BECKHOFF New Automation Technology

Manual | EN

Beckhoff RT Linux®

Operating system

2025-03-11 | Version: 1.0

Table of contents

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®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS®, and XPlanar® 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

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.

1.1 Symbol explanation

The following warnings are used in the documentation. Read and follow the warnings.

Warnings relating to damage to property or the environment:

NOTICE
There is a potential hazard to the environment and equipment.

Notes showing further information or tips:

This notice provides important information that will be of assistance in dealing with the product or software. There is no immediate danger to product, people or environment.

1.2 Documentation issue status

Table 1: Change notes for the documentation.

Version	Changes
1.0	• First version

2 Setup and installation

This chapter provides detailed instructions for installing Beckhoff RT Linux® on an industrial PC. The first section explains the steps for creating a bootable USB stick using the Rufus tool, including the necessary prerequisites. The second section describes the necessary steps to configure the BIOS settings to ensure that the industrial PC can boot from this USB stick. Finally, the third section covers the entire installation process.

2.1 Create bootable USB stick

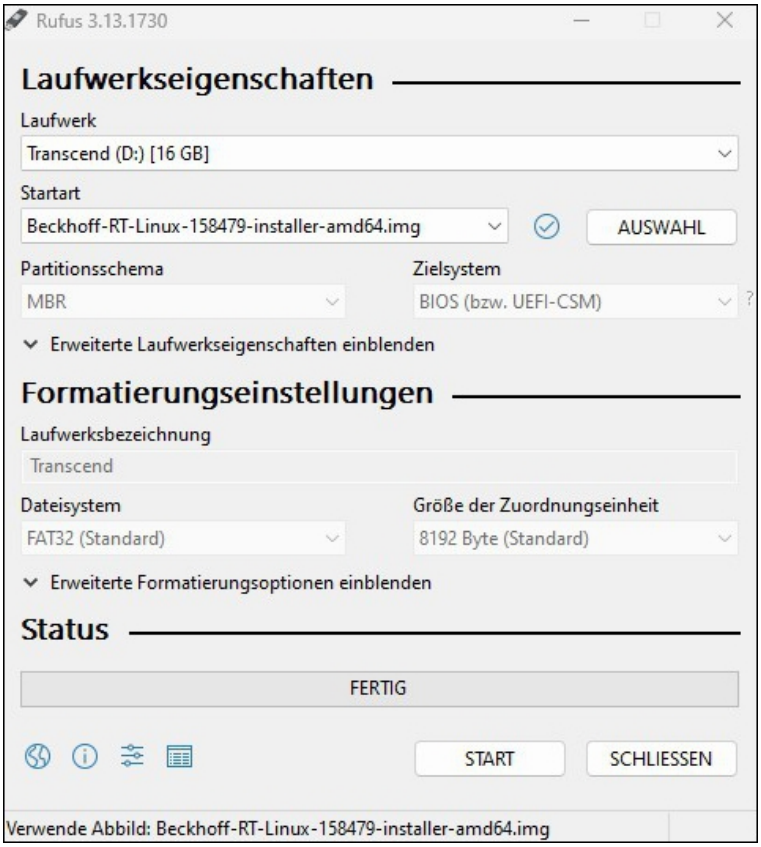
Before you can install Beckhoff RT Linux ® on an industrial PC, you must create a bootable USB stick and upload the current image to the USB stick. Use a flash tool such as Rufus to do this. You can then start the industrial PC from the USB stick and install Beckhoff RT Linux®.

Requirements for this step:

- Download the Rufus tool from https://rufus.ie/
  Notice **:** Note that newer versions of Rufus may not be compatible with hard disk encryption tools. Rufus 3.13 is therefore recommended.
- Image for the installation of Beckhoff RT Linux ® and USB stick with at least 2 GB storage space.

Proceed as follows:

1. Start the Rufus tool on a PC with Windows operating system.
2. Click **Select** and select the image you want to upload to the USB stick.



3. Under **Device**, select a USB stick as the target drive. If only one external drive is connected to your PC, the USB stick is selected automatically.
4. Click **Start** to upload the image to the USB stick.

