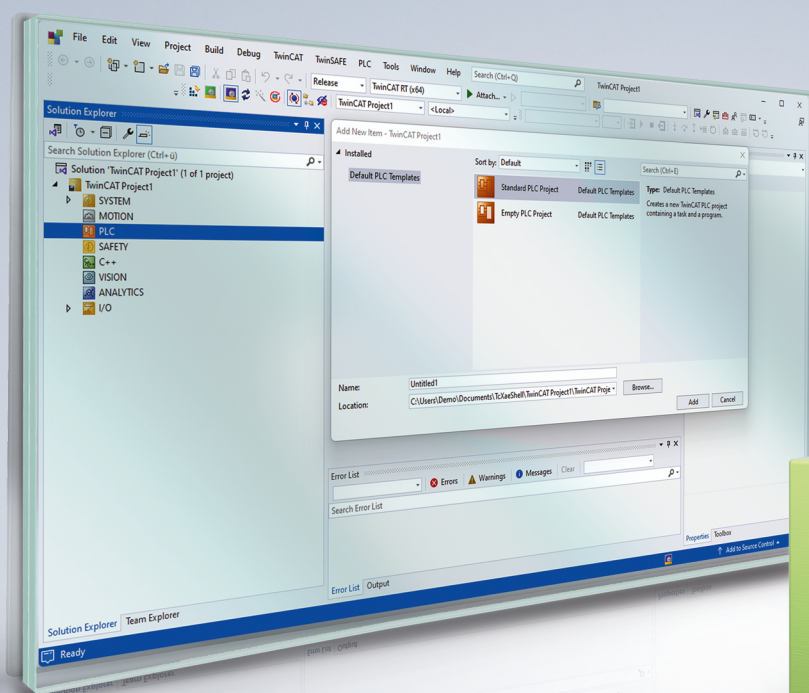


手册 | ZH

TE1000

TwinCAT 3 | ADS over MQTT



目录

1 前言 5

1.1 文档说明 5

1.2 安全信息 5

1.3 信息安全说明 6

2 概述 7

3 安装 9

3.1 系统要求 9

3.2 安装 9

4 技术简介 10

4.1 快速入门 10

4.2 虚拟网络 11

4.3 通信流 12

4.4 配置文件 13

4.5 启用/禁用 15

4.6 安全 16

4.6.1 Mosquitto 插件 16

4.6.2 Mosquitto ACL 17

5 示例 19

1 前言

1.1 文档说明

本说明仅适用于熟悉国家标准且经过培训的控制和自动化工程专家。
在安装和调试组件时，必须遵循文档和以下说明及解释。
操作人员应具备相关资质，并始终使用最新的生效文档。

相关负责人员必须确保所述产品的应用或使用符合所有安全要求，包括所有相关法律、法规、准则和标准。

免责声明

本文档经过精心准备。然而，所述产品正在不断开发中。
我们保留随时修改和更改本文档的权利，恕不另行通知。
不得依据本文档中的数据、图表和说明对已供货产品的修改提出赔偿。

商标

Beckhoff®、ATRO®、EtherCAT®、EtherCAT G®、EtherCAT G10®、EtherCAT P®、MX-System®、Safety over EtherCAT®、TC/BSD®、TwinCAT®、TwinCAT/BSD®、TwinSAFE®、XFC®、XPlanar® 和 XTS® 是 Beckhoff Automation GmbH 的注册商标并由其授权使用。本出版物中所使用的其它名称可能是商标名称，任何第三方出于其自身目的使用它们可能会侵犯商标所有者的权利。



EtherCAT® 是注册商标和专利技术，由 Beckhoff Automation GmbH 授权使用。

版权所有

© Beckhoff Automation GmbH。
未经明确授权，不得复制、分发、使用和传播本文档内容。
违者将被追究赔偿责任。Beckhoff Automation GmbH 保留所有发明、实用新型和外观设计专利权。

第三方商标

本文档可能使用了第三方商标。有关商标信息，可以访问：<https://www.beckhoff.com/trademarks>。

1.2 安全信息

安全规范

为了确保您的使用安全，请务必仔细阅读
并遵守本文档中每个产品的安全使用说明。

责任免除

所有组件在供货时都配有适合应用的特定硬件和软件配置。严禁未按文档所述修改硬件或软件配置，否则，德国倍福自动化有限公司对由此产生的后果不承担责任。

人员资格

本说明仅供熟悉适用国家标准的控制、自动化和驱动工程专家使用。

警示性词语

文档中使用的警示信号词分类如下。为避免人身伤害和财产损失，请阅读并遵守安全和警告注意事项。

人身伤害警告

⚠ 危险

存在死亡或重伤的高度风险。

⚠ 警告

存在死亡或重伤的中度风险。

⚠ 谨慎

存在可能导致中度或轻度伤害的低度风险。

财产或环境损害警告

注意

可能会损坏环境、设备或数据。

操作产品的信息



这些信息包括：
有关产品的操作、帮助或进一步信息的建议。

1.3 信息安全说明

Beckhoff Automation GmbH & Co.KG (简称 Beckhoff) 的产品，只要可以在线访问，都配备了安全功能，支持工厂、系统、机器和网络的安全运行。尽管配备了安全功能，但为了保护相应的工厂、系统、机器和网络免受网络威胁，必须建立、实施和不断更新整个操作安全概念。Beckhoff 所销售的产品只是整个安全概念的一部分。客户有责任防止第三方未经授权访问其设备、系统、机器和网络。它们只有在采取了适当的保护措施的情况下，方可与公司网络或互联网连接。

此外，还应遵守 Beckhoff 关于采取适当保护措施的建议。关于信息安全和工业安全的更多信息，请访问本公司网站 <https://www.beckhoff.com/secguide>。

Beckhoff 的产品和解决方案持续进行改进。这也适用于安全功能。鉴于持续进行改进，Beckhoff 明确建议始终保持产品的最新状态，并在产品更新可用后马上进行安装。使用过时的或不支持的产品版本可能会增加网络威胁的风险。

