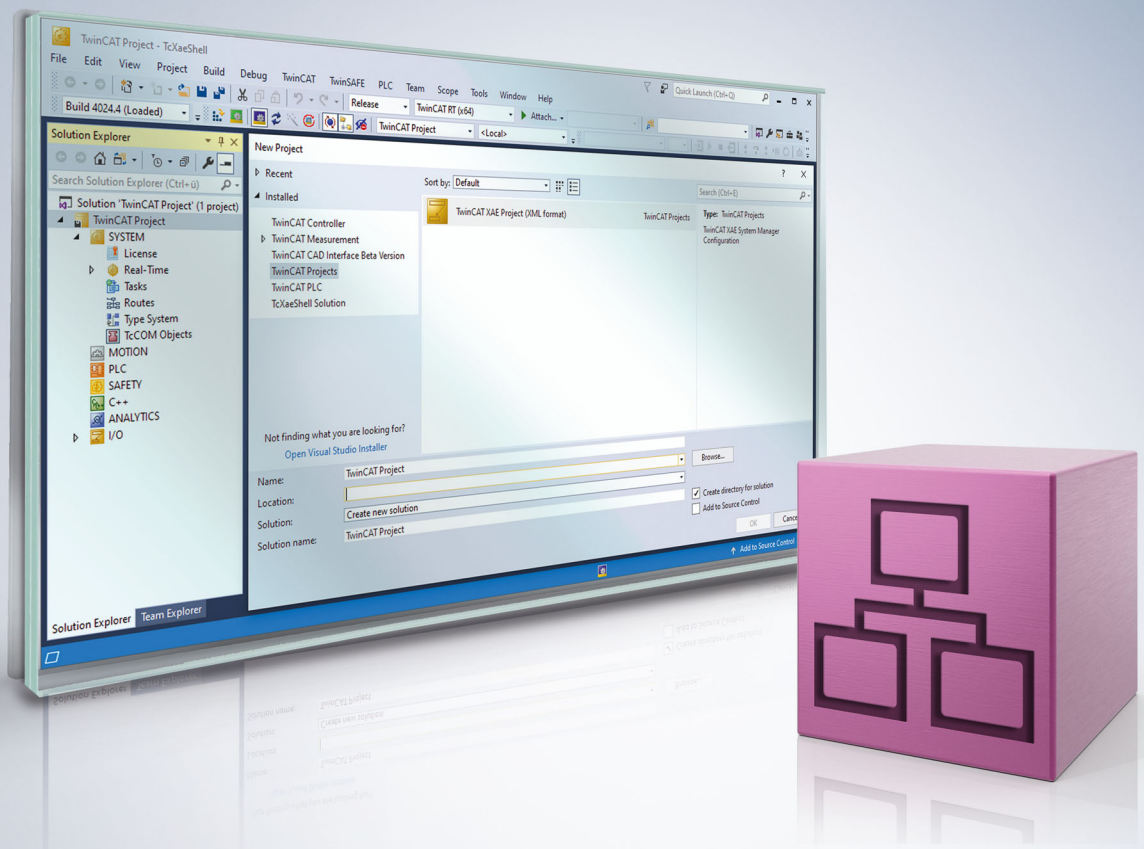
BECKHOFF New Automation Technology

Manual | EN

TF6650

TwinCAT 3 | DBC File Import for CAN



2022-06-28 | Version: 1.1

Table of contents

1 Foreword

1.1 Notes on the documentation

This description is only intended for the use of trained specialists in control and automation engineering who are familiar with applicable national standards.
It is essential that the documentation and the following notes and explanations are followed when installing and commissioning the components.
It is the duty of the technical personnel to use the documentation published at the respective time of each installation and commissioning.

The responsible staff must ensure that the application or use of the products described satisfy all the requirements for safety, including all the relevant laws, regulations, guidelines and standards.