⇨ The process may take a few minutes. Do not cancel the process until the **Ready** message appears. You have successfully created a bootable USB stick and can install Beckhoff RT Linux® on the industrial PC in the next step.

2.2 Check BIOS settings

Check the BIOS settings to be able to start the industrial PC from the bootable USB stick you created. For Beckhoff RT Linux® the boot mode in the BIOS must be set to UEFI.

Start the BIOS setup and adjust the boot mode if it differs from the settings on your industrial PC.

Proceed as follows:

1. Restart your industrial PC and press **[Del]** to start the BIOS setup.
   The BIOS setup window appears.
2. Set the **UEFI** option under **Boot > Boot mode select**.
3. Press **[F4]** to save the settings and exit the BIOS setup.
   The device is restarted.

⇨ You have successfully configured the BIOS and can install Beckhoff RT Linux® in the next step.

2.3 Installing Beckhoff RT Linux®

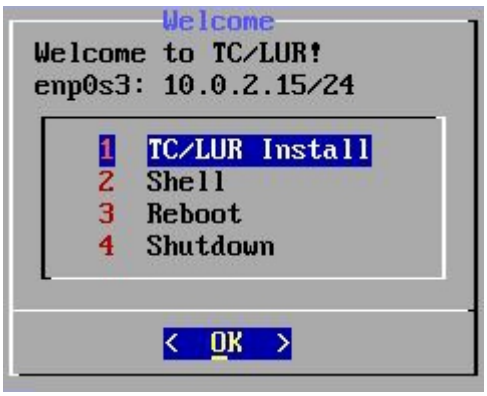
Connect the boot-capable USB stick with Beckhoff RT Linux® image to an industrial PC and start the device.

Prerequisites:

- Bootable USB stick with Beckhoff RT Linux® image.
- Min. 4 GB free space on the memory card**.**

Proceed as follows:

1. Connect the USB stick with Beckhoff RT Linux® image to the industrial PC.
2. Start the industrial PC and press **[F7]** to enter the boot menu.
3. Select the UEFI entry for the USB stick and confirm with **[Enter]**.
   The industrial PC boots from the USB stick and the Beckhoff installer is executed.
4. Select the **TC/LUR Install** option to install Beckhoff RT Linux®.



5. Select the appropriate hard disk as the destination for the installation.

6. Enter a password and follow the further installation instructions.

⇨ Restart the industrial PC. Beckhoff RT Linux® is loaded.

2.4 Industrial PCs with Arm® processors: Update image

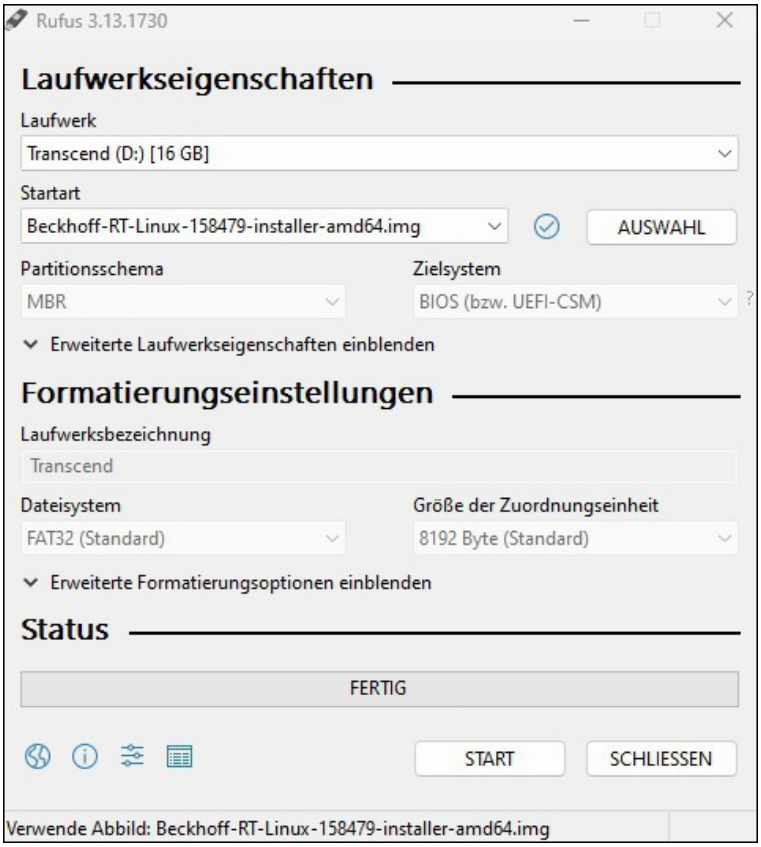
For industrial PCs with Arm® processors, the image is copied directly to the microSD card. Use a flash tool such as Rufus to do this.

