated by a semicolon. The string can also remain empty. A copy of the email will be sent to the recipient(s). The recipient's email address is **visible** to other recipients. The string is limited to 255 characters.
sBcc	T_MaxString	A string containing the email addresses of additional recipients (Bcc = blind carbon copy). It is possible to enter multiple recipient addresses. These must then be separated by a semicolon. The string can also remain empty. A copy of the email will be sent to the recipient(s). The recipients' email address is **not visible** to other recipients. The string is limited to 255 characters.
sSubject	T_MaxString	This string contains the subject of the email. If the email is sent without a subject, the computer name of the sender is automatically written in the subject line (e.g. "Email sent by: CX-00762C"). The subject line string is limited to 255 characters.
pMessage	POINTER TO STRING	This parameter specifies the address of the string containing the message text. If the email is sent without text, the date and time will be inserted automatically (e.g. "Mail send at Thu, 23 Mar 2006 02:31:44 -0800"). The address of the string can be specified with the ADR operator.
cbMessage	UDINT	Length of the email text. The length can be specified by the LEN operator.
bExecute	BOOL	The function block is enabled by a rising edge on this input variable.
tTimeout	TIME	The maximum time allowed to execute a command.

Outputs

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

Name	Type	Description
bBusy	BOOL	The output variable remains TRUE until the function block has executed a command, but only until tTimeOut has expired.
bError	BOOL	The output variable is switched to TRUE as soon as an error occurs during the execution of the command. The command-specific error is contained in nErrId.
nErrId	UDINT	Contains the command-specific error code of the most recently executed command.

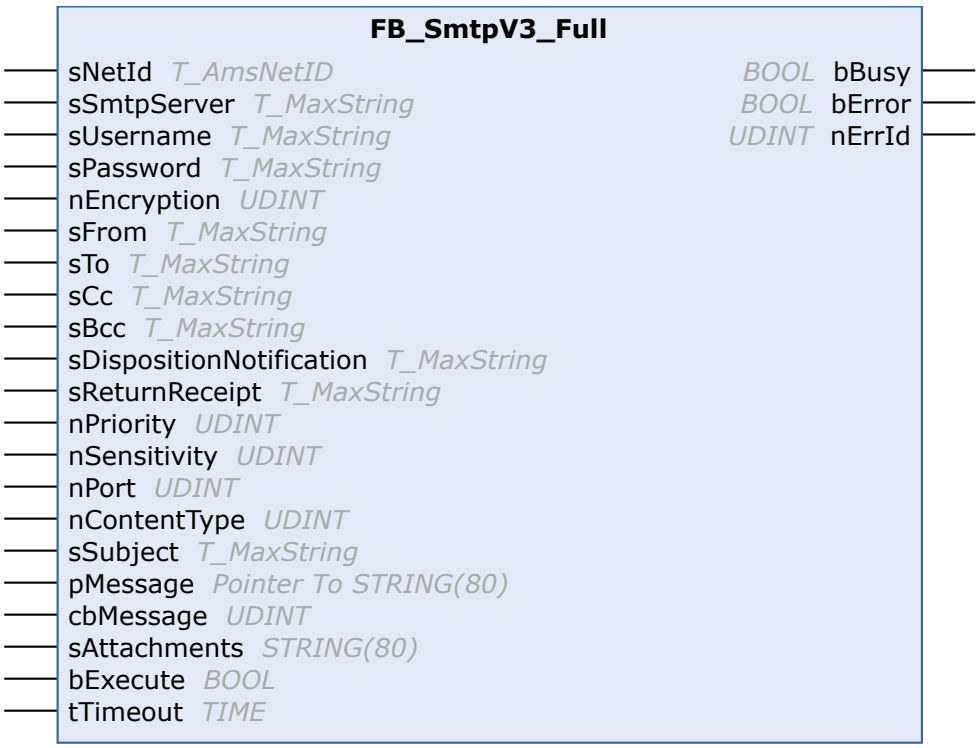
Make sure, that you don't use \o within byte-arrays. Otherwise the Message will be cut.

The maximum amount of characters, that can be used in a message, is 510.725 - you have 1275 characters for From, To, Cc, Bcc and Subject.

Requirements

Development Environment	Target platform	PLC library to include
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Smtp

4.2.1.2 FB_SmtpV3_Full



This function block FB_SmtpV3_Full communicates with the TwinCAT SMTP Server via ADS. It offers very extensive email functionalities, such as prioritizing emails from the PLC. The individual parameters are explained in detail in this documentation.

Inputs

```
VAR_INPUT
    sNetId          : T_AmsNetID;     (* AmsNetID *)
    sSmtpServer     : T_MaxString;    (* Smtp Server addres ( IP or Name) *)
    sUsername       : T_MaxString;    (* Smtp Username *)
    sPassword       : T_MaxString;    (* Smtp 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 recipent string *)
    sReturnReceipt  : T_MaxString;   (* Return recipent 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
```

Name	Type	Description
sNetId	T_AmsNetID	AmsNetID on which the TwinCAT SMTP Server is running.
sSmtpServer	T_MaxString	Name or IP of the SMTP server to be reached
sUsername	T_MaxString	User name of the SMTP server to be reached
sPassword	T_MaxString	Password of the SMTP server to be reached
nEncryption	UDINT	SMTP encryption type: 0 = NONE 1 = STARTTLS 2 = TLS

Name	Type	Description
sFrom	T_MaxString	A string containing the email address of the sender. The sender must be specified. The string is limited to 255 characters.
sTo	T_MaxString	A string containing the recipient's email address. At least one recipient must be entered. It is also possible to enter multiple addresses. These must be separated by a semicolon. The string is limited to 255 characters.
sCc	T_MaxString	A string containing the email addresses of additional recipients (cc=carbon copy). It is possible to enter multiple recipient addresses. These must then be separated by a semicolon. The string can also remain empty. A copy of the email will be sent to the recipient(s). The recipient's email address is **visible** to other recipients. The string is limited to 255 characters.
sBcc	T_MaxString	A string containing the email addresses of additional recipients (Bcc = blind carbon copy). It is possible to enter multiple recipient addresses. These must then be separated by a semicolon. The string can also remain empty. A copy of the email will be sent to the recipient(s). The recipients' email address is **not visible** to other recipients. The string is limited to 255 characters.
sDispositionNotification	T_MaxString	The email address specified here, receives a read notification from the recipients of sTo and sCc. However, the prerequisite for this is that it is also sent by the recipients.
sReturnReceipt	T_MaxString	A transmission confirmation of the sent email is sent to the email address specified here.
nPriority	UDINT	You can use this parameter to set the priority of the email: 1 = Highest 2 = not used 3 = Normal 4 = not used 5 = Lowest
nSensitivity	UDINT	You can use this parameter to set the confidentiality of the message: 0 = Private 1 = Personal 2 = Normal 3 = Confidential
nPort	UDINT	Here you can select the communication port. If you do not enter your own port, the default port 25 is used.
nContentType	UDINT	This parameter makes it possible, for example, to make HTML code, which is passed to the function block by pointer (pMessage) and size (cbMessage) of a string variable, readable in the email.
sSubject	T_MaxString	This string contains the subject of the email. If the email is sent without a subject, the computer name of the sender is automatically written in the subject line (e.g. "Email sent by: CX-00762C"). The subject line string is limited to 255 characters.
pMessage	POINTER TO STRING	This parameter specifies the address of the string containing the message text. If the email is sent without text, the date and time will be inserted automatically (e.g. "Mail send at Thu, 23 Mar 2006 02:31:44 -0800"). The address of the string can be determined with the ADR operator.
cbMessage	UDINT	Length of the email text. The length can be specified by the LEN operator.
bExecute	ARRAY [0..32] OF STRING	The function block is enabled by a rising edge on this input variable.