Disclaimer

The documentation has been prepared 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 prior announcement.
No claims for the modification of products that have already been supplied may 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 trademarks of and licensed by Beckhoff Automation GmbH.
Other designations used in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owners.

Patent Pending

The EtherCAT Technology is covered, including but not limited to the following patent applications and patents:
EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702
with corresponding applications or registrations in various other countries.

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

Copyright

1.2 Safety instructions

Safety regulations

Please note the following safety instructions and explanations!
Product-specific safety instructions can be found on following pages or in the areas mounting, wiring, commissioning etc.

Exclusion of liability

All the components are supplied in particular hardware and software configurations 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 engineering who are familiar with the applicable national standards.

Description of symbols

In this documentation the following symbols are used with an accompanying safety instruction or note. The safety instructions must be read carefully and followed without fail!

⚠ DANGER

Serious risk of injury!

Failure to follow the safety instructions associated with this symbol directly endangers the life and health of persons.

⚠ WARNING

Risk of injury!

Failure to follow the safety instructions associated with this symbol endangers the life and health of persons.

⚠ CAUTION

Personal injuries!

Failure to follow the safety instructions associated with this symbol can lead to injuries to persons.

NOTE

Damage to the environment or devices

Failure to follow the instructions associated with this symbol can lead to damage to the environment or equipment.

Tip or pointer

This symbol indicates information that contributes to better understanding.

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

The TF6650 function enables DBC file formats (.dbc) to be read. The DBC data format is a CAN network description and enables the definition of attributes and the assignment of these attributes to the elements of the network. DBC files are text files containing, for example, scaling information for CAN data and signal definitions. The import and pre-processing of the data takes place via the TF6650 using the parameters stored in the DBC file. As an additional function, network nodes can be simulated according to the DBC file. For a hardware interface, the function uses the EL6751 CANopen master terminal or the M510 CAN optional interface with the Embedded PCs or the FC512x plug-in card.

3 Requirements

Software

The TF6650 is contained in TwinCAT Version 3.1 Build 4024.12 or higher. No further installation is required.

Hardware

The function can only be used in conjunction with an EL6751-0000 with software version 16 (V1.14) or higher.
When using the M510 CAN optional interface with the Embedded PCs or the FC512x, the version of the CCAT driver should be V1.29 or higher.

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).

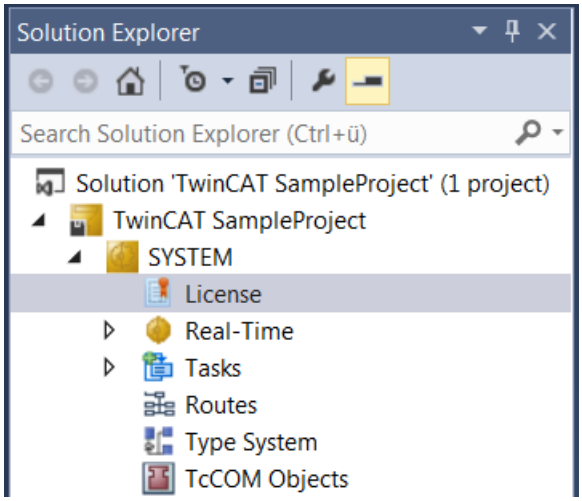
Licensing the full version of a TwinCAT 3 Function