Prerequisites:

- Card reader for microSD cards
- Download the Rufus tool from: https://rufus.ie/

Proceed as follows:

1. Start the Rufus tool on a PC with Windows operating system.
2. Click on **Select** and select the image you want to copy to the microSD card.



3. Select the microSD card as the target drive under **Device**.
4. Click **Start** to copy the image to the microSD card.

⇨ The process may take a few minutes. Do not cancel the process until the message Ready appears. Then boot the industrial PC from the microSD card and start Beckhoff RT Linux®.

3 First steps

The following chapter describes the basic first steps with Beckhoff RT Linux®. An industrial PC with a pre-installed Beckhoff RT Linux® operating system or alternatively a current image for installing the operating system is required.

First system startup

Connect all network cables to the industrial PC and switch on the power supply. If you have connected a monitor to the device, you will see the boot screen.

During the boot phase of Beckhoff RT Linux® various messages will appear, ending with the login prompt. Log in to the console with user "Administrator" and password "1". The password should be changed immediately after logging in (see: Change default password).

```
Debian GNU/Linux BTN-000twtq7 tty1

BTN-000twtq7 login: Administrator
Password:
Linux BTN- 000twtq7 6.12.9-bhf1 #178445 SMP PREEMPT_RT Fri Jan 10 10:43:52 UTC 2025 aarch64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Feb 5 09:37:45 2025
```

3.1 Determining the IP address

For all subsequent work with the industrial PC and the system, the IP address of the system is required. Using the IP address, remote access via SSH, the Beckhoff Device Manager (web interface) or working with TwinCAT is possible, for example.

Determine IP address with monitor

There are several ways to determine the IP address to access the device via the network. If you have connected a local monitor, you can log in with the user "Administrator" and the password "1".

Enter the command `ip addr show` in the console to output all available Ethernet interfaces in the system:

```
Administrator@BTN-000twtq7 ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: end1: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc mq state DOWN group default qlen 1000
    link/ether 00:01:05:3d:69:12 brd ff:ff:ff:ff:ff:ff
3: end0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:01:05:3d:69:13 brd ff:ff:ff:ff:ff:ff
    inet 192.17.42.14/22 metric 1024 brd 192.17.43.255 scope global dynamic end0
       valid_lft 689597sec preferred_lft 689597sec
    inet6 fe80::201:5ff:fe3d:6913/64 scope link
       valid_lft forever preferred_lft forever
Administrator@BTN-000twtq7:~$
```

By default, the Ethernet interfaces are configured to receive an IP address from a local DHCP server. In this sample, this is the IP address: `192.17.42.14` which was assigned for the interface `end0` via which the industrial PC was connected to the network.

Determine IP address without monitor

You can access the system via SSH from a Windows system. Some basic network diagnostic tools and commands in PowerShell are used for this purpose.

First identify the network interface of your Windows system with the command `ipconfig`. Search for the desired network connection in the output and make a note of the interface number, which is labeled %??. This number is required to correctly ping the IPv6 link-local address. The output in PowerShell could look like this:

```
Ethernet adapter Ethernet 5:
   Connection-specific DNS Suffix  . : example.com
   Link-local IPv6 Address . . . . . : fe80::5197:ef72:a352:b7f7%17
   IPv4 Address. . . . . . . . . . . : 172.17.42.17
   Subnet Mask . . . . . . . . . . . : 255.255.252.0
   Default Gateway . . . . . . . . . : 172.17.40.1
```

Use the command `ping ff02::1%??` to check which IPv6-enabled devices are active and accessible in the local network. %?? stands for the interface number (interface index), which specifies the network interface via which the ping is to be sent. Assuming that the interface number of your network interface is 17, the command would look like this:

```
ping ff02::1%17
```

If the ping is successful, you will receive a list of the responses from the various devices in the network:

```
Pinging ff02::1%10 with 32 bytes of data:
Reply from fe80::201:5ff:fe50:5911: icmp_seq=1 ttl=64 time<1 ms
Reply from fe80::201:5ff:fe3d:6913: icmp_seq=1 ttl=64 time<1 ms
...
```

Depending on your network and firewall settings, the request may time out. However, this is not a problem and you can continue with the command `Get-NetNeighbor`.