Name	Type	Description
sAttachments	BOOL	List of attachments (path and file name) to be sent.
tTimeout	TIME	The maximum time allowed to execute a command.

Outputs

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

Name	Type	Description
bBusy	BOOL	The output remains TRUE until the function block has executed a command, but only until tTimeOut has expired.
bError	BOOL	The output is switched to TRUE as soon as an error occurs during the execution of the command. The command-specific error is contained in nErrId.
nErrId	UDINT	Contains the command-specific error code of the most recently executed command (see table [▶ 35]).

Make sure, that you don't use \o within byte-arrays. Otherwise the Message will be cut.

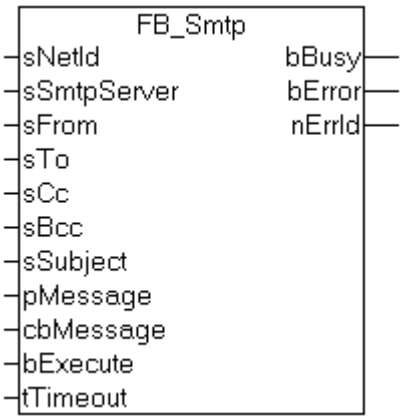
The maximum amount of characters, that can be used in a message, is 510.725 - you have 1275 characters for From, To, Cc, Bcc and Subject.

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Smtp

4.2.1.3 [obsolete functions]

4.2.1.3.1 FB_Smtp



The function block FB_Smtp sends a byte stream to an ADS remote device via ADS. The TwinCAT ADS SMTP service must be running on the remote ADS device, so that the byte stream can be received and processed into an email. The email is then sent after bytesteam processing.
Please note that the SMTP server does not support password verification, as the TwinCAT ADS SMTP service does not log in to the server via password verification.

Inputs

```
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
```

Name	Type	Description
sNetId	T_AmsNetID	AmsNetID, on which the TwinCAT mail server runs.
sSmtpServer	T_MaxString	Name or IP of the SMTP server
sFrom	T_MaxString	A string containing the email address of the sender. A sender must be specified. The string is limited to 255 characters.
sTo	T_MaxString	A string containing the email addresses of the recipients. Several addresses can be specified. These must then be separated by a semicolon. At least one recipient has to be specified. The string is limited to 255 characters.
sCc	T_MaxString	A string containing an email address of a further recipient (cc=carbon copy). This string can also be empty. A copy of the email is sent to this recipient. The email address of this recipient is **visible** to other recipients. It is possible to enter multiple recipient addresses separated by semicolons. The string is limited to 255 characters.
sBcc	T_MaxString	A string containing the email address of a further recipient (Bcc = blind carbon copy). This string can also be empty. A copy of the email is sent to this\these recipient\s. The email address of this recipient is not visible to other recipients. It is possible to enter multiple recipient addresses separated by semicolons. The string is limited to 255 characters.
sSubject	T_MaxString	A string containing the subject line for the email. The email can also be sent without a subject. The name of the sending computer is then automatically entered in the subject line (e.g., “Mail sent from: CX_00762C”). The string for the subject line is limited to 255 characters.
pMessage	DWORD	The address (a pointer) to a null-terminated string containing the email text. The email can also be sent without body text; in this case, a short sentence with the date and time of sending is automatically added (e.g., “Mail sent at: Thu, 23 Mar 2006 02:31:44 -0800”). The address of the string can be determined with the ADR operator.
cbMessage	UDINT	The length of the email text. The length can be determined using the LEN operator.
bExecute	BOOL	A rising edge at this input activates the function block.
tTimeout	TIME	Maximum time allowed for the execution of the command.

Outputs

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

Name	Type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the tTimeOut input.
bError	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in iErrorId.
nErrId	UDINT	Contains the command-specific error code of the most recently executed command (see table [▶ 35]).

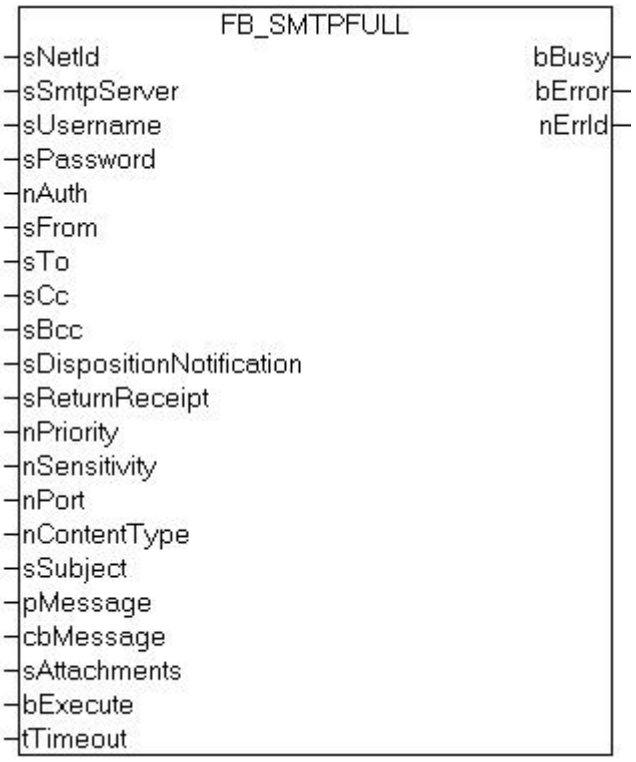
Make sure, that you don't use \o within byte-arrays. Otherwise the Message will be cut.

The maximum amount of characters, that can be used in a message, is 510.725 - you have 1275 characters for From, To, Cc, Bcc and Subject.

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Smtp

4.2.1.3.2 FB_SmtpFull



The function block FB_SmtpFull communicates with the TwinCAT SMTP Server via ADS. It offers very extensive email functionalities, such as prioritizing emails from the PLC. The individual parameters are explained in detail in this documentation.

Inputs

```
VAR_INPUT
    sNetId          : T_AmsNetID;     (* AmsNetID *)
    sSmtpServer     : T_MaxString;    (* Smtp Server addres ( IP or Name) *)
    sUsername       : T_MaxString;    (* Smtp Username *)
    sPassword       : T_MaxString;    (* Smtp Password *)
    nAuth           : UDINT;          (* Smtp 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 recipent string *)
    sReturnReceipt : T_MaxString;    (* Return recipent 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
```