如需了解 Beckhoff 产品信息安全的信息，请订阅 <https://www.beckhoff.com/secinfo> 上的 RSS 源。

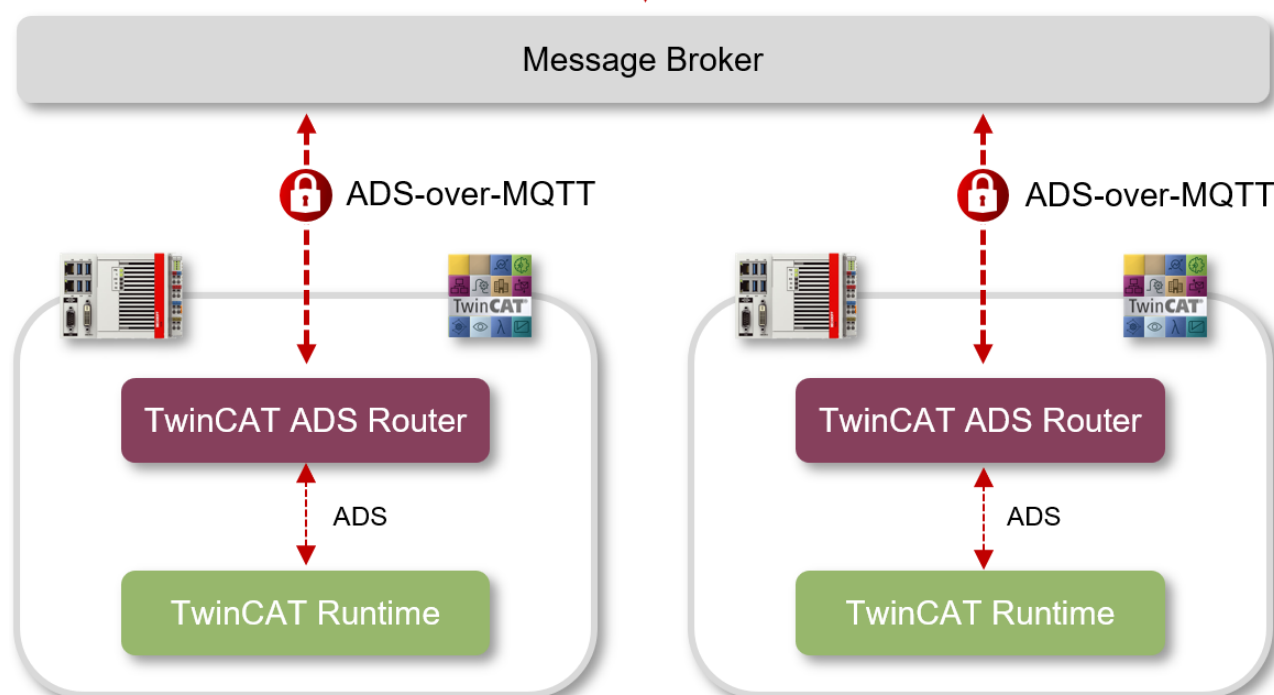
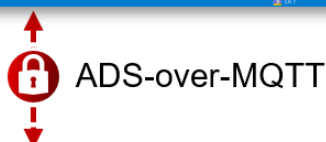
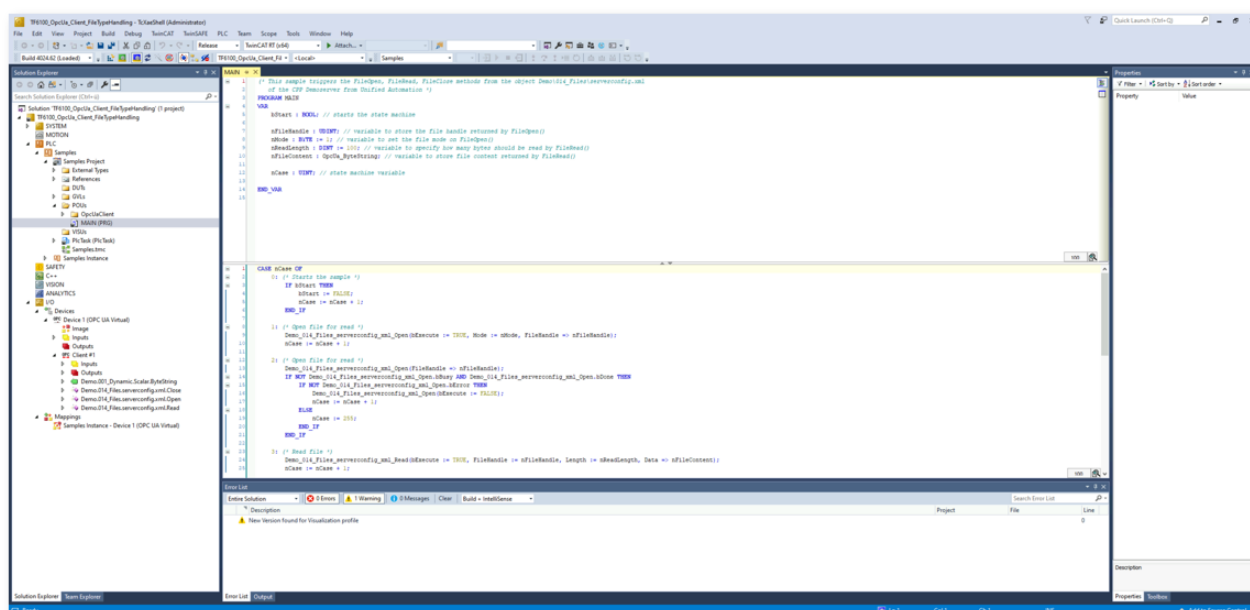
2 概述