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

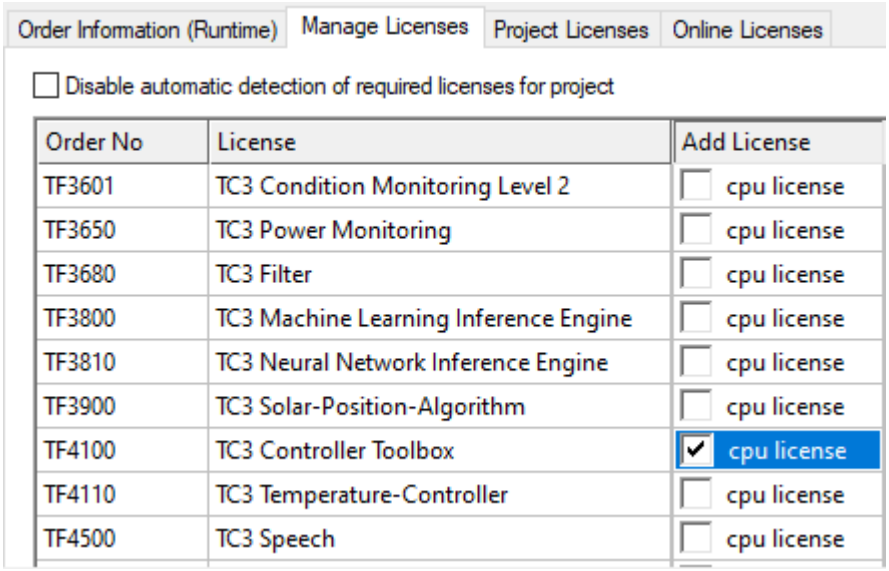
A 7-day test version cannot be enabled for a TwinCAT 3 license dongle.

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.

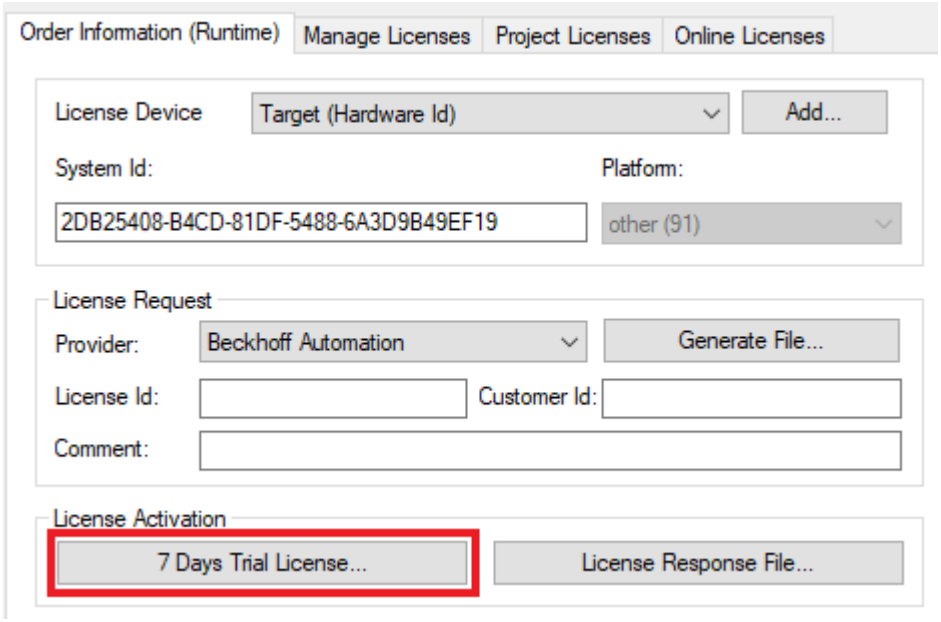


⇨ The TwinCAT 3 license manager opens.

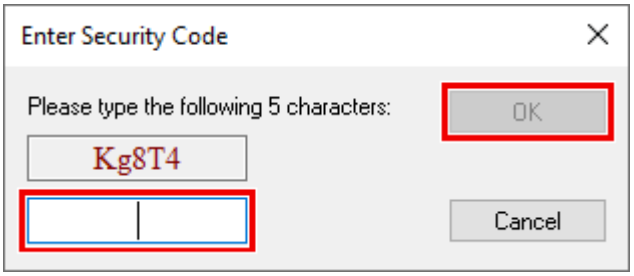
5. Open the **Manage Licenses** tab. In the **Add License** column, check the check box for the license you want to add to your project (e.g. "TF4100 TC3 Controller Toolbox").