Name	Type	Description
sNetId	T_AmsNetID	AmsNetID on which the TwinCAT SMTP Server is running.
sSmtpServer	T_MaxString	Name or IP of the SMTP server
sUsername	T_MaxString	User name of the SMTP server
sPassword	T_MaxString	Password for the SMTP server
nAuth	UDINT	Smtp Auth Type: 0 = AUTH NONE 1 = RESERVED 2 = AUTH LOGIN 3 = AUTH NTLM 4 = AUTH PLAIN
sFrom	T_MaxString	A string containing the email address of the sender. The sender must be specified. The string is limited to 255 characters.
sTo	T_MaxString	A string containing the recipient's email address. At least one recipient must be entered. It is also possible to enter multiple addresses. These must be separated by a semicolon. The string is limited to 255 characters.
sCc	T_MaxString	A string containing the email addresses of additional recipients (cc=carbon copy). It is possible to enter multiple recipient addresses. These must then be separated by a semicolon. The string can also remain empty. A copy of the email will be sent to the recipient(s). The recipient's email address is **visible** to other recipients. The string is limited to 255 characters.
sBcc	T_MaxString	A string containing the email addresses of additional recipients (Bcc = blind carbon copy). It is possible to enter multiple recipient addresses. These must then be separated by a semicolon. The string can also remain empty. A copy of the email will be sent to the recipient(s). The recipients' email address is **not visible** to other recipients. The string is limited to 255 characters.
sDispositionNotification	T_MaxString	The email address specified here, receives a read notification from the recipients of sTo and sCc. However, the prerequisite for this is that it is also sent by the recipients.
sReturnReceipt	T_MaxString	A transmission confirmation of the sent email is sent to the email address specified here.
nPriority	UDINT	You can use this parameter to set the priority of the email: 1 = Highest 2 = not used 3 = Normal 4 = not used 5 = Lowest
nSensitivity	UDINT	You can use this parameter to set the confidentiality of the message: 0 = Private

Name	Type	Description
		1 = Personal 2 = Normal 3 = Confidential
nPort	UDINT	Here you can select the communication port. If you do not enter your own port, the default port 25 is used.
nContentType	UDINT	This parameter makes it possible, for example, to make HTML code, which is passed to the function block by pointer (pMessage) and size (cbMessage) of a string variable, readable in the email.
sSubject	T_MaxString	This string contains the subject of the email. If the email is sent without a subject, the computer name of the sender is automatically written in the subject line (e.g. "Email sent by: CX00762C"). The subject line string is limited to 255 characters.
pMessage	DWORD	This parameter specifies the address of the string containing the message text. If the email is sent without text, the date and time will be inserted automatically (e.g. "Mail send at Thu, 23 Mar 2006 02:31:44 -0800"). The address of the string can be determined with the ADR operator.
cbMessage	UDINT	Length of the email text. The length can be specified by the LEN operator.
bExecute	ARRAY [0..32] OF STRING	The function block is enabled by a rising edge on this input variable.
sAttachments	BOOL	Listing of file names
tTimeout	TIME	The maximum time allowed to execute a command.

Outputs

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

Name	Type	Description
bBusy	BOOL	The output remains TRUE until the function block has executed a command, but only until tTimeOut has expired.
bError	BOOL	The output is switched to TRUE as soon as an error occurs during the execution of the command. The command-specific error is contained in iErrorId.
nErrId	UDINT	Contains the command-specific error code of the most recently executed command (see table [▶ 35]).

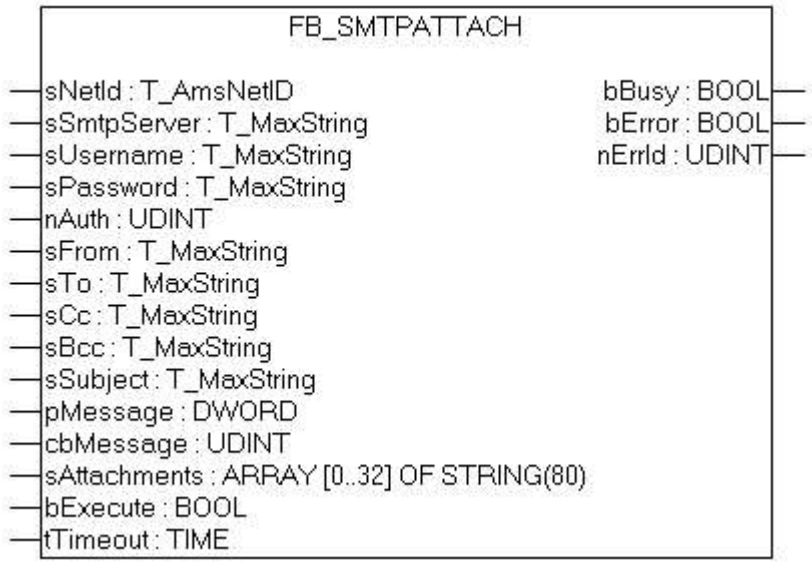
Make sure, that you don't use \o within byte-arrays. Otherwise the Message will be cut.

The maximum amount of characters, that can be used in a message, is 510.725 - you have 1275 characters for From, To, Cc, Bcc and Subject.

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Smtp

4.2.1.3.3 FB_SmtpAttach



The function block FB_SmtpAttach sends a data stream to a remote ADS device via ADS. The TwinCAT ADS SMTP service must be running on the remote ADS device so that the data stream can be received and processed into an email. As soon as the data stream has been processed, the email is sent.

Inputs

```
VAR_INPUT
    sNetId       : T_AmsNetID;     (* AmsNetID *)
    sSmtpServer  : T_MaxString;    (* Smtp Server addres ( IP or Name) *)
    sUsername    : T_MaxString;    (* Smtp Username *)
    sPassword    : T_MaxString;    (* Smtp Password *)
    nAuth        : UDINT;          (* Smtp 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
```

Name	Type	Description
sNetId	T_AmsNetID	AmsNetID on which the TwinCAT SMS server is running.
sSmtpServer	T_MaxString	Name or IP of the SMTP server
sUsername	T_MaxString	User name of the SMTP server
sPassword	T_MaxString	Password for the SMTP server
nAuth	UDINT	Smtp Auth Type: 0 = AUTH NONE 1 = RESERVED 2 = AUTH LOGIN 3 = AUTH NTLM 4 = AUTH PLAIN
sFrom	T_MaxString	A string containing the email address of the sender. The sender must be specified. The string is limited to 255 characters.
sTo	T_MaxString	A string containing the recipient's email address. At least one recipient must be entered. It is also possible to enter multiple addresses. These must be separated by a semicolon. The string is limited to 255 characters.

Name	Type	Description
sCc	T_MaxString	A string containing the email addresses of additional recipients (cc=carbon copy). It is possible to enter multiple recipient addresses. These must then be separated by a semicolon. However, the string can also remain empty. A copy of the email will be sent to the recipient(s). The recipient's email address is **visible** to other recipients. The string is limited to 255 characters.
sBcc	T_MaxString	A string containing the email addresses of additional recipients (Bcc = blind carbon copy). It is possible to enter multiple recipient addresses. These must then be separated by a semicolon. However, the string can also remain empty. A copy of the email will be sent to the recipient(s). The recipients' email address is **not visible** to other recipients. The string is limited to 255 characters.
sSubject	T_MaxString	This string contains the subject of the email. If the email is sent without a subject, the computer name of the sender is automatically written in the subject line (e.g. "Email sent by: CX00762C"). The subject line string is limited to 255 characters.
pMessage	DWORD	This parameter specifies the address of the string containing the message text. If the email is sent without text, the date and time will be inserted automatically (e.g. "Mail send at Thu, 23 Mar 2006 02:31:44 -0800"). The address of the string can be specified with the ADR operator.
cbMessage	UDINT	Length of the email text. The length can be specified by the LEN operator.
bExecute	ARRAY [0..32] OF STRING	The function block is enabled by a rising edge on this input variable.
sAttachments	BOOL	Listing of file names
tTimeout	TIME	The maximum time allowed to execute a command.

Outputs

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

Name	Type	Description
bBusy	BOOL	The output variable remains TRUE until the function block has executed a command, but only until tTimeOut has expired.
bError	BOOL	The output variable is switched to TRUE as soon as an error occurs during the execution of the command. The command-specific error is contained in iErrorId.
nErrId	UDINT	Contains the command-specific error code of the most recently executed command (see table [▶ 35]).