倍福 ADS（自动化设备规范）是倍福为工业自动化系统中的高效数据交换开发的通信协议。它是将设备和软件集成到倍福基于 PC 的控制技术中的重要组成部分。

从 ADS 协议的角度来看，ADS-over-MQTT 是一个额外的传输通道，可用于传输 ADS。通过 MQTT 消息代理对通信进行解耦可带来许多优势，尤其是在集成其他 ADS 应用程序时，其扩展性和灵活性方面表现尤为突出。在传输层可采用 TLS 等安全机制来确保通信连接的安全性。

通过 ADS-over-MQTT，整个数据交换对 ADS 应用程序而言是透明的，因为 ADS 路由器只需感知并维护 MQTT 传输通道的相关信息。特别是，这也使现有应用程序可以更轻松地实现改造升级。

ADS-over-MQTT 的主要典型用例是远程维护和远程诊断场景，即 TwinCAT 工程环境（TwinCAT XAE）需要与一个或多个控制器连接，以进行远程调试。下图说明了此处创建的架构。



不过，ADS-over-MQTT 还有许多其他用例，尤其在整合多个分布式 PLC 系统时更具优势。

本文件概述了使用可能性，并从技术上说明了如何通过 MQTT 消息代理配置“虚拟 ADS 网络”。

基于 MQTT 的 ADS 网络的优势

- **子网、基于 NAT 的网络和防火墙：**

在传统 ADS 设置中，传入 TCP/IP 连接在两个方向上均被使用。因此，在正常情况下，设备必须定位在同一网络中。在具有不同子网的分布式系统中，这需要复杂配置，以便使 ADS 路由可用。在基于 MQTT 的 ADS 网络中，设备只使用一个传出 TCP/IP 连接。这样，上层网络中的代理就可以在所有设备之间进行代理。由于传出连接，因此可以使用一般防火墙，无需注册传入端口。

- **访问控制：**

创建适当的路由后，就可以在传统 ADS 设置中执行双向通信。设备 A 对设备 B 的访问也允许设备 B 访问设备 A。基于 MQTT 的 ADS 网络可以配置为设备 A 可以访问 B，但不能反向配置。

- **安全/加密：**

TwinCAT 与代理之间的通信可以通过 TLS（使用证书或 PreSharedKey (PSK)）加密。在这种情况下，传输 MQTT 协议是加密的，因此 ADS 协议可以在载荷中以不加密的形式传输。

- **改造：**

ADS-over-MQTT 对 ADS 应用程序而言是透明的，这意味着无需对其进行更改。

注意

ADS 访问意味着完全访问

正如安全公告 2017-01 中所述，ADS 提供对设备的完全访问。

安全 ADS 为通信提供授权和加密；因此，它代表了一种传输加密。因此，如果存在 ADS 路由，那么就存在完全访问。

OPC UA 等解决方案可提供专用的、与角色相关的单个文件访问权限。

3 安装

3.1 系统要求

以下系统要求适用于本产品的安装和操作。

技术数据	描述
操作系统	Windows 10 Windows Server 2022 TwinCAT/BSD TwinCAT/Linux®
目标平台	PC 架构 (x86、x64、Arm®)
TwinCAT 版本	TwinCAT 3 (版本 4022.0 及以上)
TwinCAT 安装级别	TwinCAT 3 XAE、XAR、ADS
所需 TwinCAT 授权	---
支持 TwinCAT 3 Usermode Runtime	是, 另请参见 配置文件 [► 13]

● MQTT 消息代理



要使用该产品, 需要一个 MQTT 消息代理, 通过它可以建立连接。任何 MQTT 消息代理均可使用, 如 Mosquitto 或 HiveMQ。此外, 还可以使用托管云服务, 如 AWS IoT Core。

● Mosquitto 消息代理插件



目前, 所提供的 Mosquitto 消息代理插件仅适用于 Windows 操作系统。

3.2 安装

ADS-over-MQTT 功能是 TwinCAT 基础安装的固定组成部分, 无论是 XAE、XAR 还是 ADS 安装环境均默认包含, 无需在 TwinCAT 端进一步安装。要安装 MQTT 消息代理, 请查阅您的消息代理软件文档。

4 技术简介

4.1 快速入门

本文档文章旨在帮助您快速入门，初步了解如何使用本产品。按照以下步骤操作，即可建立与 MQTT 消息代理的 ADS-over-MQTT 连接，并发送和接收 ADS 消息。

● 安装消息代理



本文档假设您已在本地安装并运行 MQTT 消息代理。例如，我们在此处使用的是 Mosquitto 消息代理，不过，您可以使用任何消息代理。

概述

为了进行功能演示，两个 TwinCAT 设备要与消息代理建立 ADS-over-MQTT 连接。然后，我们在第一个设备上使用 TwinCAT XAE（工程环境），通过 ADS-over-MQTT 路由与第二个系统建立连接。为了使安装方案尽可能简单，两个设备应处在同一网络上。

设备 1:

- Mosquitto 消息代理
- TwinCAT 3 XAE

设备 2:

- TwinCAT 3 XAE 或 XAR

请注意设备 1 的 IP 地址或主机名。在后一种情况下，请确保您的网络中的域名解析功能正常。

消息代理设置

自 2.x 版本起，Mosquitto 消息代理默认提供的配置需设置安全措施，以确保消息代理的安全运行。在下文中，我们将向您展示如何修改 Mosquitto 消息代理配置，以便与该代理建立非安全通信连接。不过，此操作仅限在可信操作环境中用于测试目的。对于生产环境操作，我们建议采用安全的消息代理配置。

1. 在您的系统上安装 Mosquitto 消息代理。
 2. 备份 Mosquitto 安装目录中的 *mosquitto.conf* 文件。它通常位于 *C:\Program Files\mosquitto* 下。
 3. 使用您所选的文本编辑器打开 *mosquitto.conf* 文件，并删除现有内容。添加以下内容，并保存文件。


```
listener 1883
allow_anonymous true
```
 4. 通过相应的 Windows 服务重启 Mosquitto 消息代理，或者通过控制台或 *mosquitto.exe* 进行手动重启。
- ⇒ 现在，您已将 Mosquitto 消息代理配置为在端口 1883 上监听传入的客户端连接，并且无需任何安全措施（既不需要用户身份验证，也不需要客户端证书）。