Use the following command in PowerShell to determine the MAC address of the target system:

```
Get-NetNeighbor -LinkLayerAddress 00-01-05* -AddressFamily IPv6
```

The output lists the MAC addresses and the corresponding IPv6 addresses of all devices in the network whose MAC address begins with 00-01-05. Identify the Beckhoff RT Linux® system using the MAC address on the name plate of the industrial PC and make a note of the IPv6 address.

Use the following ping command to ensure that the target system can be reached. Replace `%??` with the corresponding interface number:

```
ping fe80::201:5ff:fe3d:6913%17
```

Finally, you can establish an SSH connection to the target system. To do this, use the following command:

```
ssh Administrator@fe80::201:5ff:fe3d:6913%17
```

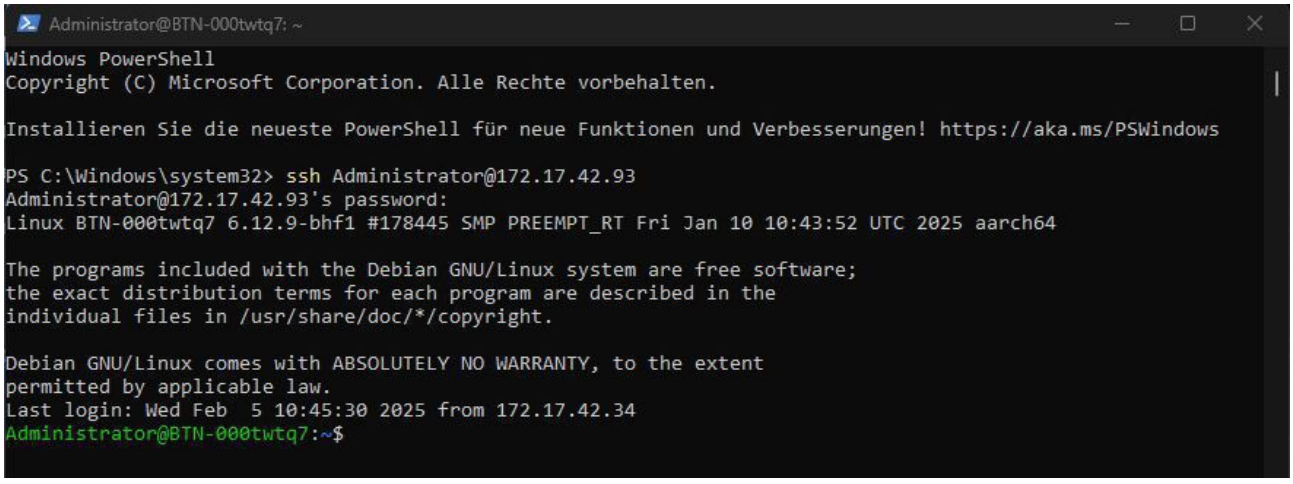


Fig. 1: Remote access via SSH using Windows PowerShell.

Determine IP address without PowerShell

If PowerShell cannot be used, the IP address can be derived from the MAC address using the EUI-64 method instead. A typical MAC address consists of 48 bits and is represented in the form `XX:XX:XX:XX:XX:XX`. Example: `00:1A:2B:3C:4D:5E`.