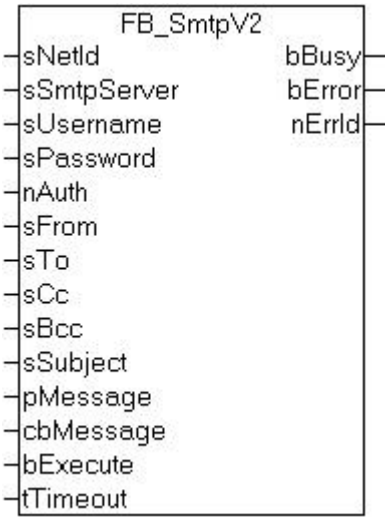
Make sure, that you don't use \o within byte-arrays. Otherwise the Message will be cut.

The maximum amount of characters, that can be used in a message, is 510.725 - you have 1275 characters for From, To, Cc, Bcc and Subject.

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Smtp

4.2.1.3.4 FB_SmtpV2



The function block FB_SmtpV2 sends a data stream to a remote ADS device via ADS. The TwinCAT ADS SMTP service must be running on the remote ADS device so that the data stream can be received and processed into an email. As soon as the data stream has been processed, the email is sent.

Inputs

```
VAR_INPUT
    sNetId       : T_AmsNetID;  (* AmsNetID *)
    sSmtpServer  : T_MaxString; (* Smtp Server addres ( IP or Name) *)
    sUsername    : T_MaxString; (* Smtp Username *)
    sPassword    : T_MaxString; (* Smtp Password *)
    nAuth        : UDINT;       (* Smtp 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
```

Name	Type	Description
sNetId	T_AmsNetID	AmsNetID on which the TwinCAT SMS server is running.
sSmtpServer	T_MaxString	Name or IP of the SMTP server
sUsername	T_MaxString	User name of the SMTP server
sPassword	T_MaxString	Password for the SMTP server
nAuth	UDINT	Smtp Auth Type: 0 = AUTH NONE 1 = RESERVED 2 = AUTH LOGIN 3 = AUTH NTLM 4 = AUTH PLAIN
sFrom	T_MaxString	A string containing the email address of the sender. The sender must be specified. The string is limited to 255 characters.

Name	Type	Description
sTo	T_MaxString	A string containing the recipient's email address. At least one recipient must be entered. It is also possible to enter multiple addresses. These must be separated by a semicolon. The string is limited to 255 characters.
sCc	T_MaxString	A string containing the email addresses of additional recipients (cc=carbon copy). It is possible to enter multiple recipient addresses. These must then be separated by a semicolon. However, the string can also remain empty. A copy of the email will be sent to the recipient(s). The recipient's email address is **visible** to other recipients. The string is limited to 255 characters.
sBcc	T_MaxString	A string containing the email addresses of additional recipients (Bcc = blind carbon copy). It is possible to enter multiple recipient addresses. These must then be separated by a semicolon. However, the string can also remain empty. A copy of the email will be sent to the recipient(s). The recipients' email address is **not visible** to other recipients. The string is limited to 255 characters.
sSubject	T_MaxString	This string contains the subject of the email. If the email is sent without a subject, the computer name of the sender is automatically written in the subject line (e.g. "Email sent by: CX00762C"). The subject line string is limited to 255 characters.
pMessage	DWORD	This parameter specifies the address of the string containing the message text. If the email is sent without text, the date and time will be inserted automatically (e.g. "Mail send at Thu, 23 Mar 2006 02:31:44 -0800"). The address of the string can be specified with the ADR operator.
cbMessage	UDINT	Length of the email text. The length can be specified by the LEN operator.
bExecute		The function block is enabled by a rising edge on this input variable.
tTimeout		The maximum time allowed to execute a command.

Outputs

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

Name	Type	Description
bBusy	BOOL	The output variable remains TRUE until the function block has executed a command, but only until tTimeOut has expired.
bError	BOOL	The output variable is switched to TRUE as soon as an error occurs during the execution of the command. The command-specific error is contained in iErrorId.
nErrId	UDINT	Contains the command-specific error code of the most recently executed command (see table [▶ 35]).

Make sure, that you don't use \o within byte-arrays. Otherwise the Message will be cut.

The maximum amount of characters, that can be used in a message, is 510.725 - you have 1275 characters for From, To, Cc, Bcc and Subject.

Requirements

Development Environment	Target platform	PLC library to include
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Smtp

4.3 Samples

4.3.1 Mail dispatch

With the function block FB_SmtpV3 the encrypted mail dispatch via TLS or STARTTLs is available.

Description

An email is sent after triggering the variable bStart.

ToDo: configure the mail server address and credentials.

Download: https://infosys.beckhoff.com/content/1033/TF6350_Tc3_SMS_SMTP/Resources/19622023179.zip

Program variables

```
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
```

Program 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
```

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Smtp

4.3.2 SmtpFull sample with features

With the FB_SmtpFull very extensive email functionalities are available for the PLC.

Description

These possibilities are shown in the sample:

- Among other things, the email text is transferred in HTML code, which offers completely new formatting possibilities.
- Furthermore, the priority of an email can now be set or read receipts can be requested.
- Multiple files can be sent as attachments.

An email is sent after triggering the variable bStart.

ToDo: configure the mail server address, the credentials and the paths to the file attachments.

https://infosys.beckhoff.com/content/1033/TF6350_Tc3_SMS_SMTP/Resources/19622025355.zip

Program variables

```
VAR
fbSmtpFull : FB_SmtpV3_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
```

Program code