● 设备 1 上的 Windows 防火墙



请在设备 1 的 Windows 防火墙中启用标准 MQTT 端口 1883/tcp 作为传入端口，以便设备 2 可以与消息代理建立连接。

现在，您可以开始配置 TwinCAT。

配置第一个 TwinCAT 设备

现在，该设备上的 TwinCAT ADS 路由器应建立一条通往本地消息代理的路由。创建一个新的 XML 文件，例如使用记事本进行创建，并插入以下内容。

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<TcConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://www.beckhoff.com/schemas/2015/12/TcConfig">
<RemoteConnections>
  <Mqtt>
    <Address Port="1883">127.0.0.1</Address>
    <Topic>VirtualAmsNetwork1</Topic>
```

```
</Mqtt>
</RemoteConnections>
</TcConfig>
```

将此 XML 文件以任何名称保存在以下目录中，并重新启动 TwinCAT，以便更改生效。

\\TwinCAT\\3.1\\Target\\Routes

配置第二个 TwinCAT 设备

该设备上的 TwinCAT ADS 路由器应建立一条通往设备 1 消息代理的路由。创建一个新的 XML 文件，例如使用记事本进行创建，并插入以下内容。

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<TcConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://www.beckhoff.com/schemas/2015/12/TcConfig">
<RemoteConnections>
  <Mqtt>
    <Address Port="1883">%IPAddress%</Address>
    <Topic>VirtualAmsNetwork1</Topic>
  </Mqtt>
</RemoteConnections>
</TcConfig>
```

将 %IPAddress% 条目替换为设备 1 的 IP 地址或主机名。将此 XML 文件以任何名称保存在以下目录中，并重新启动 TwinCAT，以便更改生效。

\\TwinCAT\\3.1\\Target\\Routes

连接工程环境

现在，在设备 1 上启动 TwinCAT XAE，并创建一个新的 TwinCAT 项目。或者，您也可以打开一个现有项目。在本教程中，我们只需使用工具栏与远程 ADS 设备建立连接。

现在，您可以在工具栏中找到 TwinCAT 设备 2，并与其建立连接。

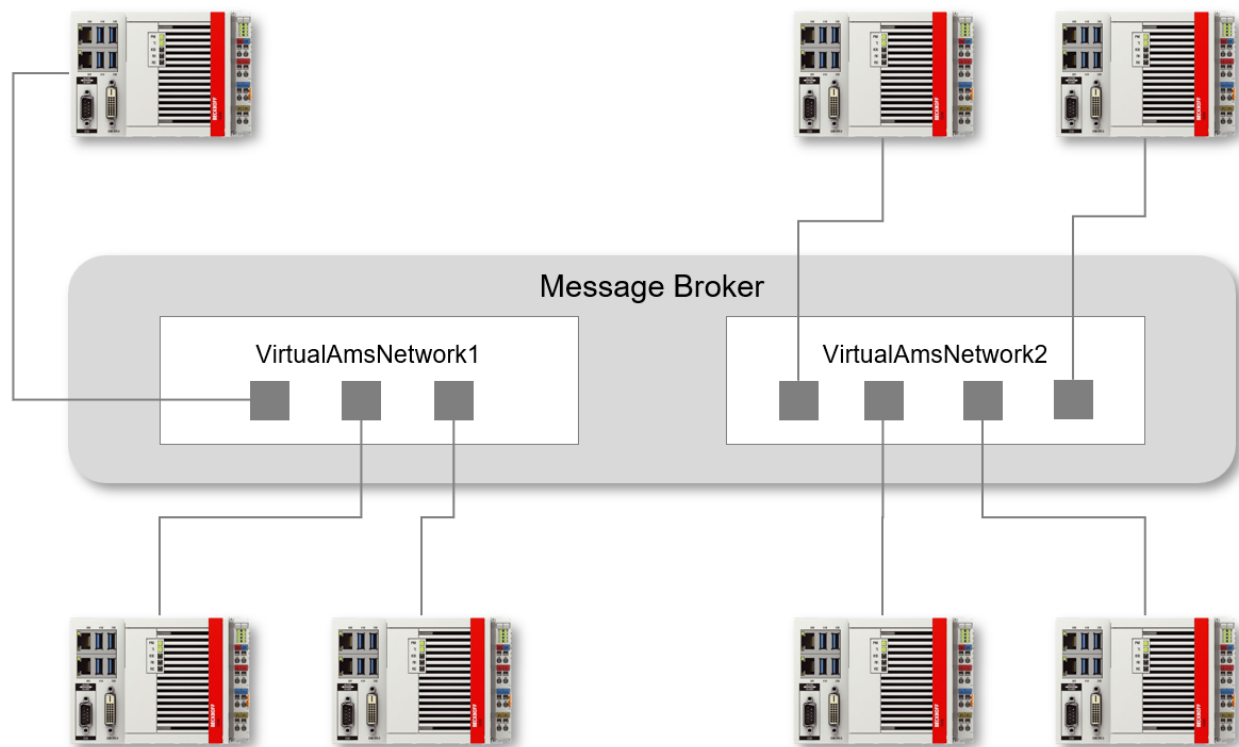


后续步骤

在成功建立与消息代理的 ADS-over-MQTT 连接后，我们建议您将 Mosquitto 消息代理重置为默认设置。为此，请备份您在前面步骤中创建的 *mosquitto.conf* 文件。该文件与代理文档共同构成了后续操作的良好基础，例如，可通过配置客户端/服务器证书和用户身份验证，建立安全的消息代理运行环境。