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.

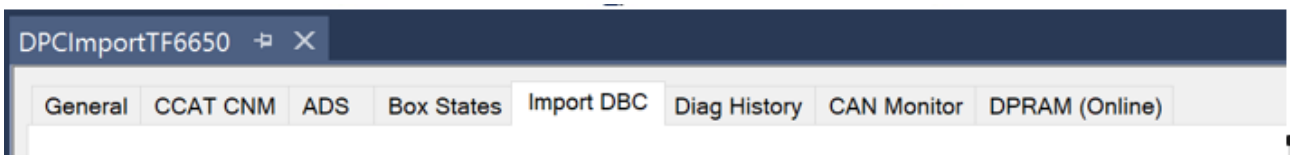


8. Enter the code exactly as it is displayed and confirm the entry.
9. Confirm the subsequent dialog, which indicates the successful activation.
   ⇨ In the tabular overview of licenses, the license status now indicates the expiry date of the license.

10. Restart the TwinCAT system.
⇨ The 7-day trial version is enabled.

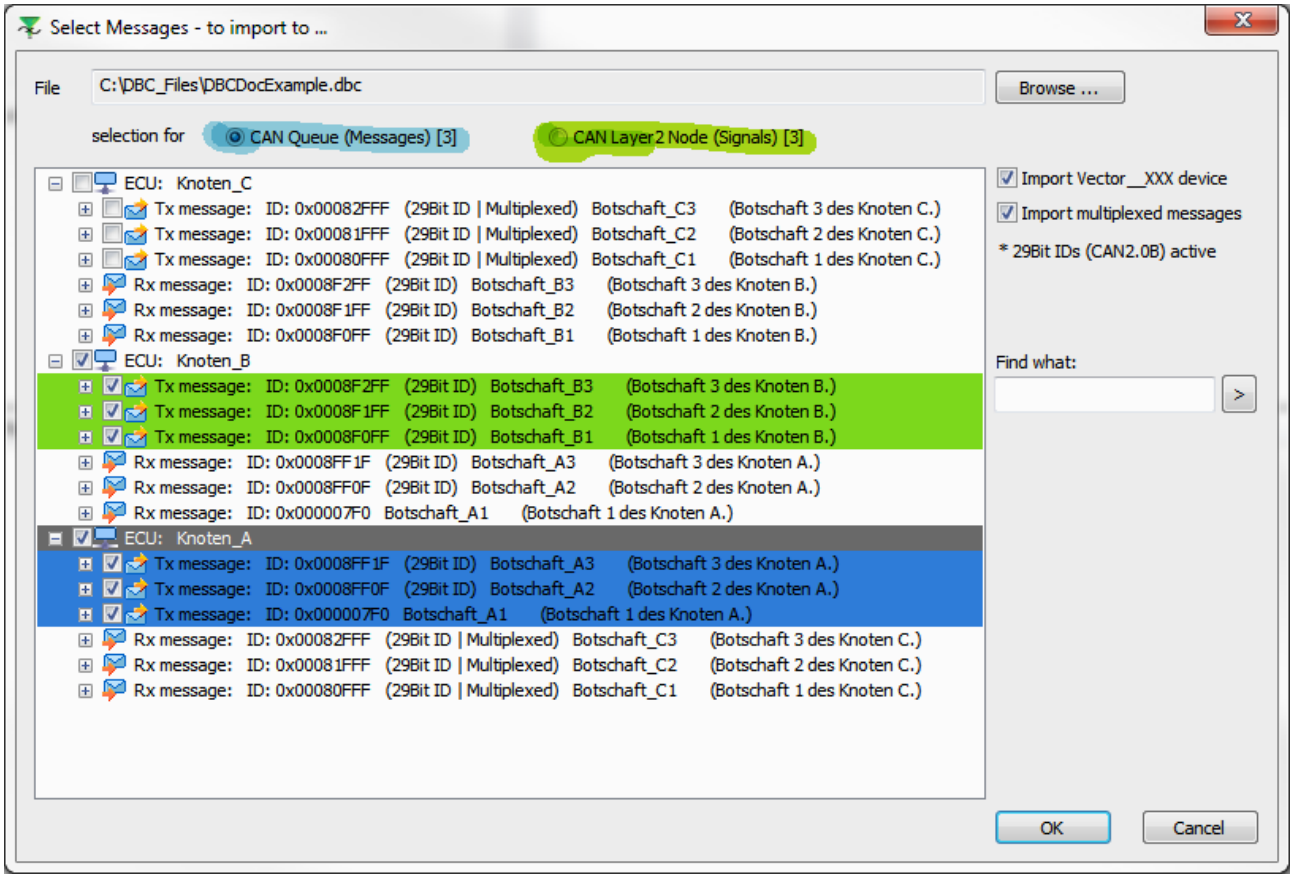
5 CAN-Layer2 nodes

The CAN device of the EL6731 or of the CCAT CAN master has an additional tab with the name **Import DBC**:



CAN nodes (without CANopen Application Layer) can be created by importing DBC file.

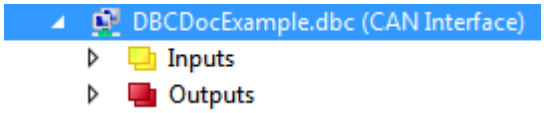