```
fbSmtpFull(
sNetId:= '',
sSmtpServer:= '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
```

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Smtp

4.4 Appendix

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 [▶ 35]

ADS Return codes

Win Socket Error codes [▶ 36]

2. **Activate the logging option in the TcSmtpConfig.xml**

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

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

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

   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/STMP 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 Error Codes

This list gives possible error codes for the TwinCAT SMTP Server supplement product. If an error code is generated that is not included in the list, please refer to the ADS Return Codes [▶ 47] or the WinSockErrorCodes [▶ 36] list.

Error code (hex)	Error code (dec)	Description
< 0x8000	< 32778	ADS return code
0x800A	32778	Not connected
0x800B	32779	Sender expected
0x800C	32780	Recipients expected

Error code (hex)	Error code (dec)	Description
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

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4000	PC or CX (x86)	Tc2_Smtp

4.4.3 Windows Socket Error Codes

The following table describes the possible error codes, returned by the WSAGetLastError function. The errors are sorted in alphabetical order. Some error codes that are defined in Winsock2.h are not returned. They are not included in the list.

Return Value	Description
WSAEINTR10004	Interrupted function call.blocking operation was interrupted by a call to WSACancelBlockingCall.

Return Value	Description
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.
WSAEWOULDBLOCK 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 WSAEWOULDBLOCK 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.
WSAEDESTADDRREQ 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.
WSAENOPROTOOPT 10042	Bad protocol option.An unknown, invalid or unsupported option or level was specified in a getsockopt or setsockopt call.
WSAEPROTONOSUPPORT 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.
WSAESOCKTNOSUPPORT 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.
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.

Return Value	Description
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.
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.

Return Value	Description
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.WSAStartup may fail with this error if the limit has been reached.
WSASYSNOTREADY 10091	Network subsystem is unavailable.This error is returned by WSAStartup 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: • That the appropriate Windows Sockets DLL file is in the current path. • 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. • The Windows Sockets implementation documentation to be sure all necessary components are currently installed and configured correctly.
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.
WSANOTINITIALISED 10093	Successful WSAStartup not yet performed.Either the application has not called WSAStartup or WSAStartup 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 WSARecv and WSARecvFrom 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.
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

Return Value	Description
	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 WSAIoctl.
WSAINVALIDPROCTABLE 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.)
WSAINVALIDPROVIDER OS dependent	Invalid service provider version number.A service provider returned a version number other than 2.0.
WSAPROVIDERFAILEDINIT 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.
WSASYSCALLFAILURE 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.

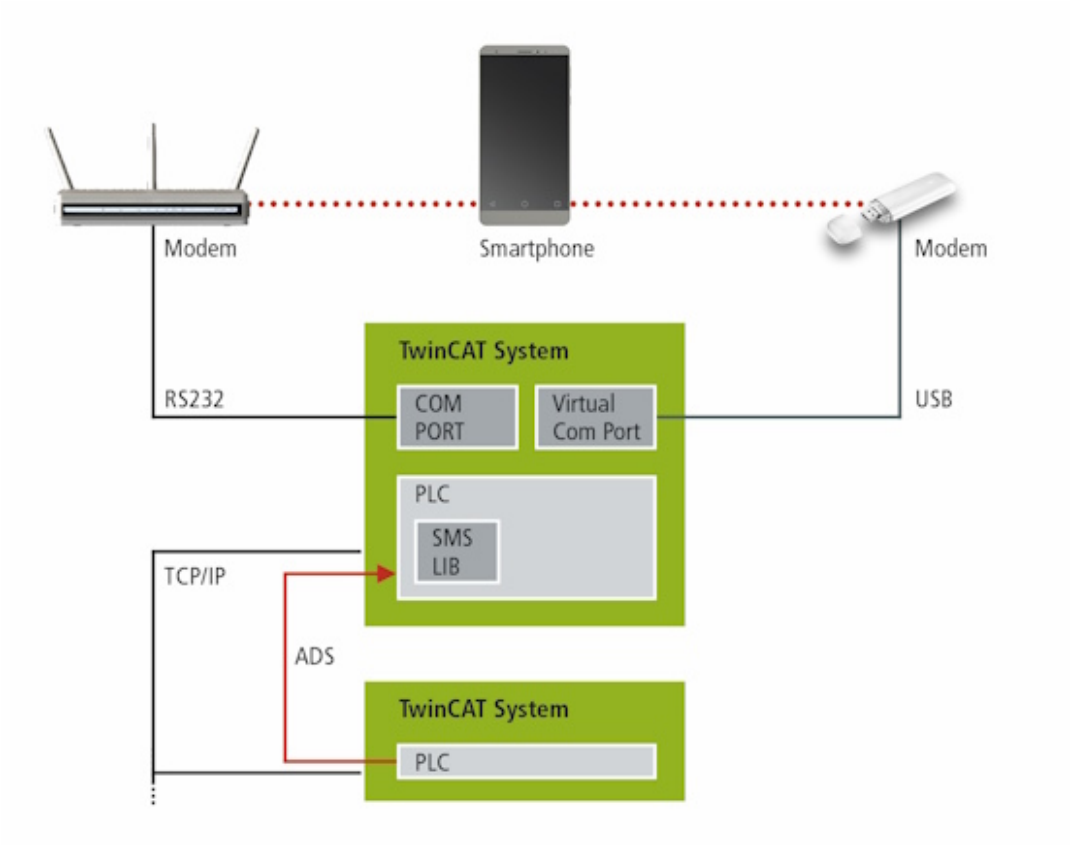
Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Smtp

5 TwinCAT SMS

The TwinCAT SMS libraries contain a function block for sending SMS messages directly from the PLC. The SMS library is based on the 'Serial Communication' library (which is also installed). This makes it possible to communicate with the PC's serial interface and with the serial terminal (ELxxxx, KL6xxx) in the same way.

In addition, via the ADS server of TF6340 TwinCAT 3 Serial Communication USB devices can also be addressed via virtual serial interfaces. We recommend the CU8210 LTE USB stick.

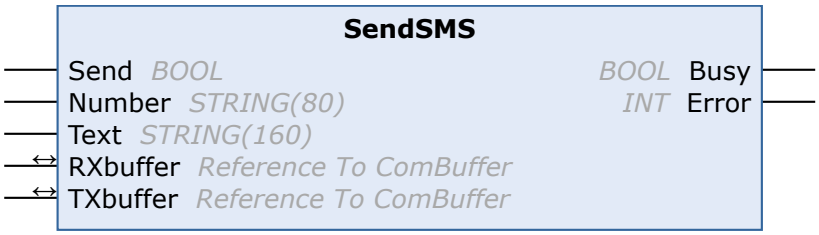


Furthermore, it is possible to output alarm or status messages from another TwinCAT system that is connected via a network.

5.1 PLC API

5.1.1 Function blocks

5.1.1.1 SendSMS



The function block SendSMS sends an SMS via a connected GSM modem. The function block is based on the 'Serial Communication' library.

Since the function block only communicates via the ComBuffer structure of the ‘Serial Communication’ library, it can be instantiated and applied to any type of serial interface.

Inputs

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

Name	Type	Description
Send	BOOL	The function block is enabled by a positive edge at this input.
Number	String	Phone number to be dialed in national format (e.g.: 0170123456)
Text	String(160)	The SMS message to be sent

Inputs/outputs

```
VAR_IN_OUT
    RXbuffer    : ComBuffer;
    TXbuffer    : ComBuffer;
END_VAR
```

Name	Type	Description
RXbuffer	ComBuffer	Structure for communication with the serial interface. An interface-specific function block in the ‘Serial Communication’ library fills this buffer with the interface data.
TXbuffer	ComBuffer	Structure for communication with the serial interface. An interface-specific function block of the ‘Serial Communication’ library transfers the data from this buffer to the interface.

These structures, and their usage, are described in more detail in the documentation for the ‘Serial Communication' library. The SendSMS function block is connected in the same way as a SendString or ReceiveString function block.

Outputs

```
VAR_OUTPUT
    Busy   : BOOL;
    Error  : INT;
END_VAR
```

Name	Type	Description
Busy	BOOL	This output is set when there is a rising edge at the Send input, and remains set until the SMS has been sent to the modem or until an error has occurred.
Error	INT	If an error occurs during SMS transmission, the Busy output is reset and an error code is output at the Error output.

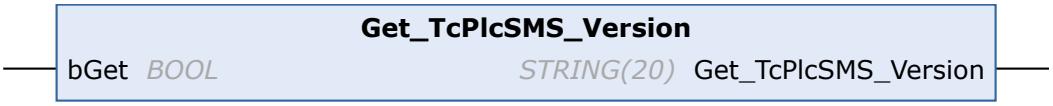
Error code	Meaning	Cause
0	No error	The SMS was successfully transmitted.
1	No communication with the modem possible.	Is the terminal configured correctly? Is the appropriate ComLib library being used?
2	Modem reports an error during configuration.	Is a compatible GSM modem connected?
3	Modem cannot send SMS.	Is the SIM card working properly? Can the card be used without entering the PIN? Is the modem connected to the network? Is a compatible modem connected?
4	Communication error.	Is the correct transmission speed set?

Requirements

Development Environment	Target platform	PLC library to include
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Sms

5.1.2 Functions

5.1.2.1 FUNCTION Get_TcPlcSMS_Version



This function can be used to read the version information of the PLC library.

FUNCTION Get_TcPlcSMS_Version: STRING(20)

Inputs