4.2 虚拟网络

在配置 ADS-over-MQTT 时，可以定义所谓的“虚拟网络”。这样可以将 ADS 设备部署到独立的网络中。只有同一网络内的设备才能相互通信。在 MQTT 层面上，这种分发部署以共同的基本主题为基础。



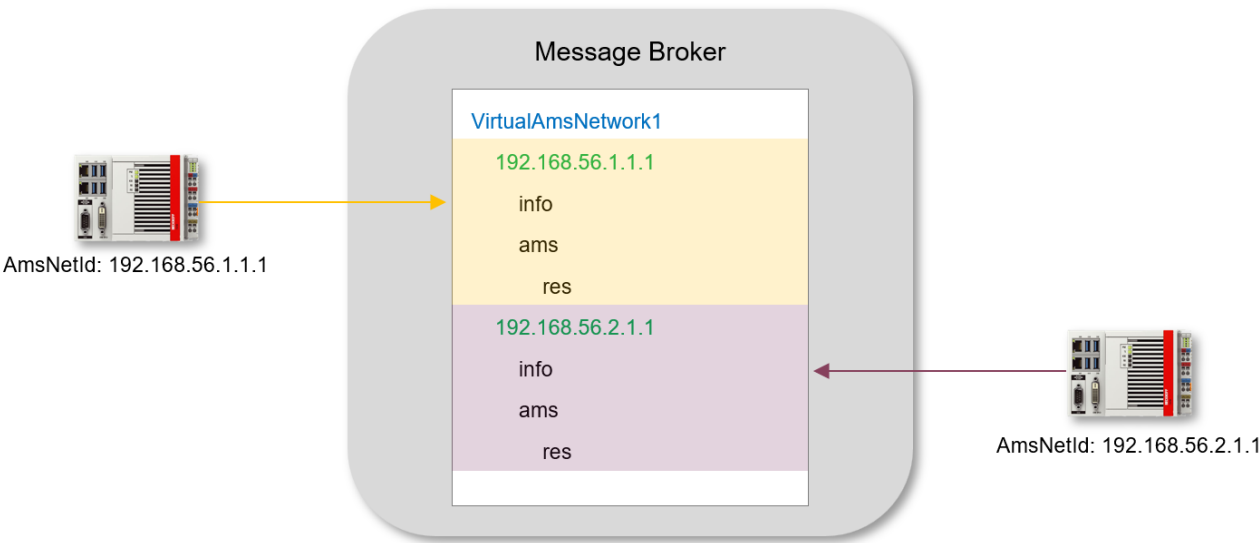
使用 ADS-over-MQTT 配置文件 [► 13] 在客户端层面可以定义这些虚拟网络，如有需要，还可使用 Mosquitto 消息代理的 `TcMqttPlugin` [► 16] 分配访问权限。

4.3 通信流

为了实现通过 ADS-over-MQTT 的数据交换与设备发现功能，所有 ADS 设备需在消息代理上采用统一的主题结构。主题结构取决于配置的虚拟网络 [► 11] 名称和相应设备的 AMS Net ID。

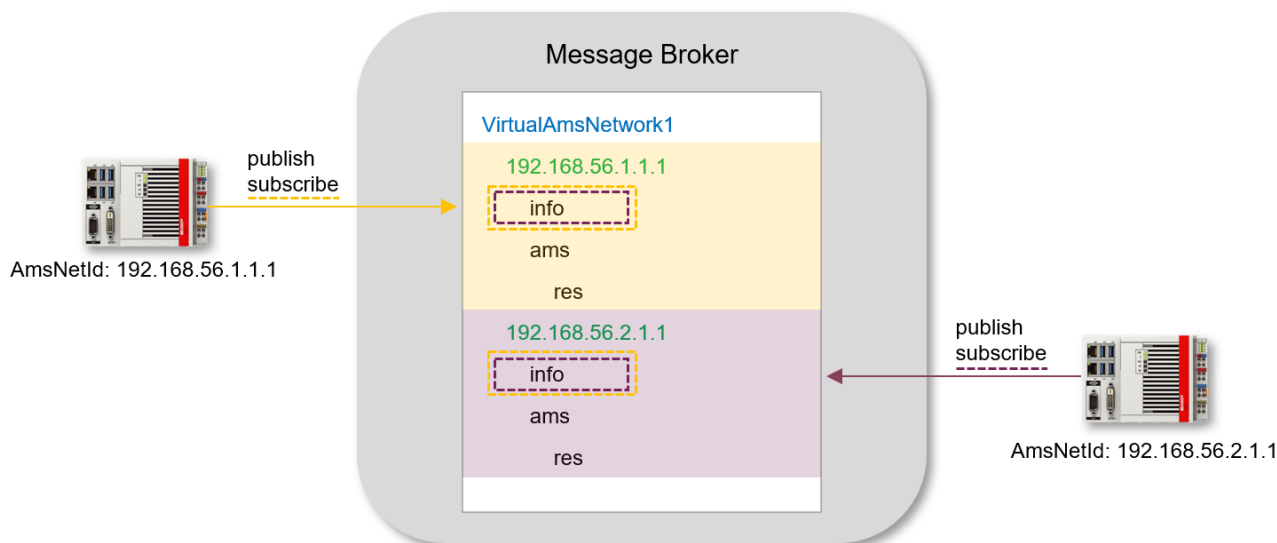
类型	主题
发现	<code><NetworkName>/<AmsNetId>/info</code>
通信	<code><NetworkName>/<AmsNetId>/ams</code> <code><NetworkName>/<AmsNetId>/ams/res</code>

因此，每个 ADS 设备在消息代理上都有自己的“主题区域”。下图说明了这种关系。



发现

连接的 TwinCAT ADS 路由器向其发现主题发送包含设备信息的保留消息，同时订阅主题 <NetworkName>/+/info，以便了解所有其他连接路由器的情况。