1. Split the 48-bit MAC address into two 24-bit halves.

- First 24-bit half: `00:1A:2B`
- Second 24-bit half: `3C:4D:5E`

2. Insert the 16-bit FFFE mark in the center: the 16-bit sequence `FFFE` is inserted between the two halves. This results in a 64-bit address:
   - `00:1A:2B:FF:FE:3C:4D:5E`
3. Invert the 7th bit of the first byte (universal/local bit). In the example address `00:1A:2B:FF:FE:3C:4D:5E`, the 7th bit changes from 00 (binary 00000000) to 02 (binary 00000010).
   - New address: `02:1A:2B:FF:FE:3C:4D:5E`
4. Convert to IPv6 format and add prefix: The modified 64-bit address is combined with the link-local prefix `fe80::` to obtain the full IPv6 address:
   - Result of the conversion: `fe80::021a:2bff:fe3c:4d5e`
5. A ping command `ping <IPv6-Adresse>` can be used to check the accessibility and connectivity of a device in the local network.

3.2 Configuring the Package Server

Under Beckhoff RT Linux®, the package management tool `apt` (Advanced Package Tool) is used to download and install software packages provided by Beckhoff. These packages are provided by Package Servers, which require authentication (login) before they can be accessed.

Login data must be stored in a configuration file so that Beckhoff RT Linux® can access the Beckhoff Package Server and download packages. Without this information, you will not be able to update or install packages from this server.

The login data of your myBeckhoff account is used as login data. Register at myBeckhoff if you have not yet activated a user account.

3.2.1 Creation of an authentication file for APT:

The `/etc/apt/auth.conf.d/` directory is intended for the authentication files of `apt`. Files in this directory are automatically read by `apt` and used for authentication. Create a file named `bhf.conf` in that directory using the command:

```
sudo nano /etc/apt/auth.conf.d/bhf.conf
```

Store the access data for the Beckhoff Package Server in this file. Contents of the file `bhf.conf`:

```
machine deb.beckhoff.com
login example@mail.com
password xyz123

machine deb-mirror.beckhoff.com
login example@mail.com
password xyz123
```

- **Machine:** The name of the Package Server that `apt` should connect to (in this case deb.beckhoff.com and deb-mirror.beckhoff.com) is specified here.
- **Login and password:** The login details of your myBeckhoff account are used for authentication.

To save the file, press **[Ctrl] + [O]**. To exit the editor, press **[Ctrl] + [X].**

3.2.2 Reference to the unstable area of the Beckhoff repository

In the next step, the Sources list must be modified so that packages are obtained from an unstable area of the Beckhoff repository during the development phase. To do this, the package source must be changed so that it refers to this unstable area.

Open the file `bhf.list` with the command `sudo nano /etc/apt/sources.list.d/bhf.list` and change the existing entry by adding `-unstable`:

```
https://deb.beckhoff.com/debian bookworm-unstable main
```

When `apt` is executed with the command `apt update` or `apt install`, it attempts to connect to the Package Server to obtain packages. In this case, `apt` checks the authentication files under `/etc/apt/auth.conf.d/` to see whether login data is available for a server.

If the data is available and correct, `apt` automatically establishes the connection and can download packages from the Beckhoff Package Server.

3.3 Installing TwinCAT

After the authentication file for `apt` has been created and the reference to the Unstable area of the Beckhoff repository has been created (see: Configuring the Package Server [▶ 13]), the system and the installed packages can be updated:

```
sudo apt update
```

The installation of TwinCAT on industrial PCs with Arm® processors can then be started with the following command:

```
sudo apt install tc31-xar-um
```

For all other industrial PCs, start the installation with the following command instead:

```
sudo apt install tc31-xar-um libtcrte
```

4 Network configuration

Under Beckhoff RT Linux® the network configuration is managed with the system service `systemd-networkd`. The service is integrated into `systemd` by default and is used to manage network configurations and configure network devices such as Ethernet interfaces, WLAN, VLANs or virtual network devices.

4.1 Setting up a static IP address

To configure a static IP address with `systemd-networkd`, create a configuration file in the */etc/systemd/network/* directory. In this configuration file, the corresponding network settings for the network interface are defined and parameters such as IP address, gateway and DNS server are specified. After restarting the system, the `systemd-networkd` service automatically adopts the settings. Create a separate configuration file with the individual network settings for each network interface.

By default, the Ethernet interfaces are configured to obtain an IP address from a local DHCP server. This standard configuration is pre-installed as `/usr/lib/systemd/network/20-wired.network` and should not be edited.