```
VAR_INPUT
    bGet   : BOOL;
END_VAR
```

Name	Type	Description
bGet	BOOL	This parameter has no significance for the function execution and can be set as required.

Requirements

Development Environment	Target platform	PLC library to include
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Sms

5.1.3 Global constants

5.1.3.1 Library version

All libraries have a certain version. The version is indicated in the PLC library repository, for example. A global constant contains the information about the library version:

Global_Version

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

ST_LibVersion

To see whether the version you have is the one you need, use the function F_CmpLibVersion (defined in the Tc2_System library).

Compare versions

All other options for comparing library versions, which you may know from TwinCAT 2, are outdated.

5.2 Samples

5.2.1 Sending an SMS via the Beckhoff 4G stick

The installed 4G stick is connected as a virtual serial interface, through which the PLC can connect via ADS. By calling the function block "SendSMS" the SMS is sent to a recipient.

1. Enter in the data structure stSerialCfg.
2. Enter in the string variable sNumber of the recipient.
3. Send an SMS by a rising edge at bConnect.

Program variables

```
(*
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
```

Program 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);
```

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Sms, Tc2_SerialCom Library

5.3 Appendix

5.3.1 Fault Finding

There are a number of reasons why an SMS may fail to be sent with the SendSMS function block or via the SMS COM server:

- no connection to the GSM modem
- the use of an unsupported GSM modem
- incorrect phone number
- incorrect GSM network
- PIN required (the SIM card must not be protected by a PIN)
- serial terminal not initialized (call KL6Init)
- incorrect call of the ADS service

A variety of tools are available to look for these errors:

Using the Log File

Keeping records in a log file can be activated with the TwinCAT SMS Server Configurator. Once this has been done, all the messages sent and the errors are written into the TcSmsSrvCfg.xml file. The file can be found in the TwinCAT installation directory.

NT Event Log

Errors when sending messages are also always recorded in the NT Event Log. The Event Log can be opened through the TwinCAT icon on the task bar.

ADS Error Messages

If the call to an ADS Function fails, the error is coded in the function's return value. A list of these error codes can be found under ADS Return Codes. [▶ 47]

Configuration of the terminal

The serial terminal can be configured in different ways. Terminals that have been differently configured, have to some extent a different representation in the process image (3 byte /5 byte terminals, advanced/standard). Please note that the ‘Serial Communication’ library must match the configuration of the terminal. See also the documentation for the KL6xxx and the ComLib documentation:

It is also important to match the transmission speed of the terminal to the modem used.

Sending a Test SMS

A test SMS can easily be sent with the Visual Basic example program, to find out whether an error lies with the ADS call or in the configuration of the SMS Server.

Sending a test SMS with a mobile phone

Enter number.

Network Selection with the Westermo GS-01

GS-01 has a number of variations for the various networks in Europe and in the USA. The lamp on the front of the modem indicates whether a network is available. The lamp flashes if the modem is connected to a network. If the lamp is continuously illuminated, the fault finding section should be consulted in the Westermo manual.

6 Appendix

6.1 ADS Return Codes

Grouping of error codes:
Global error codes: 0x0000 [▶ 47]... (0x9811_0000 ...)
Router error codes: 0x500 [▶ 47]... (0x9811_0500 ...)
General ADS errors: 0x700 [▶ 48]... (0x9811_0700 ...)
RTime error codes: 0x1000 [▶ 49]... (0x9811_1000 ...)

Global error codes

Hex	Dec	HRESULT	Name	Description
0x0	0	0x98110000	ERR_NOERROR	No error.
0x1	1	0x98110001	ERR_INTERNAL	Internal error.
0x2	2	0x98110002	ERR_NORTIME	No real time.
0x3	3	0x98110003	ERR_ALLOCLOCKEDMEM	Allocation locked – memory error.
0x4	4	0x98110004	ERR_INSERTMAILBOX	Mailbox full – the ADS message could not be sent. Reducing the number of ADS messages per cycle will help.
0x5	5	0x98110005	ERR_WRONGRECEIVEHMSG	Wrong HMSG.
0x6	6	0x98110006	ERR_TARGETPORTNOTFOUND	Target port not found – ADS server is not started, not reachable or not installed.
0x7	7	0x98110007	ERR_TARGETMACHINENOTFOUND	Target computer not found – AMS route was not found.
0x8	8	0x98110008	ERR_UNKNOWNCMDID	Unknown command ID.
0x9	9	0x98110009	ERR_BADTASKID	Invalid task ID.
0xA	10	0x9811000A	ERR_NOIO	No IO.
0xB	11	0x9811000B	ERR_UNKNOWNAMSCMD	Unknown AMS command.
0xC	12	0x9811000C	ERR_WIN32ERROR	Win32 error.
0xD	13	0x9811000D	ERR_PORTNOTCONNECTED	Port not connected.
0xE	14	0x9811000E	ERR_INVALIDAMSLENGTH	Invalid AMS length.
0xF	15	0x9811000F	ERR_INVALIDAMSNETID	Invalid AMS Net ID.
0x10	16	0x98110010	ERR_LOWINSTLEVEL	Installation level is too low –TwinCAT 2 license error.
0x11	17	0x98110011	ERR_NODEBUGINTAVAILABLE	No debugging available.
0x12	18	0x98110012	ERR_PORTDISABLED	Port disabled – TwinCAT system service not started.
0x13	19	0x98110013	ERR_PORTALREADYCONNECTED	Port already connected.
0x14	20	0x98110014	ERR_AMSSYNC_W32ERROR	AMS Sync Win32 error.
0x15	21	0x98110015	ERR_AMSSYNC_TIMEOUT	AMS Sync Timeout.
0x16	22	0x98110016	ERR_AMSSYNC_AMSERROR	AMS Sync error.
0x17	23	0x98110017	ERR_AMSSYNC_NOINDEXINMAP	No index map for AMS Sync available.
0x18	24	0x98110018	ERR_INVALIDAMSPORT	Invalid AMS port.
0x19	25	0x98110019	ERR_NOMEMORY	No memory.
0x1A	26	0x9811001A	ERR_TCPSEND	TCP send error.
0x1B	27	0x9811001B	ERR_HOSTUNREACHABLE	Host unreachable.
0x1C	28	0x9811001C	ERR_INVALIDAMSFRAGMENT	Invalid AMS fragment.
0x1D	29	0x9811001D	ERR_TLSSEND	TLS send error – secure ADS connection failed.
0x1E	30	0x9811001E	ERR_ACCESSDENIED	Access denied – secure ADS access denied.

Router error codes

Hex	Dec	HRESULT	Name	Description
0x500	1280	0x98110500	ROUTERERR_NOLOCKEDMEMORY	Locked memory cannot be allocated.
0x501	1281	0x98110501	ROUTERERR_RESIZEMEMORY	The router memory size could not be changed.
0x502	1282	0x98110502	ROUTERERR_MAILBOXFULL	The mailbox has reached the maximum number of possible messages.
0x503	1283	0x98110503	ROUTERERR_DEBUGBOXFULL	The Debug mailbox has reached the maximum number of possible messages.