发送至发现主题的消息包含一个 XML 结构，其中包含设备信息，例如，主机名和使用的 TwinCAT 版本：

```
<info>
  <online name="EC2AMAZ-2RRSQS6" osVersion="10.0.20348" osPlatform="2"
  tcVersion="3.1.4026.10">true</online>
</info>
```

如果消息代理不支持保留消息，则可在 ADS-over-MQTT 配置文件 [► 13] 中加以考虑。在这种情况下将执行通信握手而非保留消息：新连接的设备会向自身发现主题注册，所有其他连接设备则通过各自的发现主题响应其他消息。这样可以确保即使使用不支持保留消息的消息代理，设备也能找到彼此。其劣势在于，在需要频繁重连的大型操作环境中，消息量会显著增加。

通信

TwinCAT ADS 路由器在建立连接后会立即订阅其通信主题（<NetworkName>/<AmsNetId>/ams/#）。发送至该路由器的 ADS 命令将传输至 <NetworkName>/<AmsNetId>/ams，而响应则通过 <NetworkName>/<AmsNetId>/ams/res 主题接收。

4.4 配置文件

TwinCAT ADS 路由器通过 XML 文件进行配置，以便与消息代理建立 ADS-over-MQTT 连接。该配置文件以任何名称存储在以下目录中：

```
\\TwinCAT\\3.1\\Target\\Routes
```

在示例 [► 19] 一章中，您可以找到适用于下述所有用例的示例配置文件。

● 激活新的 ADS-over-MQTT 配置

1

只有在 TwinCAT ADS 路由器初始化时，新的或更改的 ADS-over-MQTT 配置才会被采用。例如，在 TwinCAT 从 RUN 到 CONFIG 或 CONFIG 到 CONFIG 的转换过程中就会发生这种情况。

● 路径信息

1

请确保配置文件中的任何路径详情均使用了操作系统的正确拼写。

● TwinCAT 3 Usermode Runtime



您可以在 TwinCAT 3 Usermode Runtime 中操作 ADS-over-MQTT。您只需考虑 Usermode Runtime 的基本路径。以交付状态为例，路由器配置文件的基本目录定义如下：

%ProgramData%\Beckhoff\TwinCAT\3.1\Runtimes\UmRT_Default\3.1\Target\Routes\

除此之外，ADS-over-MQTT 特定主题的其他路径详情（参见 技术简介 [▶ 10]）将予以保留。

基本配置

基本配置始终包含以下要素：

消息代理的地址、其 TCP 端口和虚拟网络 [▶ 11] 的名称。

通过 <Address> 节点可以指定代理的地址及其可以访问的 TCP 端口。虚拟网络通过 <Topic> 节点进行定义。在下面的示例中，与本地（127.0.0.1）安装的消息代理和 TCP 端口 1883 建立了连接。

“VirtualAmsNetwork1” 被定义为虚拟网络。

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<TcConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://www.beckhoff.com/schemas/2015/12/TcConfig">
  <RemoteConnections>
    <Mqtt>
      <Address Port="1883">127.0.0.1</Address>
      <Topic>VirtualAmsNetwork1</Topic>
    </Mqtt>
  </RemoteConnections>
</TcConfig>
```

NoRetain

设备发现的通信流 [▶ 12] 可以通过 NoRetain 属性进行自定义。当设置 NoRetain = true 时，设备搜索功能不再基于保留消息。而是采用握手机制来识别所有连接的 ADS 设备。

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<TcConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://www.beckhoff.com/schemas/2015/12/TcConfig">
  <RemoteConnections>
    <Mqtt NoRetain="true">
      <Address Port="1883">mybroker.someurl.com</Address>
      <Topic>VirtualAmsNetwork1</Topic>
    </Mqtt>
  </RemoteConnections>
</TcConfig>
```

单向

通过单向属性可以阻止传入的 ADS 消息进入本系统。一个典型的用例可以是工程系统，例如，它应该能够通过 ADS 连接到运行时系统，但反之则不成立。

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<TcConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://www.beckhoff.com/schemas/2015/12/TcConfig">
  <RemoteConnections>
    <Mqtt Unidirectional="true">
      <Address Port="1883">mybroker.someurl.com</Address>
      <Topic>VirtualAmsNetwork1</Topic>
    </Mqtt>
  </RemoteConnections>
</TcConfig>
```

TLS

<TLS> 节点可用于定义通过 TLS 确保传输通道安全的设置。可提供各种连接选项，例如，配置客户端证书或 PSK。我们的示例 [▶ 19] 介绍了所有可能的配置方案。

TLS IgnoreCn

IgnoreCn 属性可用于禁止验证服务器证书中的 CommonName (CN)。

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<TcConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://www.beckhoff.com/schemas/2015/12/TcConfig">
<RemoteConnections>
  <Mqtt>
    <Address Port="8883">mybroker.someurl.com</Address>
    <Topic>VirtualAmsNetwork1</Topic>
    <Tls IgnoreCn="false">
      <Ca>C:\TwinCAT\3.1\Target\Certificates\mybroker\CA.pem</Ca>
    </Tls>
  </Mqtt>
</RemoteConnections>
</TcConfig>
```

TLS PSK

使用带有预共享密钥（PSK）的 TLS 时，您可以将 PSK 指定为十六进制编码的 64 个字符字符串，也可以使用 Sha256（Identity + Pwd）在 TwinCAT 内部进行转换。在后一种情况下，您可以使用 IdentityCaseSensitive 属性来指定 TwinCAT 在计算时应将身份转换为 UpperCase。我们的示例 [► 19] 介绍了两种可能的配置方案。