Proceed as follows:

1. Use the command `ip addr show` to determine the available Ethernet interfaces.
   ⇨ Examples of available Ethernet interfaces: `lo, end1, end0`
2. Create a configuration file in the directory `/etc/systemd/network/`, for example with the name `10-end0-static.network`.
   ⇨ The number 10 at the beginning of the file name `10-end0-static.network` determines the order in which `systemd-networkd` processes this file compared to other files. This allows the priority to be controlled.

```
sudo nano /etc/systemd/network/10-end0-static.network
```

3. Insert the following content and adjust the values to your network requirements:

```
[Match]
Name=end0

[Network]
Address=192.168.1.100/24
Gateway=192.168.1.1
```

4. Save and close the configuration file.
5. Reload the configuration files with the following command without restarting the network service.

```
networkctl reload
```

6. Check whether the configuration has been loaded correctly. Display the network status with `networkctl status`. Use `ip addr show` and `ip route show` to check the IP address and routing.

⇨ These settings ensure that the static IP configuration in the `10-end0-static.network` file is given priority over the DHCP settings in the `20-wired.network` file.

4.2 Firewall

Under Beckhoff RT Linux®, `nftables` is used as a firewall, a framework of the Netfilter project that enables packet filtering, network address translation (NAT) and other applications. The `nftables` firewall is the default and recommended firewall framework in Debian and replaces the old `iptables` and related toolboxes. The firewall is restrictive by default with regard to incoming and forwarded connections. Necessary connections such as local loopback communication, SSH (port 22), ICMP and ICMPv6 are permitted. All outgoing connections are permitted.

The current status and the rules of `nftables` are displayed with the following command:

```
sudo nft list ruleset
```

This command provides a complete overview of the currently configured firewall rules. The configuration files for `nftables` are located in the */etc/nftables.conf.d* directory and contain an initial basic configuration for the system.

4.2.1 Disabling and enabling the firewall

The firewall is enabled by default. Disabling the firewall can be useful or even necessary in many cases, e.g. in a test environment. This step shows how to disable the firewall. Note that without a firewall, incoming and outgoing connections will no longer be checked. Never disable the firewall permanently.

Proceed as follows:

1. Stop the corresponding service.
   ```
   sudo systemctl stop nftables
   ```
2. You must disable the service to prevent the firewall from starting automatically when the system is started.
   ```
   sudo systemctl disable nftables
   ```
3. You can check the status of the service to make sure that nftables is stopped and disabled.
   ```
   sudo systemctl status nftables
   ```
4. The status indicates that the service has stopped.

⇨ These steps disable the firewall, and firewall rules are no longer applied. If the firewall is to be re-enabled, the service must be restarted and re-enabled:

```
sudo systemctl start nftables
sudo systemctl enable nftables
```

4.2.2 Enable port

Automatic port enabling for TwinCAT Functions

Ports that are required for TwinCAT Functions are automatically enabled once the TwinCAT Functions have been installed.

To enable a port in the firewall, a rule must be added that allows incoming connections on this port. As an example, an incoming connection for TCP port 502 is created in a separate configuration file, which is required for Modbus/TCP communication.

Proceed as follows:

1. Use the command `ip addr show` to determine the available Ethernet interfaces.
   ⇨ Examples of available Ethernet interfaces: lo, end1, end0
2. Select an Ethernet interface to be configured for Modbus/TCP communication.
3. Create a configuration file in the */etc/nftables.conf.d/* directory with the name `60-modbus.conf`, for example
   ```
   sudo nano /etc/nftables.conf.d/60-modbus.conf
   ```
4. Insert the following content and adjust the values to your network requirements:
   ```
   table inet filter {
     chain input {
       # accept ModbusTCP
       iifname "end1" tcp dport 502 accept
     }
   }
   ```
5. Save and close the configuration file.
6. Load the new rule with the command
   ```
   sudo systemctl reload nftables
   ```
7. Check the settings and make sure that the configuration has been applied correctly.
   ```
   sudo nft list ruleset
   ```

Trademark statements

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® and XPlanar® are registered trademarks of and licensed by 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.

Debian is a registered trademark owned by Software in the Public Interest, Inc.

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.

More Information:
www.beckhoff.com/linux

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