Hex	Dec	HRESULT	Name	Description
0x504	1284	0x98110504	ROUTERERR_UNKNOWNPORTTYPE	The port type is unknown.
0x505	1285	0x98110505	ROUTERERR_NOTINITIALIZED	The router is not initialized.
0x506	1286	0x98110506	ROUTERERR_PORTALREADYINUSE	The port number is already assigned.
0x507	1287	0x98110507	ROUTERERR_NOTREGISTERED	The port is not registered.
0x508	1288	0x98110508	ROUTERERR_NOMOREQUEUES	The maximum number of ports has been reached.
0x509	1289	0x98110509	ROUTERERR_INVALIDPORT	The port is invalid.
0x50A	1290	0x9811050A	ROUTERERR_NOTACTIVATED	The router is not active.
0x50B	1291	0x9811050B	ROUTERERR_FRAGMENTBOXFULL	The mailbox has reached the maximum number for fragmented messages.
0x50C	1292	0x9811050C	ROUTERERR_FRAGMENTTIMEOUT	A fragment timeout has occurred.
0x50D	1293	0x9811050D	ROUTERERR_TOBEREMOVED	The port is removed.

General ADS error codes

Hex	Dec	HRESULT	Name	Description
0x700	1792	0x98110700	ADSERR_DEVICE_ERROR	General device error.
0x701	1793	0x98110701	ADSERR_DEVICE_SRVNOTSUPP	Service is not supported by the server.
0x702	1794	0x98110702	ADSERR_DEVICE_INVALIDGRP	Invalid index group.
0x703	1795	0x98110703	ADSERR_DEVICE_INVALIDOFFSET	Invalid index offset.
0x704	1796	0x98110704	ADSERR_DEVICE_INVALIDACCESS	Reading or writing not permitted. Several causes are possible. For example, an incorrect password was entered when creating routes.
0x705	1797	0x98110705	ADSERR_DEVICE_INVALIDSIZE	Parameter size not correct.
0x706	1798	0x98110706	ADSERR_DEVICE_INVALIDDATA	Invalid data values.
0x707	1799	0x98110707	ADSERR_DEVICE_NOTREADY	Device is not ready to operate.
0x708	1800	0x98110708	ADSERR_DEVICE_BUSY	Device is busy.
0x709	1801	0x98110709	ADSERR_DEVICE_INVALIDCONTEXT	Invalid operating system context. This can result from use of ADS blocks in different tasks. It may be possible to resolve this through multitasking synchronization in the PLC.
0x70A	1802	0x9811070A	ADSERR_DEVICE_NOMEMORY	Insufficient memory.
0x70B	1803	0x9811070B	ADSERR_DEVICE_INVALIDPARM	Invalid parameter values.
0x70C	1804	0x9811070C	ADSERR_DEVICE_NOTFOUND	Not found (files, ...).
0x70D	1805	0x9811070D	ADSERR_DEVICE_SYNTAX	Syntax error in file or command.
0x70E	1806	0x9811070E	ADSERR_DEVICE_INCOMPATIBLE	Objects do not match.
0x70F	1807	0x9811070F	ADSERR_DEVICE_EXISTS	Object already exists.
0x710	1808	0x98110710	ADSERR_DEVICE_SYMBOLNOTFOUND	Symbol not found.
0x711	1809	0x98110711	ADSERR_DEVICE_SYMBOLVERSIONINVALID	Invalid symbol version. This can occur due to an online change. Create a new handle.
0x712	1810	0x98110712	ADSERR_DEVICE_INVALIDSTATE	Device (server) is in invalid state.
0x713	1811	0x98110713	ADSERR_DEVICE_TRANSMODENOTSUPP	AdsTransMode not supported.
0x714	1812	0x98110714	ADSERR_DEVICE_NOTIFYHNDINVALID	Notification handle is invalid.
0x715	1813	0x98110715	ADSERR_DEVICE_CLIENTUNKNOWN	Notification client not registered.
0x716	1814	0x98110716	ADSERR_DEVICE_NOMOREHDLS	No further handle available.
0x717	1815	0x98110717	ADSERR_DEVICE_INVALIDWATCHSIZE	Notification size too large.
0x718	1816	0x98110718	ADSERR_DEVICE_NOTINIT	Device not initialized.
0x719	1817	0x98110719	ADSERR_DEVICE_TIMEOUT	Device has a timeout.
0x71A	1818	0x9811071A	ADSERR_DEVICE_NOINTERFACE	Interface query failed.
0x71B	1819	0x9811071B	ADSERR_DEVICE_INVALIDINTERFACE	Wrong interface requested.
0x71C	1820	0x9811071C	ADSERR_DEVICE_INVALIDCLSID	Class ID is invalid.
0x71D	1821	0x9811071D	ADSERR_DEVICE_INVALIDOBJID	Object ID is invalid.
0x71E	1822	0x9811071E	ADSERR_DEVICE_PENDING	Request pending.
0x71F	1823	0x9811071F	ADSERR_DEVICE_ABORTED	Request is aborted.
0x720	1824	0x98110720	ADSERR_DEVICE_WARNING	Signal warning.
0x721	1825	0x98110721	ADSERR_DEVICE_INVALIDARRAYIDX	Invalid array index.
0x722	1826	0x98110722	ADSERR_DEVICE_SYMBOLNOTACTIVE	Symbol not active.
0x723	1827	0x98110723	ADSERR_DEVICE_ACCESSDENIED	Access denied. Several causes are possible. For example, a unidirectional ADS route is used in the opposite direction.

Hex	Dec	HRESULT	Name	Description
0x724	1828	0x98110724	ADSERR_DEVICE_LICENSENOTFOUND	Missing license.
0x725	1829	0x98110725	ADSERR_DEVICE_LICENSEEXPIRED	License expired.
0x726	1830	0x98110726	ADSERR_DEVICE_LICENSEEXCEEDED	License exceeded.
0x727	1831	0x98110727	ADSERR_DEVICE_LICENSEINVALID	Invalid license.
0x728	1832	0x98110728	ADSERR_DEVICE_LICENSESYSTEMID	License problem: System ID is invalid.
0x729	1833	0x98110729	ADSERR_DEVICE_LICENSENOTIMELIMIT	License not limited in time.
0x72A	1834	0x9811072A	ADSERR_DEVICE_LICENSEFUTUREISSUE	Licensing problem: time in the future.
0x72B	1835	0x9811072B	ADSERR_DEVICE_LICENSETIMETOLONG	License period too long.
0x72C	1836	0x9811072C	ADSERR_DEVICE_EXCEPTION	Exception at system startup.
0x72D	1837	0x9811072D	ADSERR_DEVICE_LICENSEDUPLICATED	License file read twice.
0x72E	1838	0x9811072E	ADSERR_DEVICE_SIGNATUREINVALID	Invalid signature.
0x72F	1839	0x9811072F	ADSERR_DEVICE_CERTIFICATEINVALID	Invalid certificate.
0x730	1840	0x98110730	ADSERR_DEVICE_LICENSEOEMNOTFOUND	Public key not known from OEM.
0x731	1841	0x98110731	ADSERR_DEVICE_LICENSERESTRICTED	License not valid for this system ID.
0x732	1842	0x98110732	ADSERR_DEVICE_LICENSEDEMODENIED	Demo license prohibited.
0x733	1843	0x98110733	ADSERR_DEVICE_INVALIDFNCID	Invalid function ID.
0x734	1844	0x98110734	ADSERR_DEVICE_OUTOFRANGE	Outside the valid range.
0x735	1845	0x98110735	ADSERR_DEVICE_INVALIDALIGNMENT	Invalid alignment.
0x736	1846	0x98110736	ADSERR_DEVICE_LICENSEPLATFORM	Invalid platform level.
0x737	1847	0x98110737	ADSERR_DEVICE_FORWARD_PL	Context – forward to passive level.
0x738	1848	0x98110738	ADSERR_DEVICE_FORWARD_DL	Context – forward to dispatch level.
0x739	1849	0x98110739	ADSERR_DEVICE_FORWARD_RT	Context – forward to real-time.
0x740	1856	0x98110740	ADSERR_CLIENT_ERROR	Client error.
0x741	1857	0x98110741	ADSERR_CLIENT_INVALIDPARM	Service contains an invalid parameter.
0x742	1858	0x98110742	ADSERR_CLIENT_LISTEMPTY	Polling list is empty.
0x743	1859	0x98110743	ADSERR_CLIENT_VARUSED	Var connection already in use.
0x744	1860	0x98110744	ADSERR_CLIENT_DUPLINVOKEID	The called ID is already in use.
0x745	1861	0x98110745	ADSERR_CLIENT_SYNCTIMEOUT	Timeout has occurred – the remote terminal is not responding in the specified ADS timeout. The route setting of the remote terminal may be configured incorrectly.
0x746	1862	0x98110746	ADSERR_CLIENT_W32ERROR	Error in Win32 subsystem.
0x747	1863	0x98110747	ADSERR_CLIENT_TIMEOUTINVALID	Invalid client timeout value.
0x748	1864	0x98110748	ADSERR_CLIENT_PORTNOTOPEN	Port not open.
0x749	1865	0x98110749	ADSERR_CLIENT_NOAMSADDR	No AMS address.
0x750	1872	0x98110750	ADSERR_CLIENT_SYNCINTERNAL	Internal error in Ads sync.
0x751	1873	0x98110751	ADSERR_CLIENT_ADDHASH	Hash table overflow.
0x752	1874	0x98110752	ADSERR_CLIENT_REMOVEHASH	Key not found in the table.
0x753	1875	0x98110753	ADSERR_CLIENT_NOMORESYM	No symbols in the cache.
0x754	1876	0x98110754	ADSERR_CLIENT_SYNCRESINVALID	Invalid response received.
0x755	1877	0x98110755	ADSERR_CLIENT_SYNCPORTLOCKED	Sync Port is locked.
0x756	1878	0x98110756	ADSERR_CLIENT_REQUESTCANCELLED	The request was canceled.