用户

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<TcConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://www.beckhoff.com/schemas/2015/12/TcConfig">
  <RemoteConnections>
    <Mqtt>
      <Address Port="1883">127.0.0.1</Address>
      <Topic>VirtualAmsNetwork1</Topic>
      <User>CX-12345</User>
    </Mqtt>
  </RemoteConnections>
</TcConfig>
```

<User> 元素指定了一个身份，可在 TcMqttPlugin [► 16] 中用于配置 ADS 设备之间的访问权限。不过，身份通常是通过其他方式定义的，例如，客户端证书的 CommonName（CN）。

作为选项，<Mqtt> 元素可以包含一个 ClientId 属性，以指定 MQTT ClientID。否则，它将由 <User> 和任意字符串组成。

4.5 启用/禁用

您可以在运行时启用或禁用 ADS-over-MQTT 路由，而无需重新启动 TwinCAT 系统。例如，您可以仅在检修时启用路由，并将此功能分配给按键开关，或通过 HMI 将其与用户输入连接起来。

通过 TwinCAT 系统服务中的 ADS 接口执行启用/禁用功能。您可以在路由配置文件中使用的特殊属性来定义在启动 TwinCAT 后路由应该具有的默认状态。

路由配置文件

在 ADS-over-MQTT 路由的配置文件中，您可以指定在启动 TwinCAT 后路由应该具有的默认状态。该属性（“已禁用”）在 <Mqtt> 节点中指定。如果赋值为“true”，则启动 TwinCAT 后不会自动建立 ADS-over-MQTT 路由，必须通过 ADS 接口才能建立该路由。

```
<?xml version="1.0"?>
<TcConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://www.beckhoff.com/schemas/2015/12/TcConfig">
  <RemoteConnections>
    <Mqtt Disabled="true">
      <Name>MyBroker</Name>
      <Address Port="1883">myMessageBrokerAddress</Address>
      <Topic>VirtualAmsNetwork1</Topic>
    </Mqtt>
  </RemoteConnections>
</TcConfig>
```

ADS 接口

ADS 接口定义如下。

```
ADS-Kommando: AdsWriteRequest
ADS-Port: 10000
IndexGroup: 808
IndexOffset: siehe unten
Data: Identifizierung der Route, siehe unten
```

IndexOffset 指明您想要永久还是暂时启用/禁用某个路由。在这种情况下，“永久”意味着即使在 TwinCAT 重新启动后也会保留启用/禁用状态。提供以下选项：

IndexOffset	名称	描述
1	ADS_ROUTE_DISABLE	永久禁用路由。
2	ADS_ROUTE_ENABLE	永久启用路由。
3	ADS_ROUTE_DISABLE_TMP	暂时禁用路由。
4	ADS_ROUTE_ENABLE_TMP	暂时启用路由。

AdsWriteRequest 的数据区指定了要寻址的 ADS-over-MQTT 路由。这是通过在字符串中定义来实现的。格式如下，相当于配置文件的 <Address> 和 <Topic> 节点的组合。此处固定了关键词“MQTT”。

```
MQTT:<Address>:<Topic>
```

或者，您也可以在配置文件中添加 <name> 节点，然后使用该节点进行引用。此处使用的名称在所有 ADS 路由中必须是唯一的。在上述示例中，您可以在 AdsWriteRequest 的数据区中使用以下字符串：

```
MQTT:MyBroker
```

PLC 示例

相应的示例 [► 19] 可供下载。

4.6 安全

确保通信安全有多种选择。为此，可以使用基于 X.509 证书的 TLS 连接或预共享密钥（PSK）。建议使用 TLS 确保通信安全，尤其是通过不可信网络（如互联网）进行通信时。在 配置文件 [► 13] 和 示例 [► 19] 两章中，您可以找到通过 TLS 运行 ADS-over-MQTT 的说明和示例配置文件。

代理本身必须在可信环境中运行，这是因为代理上的所有消息都没有进行加密或者安全保护。



入侵虚拟 ADS 网络

即使设备和代理之间的通信是通过 TLS 加密进行的，设备之间也没有安全保障。ADS 命令以未加密形式存在于代理上。

如果设备被入侵，攻击者可以通过获得的权限执行所有 ADS 命令。这些命令还包括文件读取操作或启动进程的操作。

采用两种方法可以在单独的 ADS 设备之间配置访问权限：

- 通过访问控制列表 [► 17] 配置访问权限（以 Mosquitto 为例）
- 通过插件 [► 16] 配置访问权限（仅适用于 Mosquitto）

4.6.1 Mosquitto 插件

一个专门为 Mosquitto 消息代理开发的插件，以便在各个 TwinCAT ADS 路由器之间定义访问权限。另请注意操作该插件的系统要求 [► 9]。

该插件随 TwinCAT 3.1 版本 4024 下的 TwinCAT 安装一起交付。版本 4026 需要安装相应的软件包（TwinCAT.XAE.MqttPlugin）。该插件安装在以下目录中，可在 Mosquitto 配置中引用。

```
\TwinCAT\AdsApi\TcMqttPlugin
```

该插件有 32 位和 64 位版本，具体取决于您使用的 Mosquitto 消息代理的版本。然后，该插件在 Mosquitto 消息代理配置中的集成方式如下：

```
auth_plugin <Path>TcMqttPlugin.dll
auth_opt_xml_file <Path>MyACL.xml
```

MyACL.xml 文件包含对代理本身的访问配置，以及连接的 TwinCAT ADS 路由器之间的通信配置。该配置将在下一章节中进行详细介绍。

配置

插件提供配置虚拟 AMS 网络的选项。为此，要为每个目标设备指定哪个设备可以访问哪个其他设备。与传统的 ADS 路由不同，这些连接是定向的：因此，目标也无权访问源。