RTime error codes

Hex	Dec	HRESULT	Name	Description
0x1000	4096	0x98111000	RTERR_INTERNAL	Internal error in the real-time system.
0x1001	4097	0x98111001	RTERR_BADTIMERPERIODS	Timer value is not valid.
0x1002	4098	0x98111002	RTERR_INVALIDTASKPTR	Task pointer has the invalid value 0 (zero).
0x1003	4099	0x98111003	RTERR_INVALIDSTACKPTR	Stack pointer has the invalid value 0 (zero).
0x1004	4100	0x98111004	RTERR_PRIOEXISTS	The request task priority is already assigned.
0x1005	4101	0x98111005	RTERR_NOMORETCB	No free TCB (Task Control Block) available. The maximum number of TCBs is 64.
0x1006	4102	0x98111006	RTERR_NOMORESEMAS	No free semaphores available. The maximum number of semaphores is 64.
0x1007	4103	0x98111007	RTERR_NOMOREQUEUES	No free space available in the queue. The maximum number of positions in the queue is 64.
0x100D	4109	0x9811100D	RTERR_EXTIRQALREADYDEF	An external synchronization interrupt is already applied.
0x100E	4110	0x9811100E	RTERR_EXTIRQNOTDEF	No external sync interrupt applied.

Hex	Dec	HRESULT	Name	Description
0x100F	4111	0x9811100F	RTERR_EXTIRQINSTALLFAILED	Application of the external synchronization interrupt has failed.
0x1010	4112	0x98111010	RTERR_IRQLNOTLESSOREQUAL	Call of a service function in the wrong context
0x1017	4119	0x98111017	RTERR_VMXNOTSUPPORTED	Intel® VT-x extension is not supported.
0x1018	4120	0x98111018	RTERR_VMXDISABLED	Intel® VT-x extension is not enabled in the BIOS.
0x1019	4121	0x98111019	RTERR_VMXCONTROLSMISSING	Missing function in Intel® VT-x extension.
0x101A	4122	0x9811101A	RTERR_VMXENABLEFAILS	Activation of Intel® VT-x fails.

Specific positive HRESULT Return Codes:

HRESULT	Name	Description
0x0000_0000	S_OK	No error.
0x0000_0001	S_FALSE	No error. Example: successful processing, but with a negative or incomplete result.
0x0000_0203	S_PENDING	No error. Example: successful processing, but no result is available yet.
0x0000_0256	S_WATCHDOG_TIMEOUT	No error. Example: successful processing, but a timeout occurred.

TCP Winsock error codes

Hex	Dec	Name	Description
0x274C	10060	WSAETIMEDOUT	A connection timeout has occurred - error while establishing the connection, because the remote terminal did not respond properly after a certain period of time, or the established connection could not be maintained because the connected host did not respond.
0x274D	10061	WSAECONNREFUSED	Connection refused - no connection could be established because the target computer has explicitly rejected it. This error usually results from an attempt to connect to a service that is inactive on the external host, that is, a service for which no server application is running.
0x2751	10065	WSAEHOSTUNREACH	No route to host - a socket operation referred to an unavailable host.
More Winsock error codes: Win32 error codes			

6.2 Support and Service

Beckhoff and their partners around the world offer comprehensive support and service, making available fast and competent assistance with all questions related to Beckhoff products and system solutions.

Download finder

Our download finder contains all the files that we offer you for downloading. You will find application reports, technical documentation, technical drawings, configuration files and much more.

The downloads are available in various formats.

Beckhoff's branch offices and representatives

Please contact your Beckhoff branch office or representative for local support and service on Beckhoff products!

The addresses of Beckhoff's branch offices and representatives round the world can be found on our internet page: www.beckhoff.com

You will also find further documentation for Beckhoff components there.

Beckhoff Support

Support offers you comprehensive technical assistance, helping you not only with the application of individual Beckhoff products, but also with other, wide-ranging services:

- support
- design, programming and commissioning of complex automation systems
- and extensive training program for Beckhoff system components

Hotline:	+49 5246 963-157
e-mail:	support@beckhoff.com

Beckhoff Service

The Beckhoff Service Center supports you in all matters of after-sales service:

- on-site service
- repair service
- spare parts service
- hotline service

Hotline:	+49 5246 963-460
e-mail:	service@beckhoff.com

Beckhoff Headquarters

Beckhoff Automation GmbH & Co. KG

Huelshorstweg 20
33415 Verl
Germany

Phone:	+49 5246 963-0
e-mail:	info@beckhoff.com
web:	www.beckhoff.com

Trademark statements

Beckhoff®, ATRO®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, MX-System®, Safety over EtherCAT®, TC/BSD®, TwinCAT®, TwinCAT/BSD®, TwinSAFE®, XFC®, XPlanar® and XTS® are registered and licensed trademarks of Beckhoff Automation GmbH.

Third-party trademark statements

Arm, Arm9 and Cortex are trademarks or registered trademarks of Arm Limited (or its subsidiaries or affiliates) in the US and/or elsewhere.

Intel, the Intel logo, Intel Core, Xeon, Intel Atom, Celeron and Pentium are trademarks of Intel Corporation or its subsidiaries.

Microsoft, Microsoft Azure, Microsoft Edge, PowerShell, Visual Studio, Windows and Xbox are trademarks of the Microsoft group of companies.

More Information:
www.beckhoff.com/tf6350

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