```
<TcMqttAclConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="C:\TwinCAT\3.1\Config\Modules\TcMqttAclConfig.xsd">
  <Ams>
    <Topic>VirtualAmsNetwork1</Topic>
    <User>
      <Name>EngineeringStation</Name>
    </User>
    <User>
      <Name>CX-12345</Name>
      <Access>EngineeringStation</Access>
    </User>
    <User>
      <Name>CX-56789</Name>
      <Access>EngineeringStation</Access>
    </User>
  </Ams>
</TcMqttAclConfig>
```

Ams 网络的名称在 <Ams> 节点中定义。它用于识别网络使用的 MQTT 主题。单个 <User> 元素描述设备。这些元素有一个 <Name> 属性，用于描述与之建立连接的 MQTT 身份。该身份可以通过各种 TLS 机制进行传输，例如，通过 TLS-PSK 身份标识或客户端证书的 CommonName (CN)。我们此处的[示例 \[► 19\]](#)介绍了可能的配置方案。

具有访问权限的设备通过 <Access> 元素定义。在上述示例中，“EngineeringStation”身份可以访问两个 CX 设备，但 CX 设备既不能访问“EngineeringStation”，也不能相互访问。



配置文件会循环重载，因此无需重新启动代理。

对要注册的 AmsNetId 的限制

通过这种配置，每个有效连接的设备都可以拥有一个任意的 AmsNetId，因此从 ADS 的角度看也就拥有了一个身份。可根据需要进一步限制：

```
<User>
  <Name>CX-56789</Name>
  <Access>EngineeringStation</Access>
  <NetId>192.168.56.1.1.1</NetId>
</User>
```

只要指定了至少一个 NetId，就只能从该列表中注册一个 NetId。另一种解决方案是在客户端证书的 CommonName (CN) 中输入 NetId。

4.6.2 Mosquitto ACL

许多消息代理允许配置访问控制列表 (ACL)，以限制客户端与某些主题的交互。下一章将以 Mosquitto 消息为例介绍此配置。此处描述的授予访问权限的程序与 TcMqttPlugin [\[► 16\]](#) 不同。这指定了哪些其他 ADS 设备可以访问 ADS 设备。而 (Mosquitto) ACL 则恰恰相反，因为此处指定了允许 ADS 设备访问的其他 ADS 设备。

概述

Mosquitto 消息代理允许配置访问控制列表，该列表被定义为一个单独的配置文件，可在代理的主配置中进行引用。该配置条目示例如下：

```
acl_file C:\Program Files\mosquitto\mosquitto.acl
```

您还可以在我们的[示例 \[► 19\]](#)中找到可供下载的完整配置文件。

在 ACL 文件中，您可以定义发布和订阅某些主题的授权，并为每个用户分别指定。用户的访问权限始终通过下面一行引入：

```
user <username>
```

随后，读写权限根据以下方案定义：

```
topic [read|write|readwrite] <topicName>
```

ADS-over-MQTT 的配置

对于 ADS-over-MQTT，必须根据通信流 [► 12] 确保两件事：所有 ADS 设备可以访问发现主题，以及通过通信主题进行发送/接收。

ADS 设备必须始终拥有对自己主题的读取/写入访问权限。设备可接收其他 ADS 设备的发现主题的读取权限。要交换 ADS 消息，ADS 设备必须拥有对设备通信主题的读取/写入访问权限。ADS 设备在消息代理上由自己的身份标识。例如，该身份可以来自 PSK 或与客户端证书的 CommonName (CN) 相对应。下面的配置说明了这些关系。

```
user <identity>
topic readwrite <VirtualAmsNetworkName>/<OwnAmsNetId>/#
topic read <VirtualAmsNetworkName>/+/info
topic readwrite <VirtualAmsNetworkName>/+/ams/#
```

如果要拒绝 ADS 设备访问其他设备，必须确保目标 AmsNetId 的主题没有写入权限。

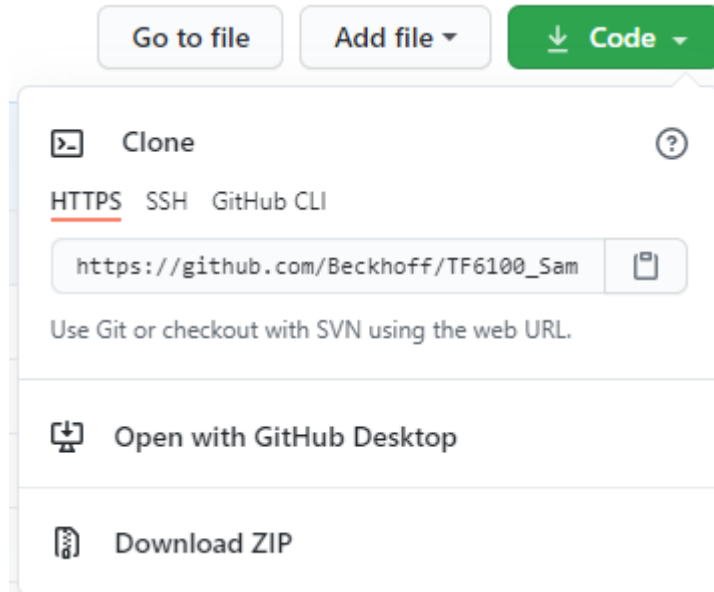
我们熟悉的 TcMqttPlugin 选项，即一个 ADS 设备只能注册一个 AmsNetId，也可以通过 Mosquitto ACL 实现。为此，条目 <OwnAmsNetId> 必须精确替换为预先设定好的唯一的一个 AmsNetId。如果 ADS 设备可以注册任意 AmsNetId，则必须为 <OwnAmsNetId> 设置通配符 (#)。

下面以两个 ADS 设备之间通信的访问权限条目为例：

```
user EngineeringStation
topic readwrite VirtualAmsNetwork1/18.153.78.19.1.1/#
topic read VirtualAmsNetwork1/+/info
topic readwrite VirtualAmsNetwork1/+/ams/#
user CX-12345
topic readwrite VirtualAmsNetwork1/3.120.15.8.1.1/#
topic read VirtualAmsNetwork1/+/info
topic readwrite VirtualAmsNetwork1/+/ams/#
```

5 示例

本产品的示例代码和配置可从 GitHub 上的相应存储库获取：https://github.com/Beckhoff/ADS-over-MQTT_Samples。您可以克隆存储库或下载包含示例的 ZIP 文件。



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.

The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

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

更多信息:
www.beckhoff.com/te1000

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