Parallel selection for CAN Queue and CAN Layer2 node:



The creation of a CAN interface is also possible via this route, since the import dialog enables the parallel selection of messages for the CAN interface and for CAN Layer2 nodes.

The assignment of the selection is illustrated by the colored background (green – CAN Layer2 node / blue – CAN Queue).

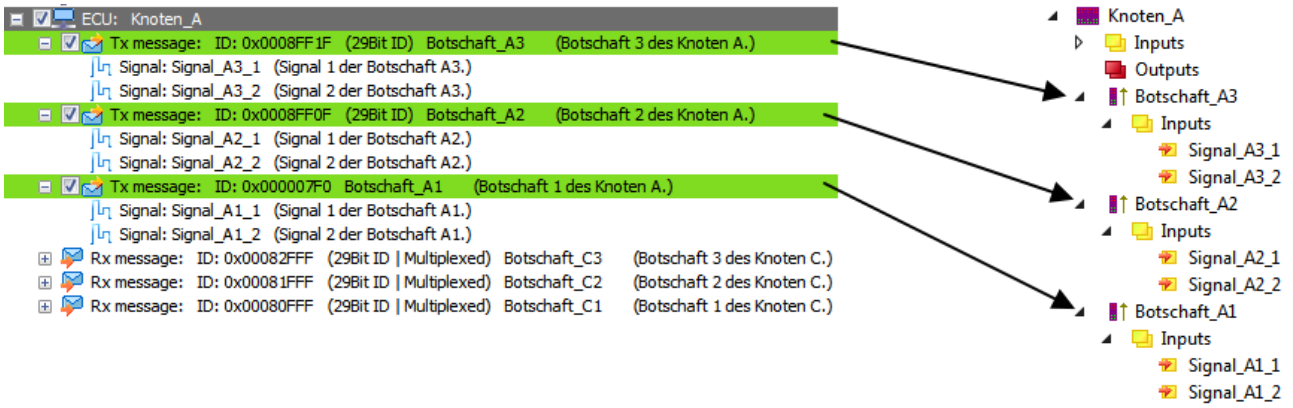
CAN interface with the name of the DBC file:



If no CAN interface has been created before the import, this takes place with the name of the loaded DBC file.

5.1 Structure of the imported nodes

Nodes marked for import in the DBC file are inserted in the TwinCAT System Manager as CAN nodes. Messages are mapped as PDOs and signals as variables.



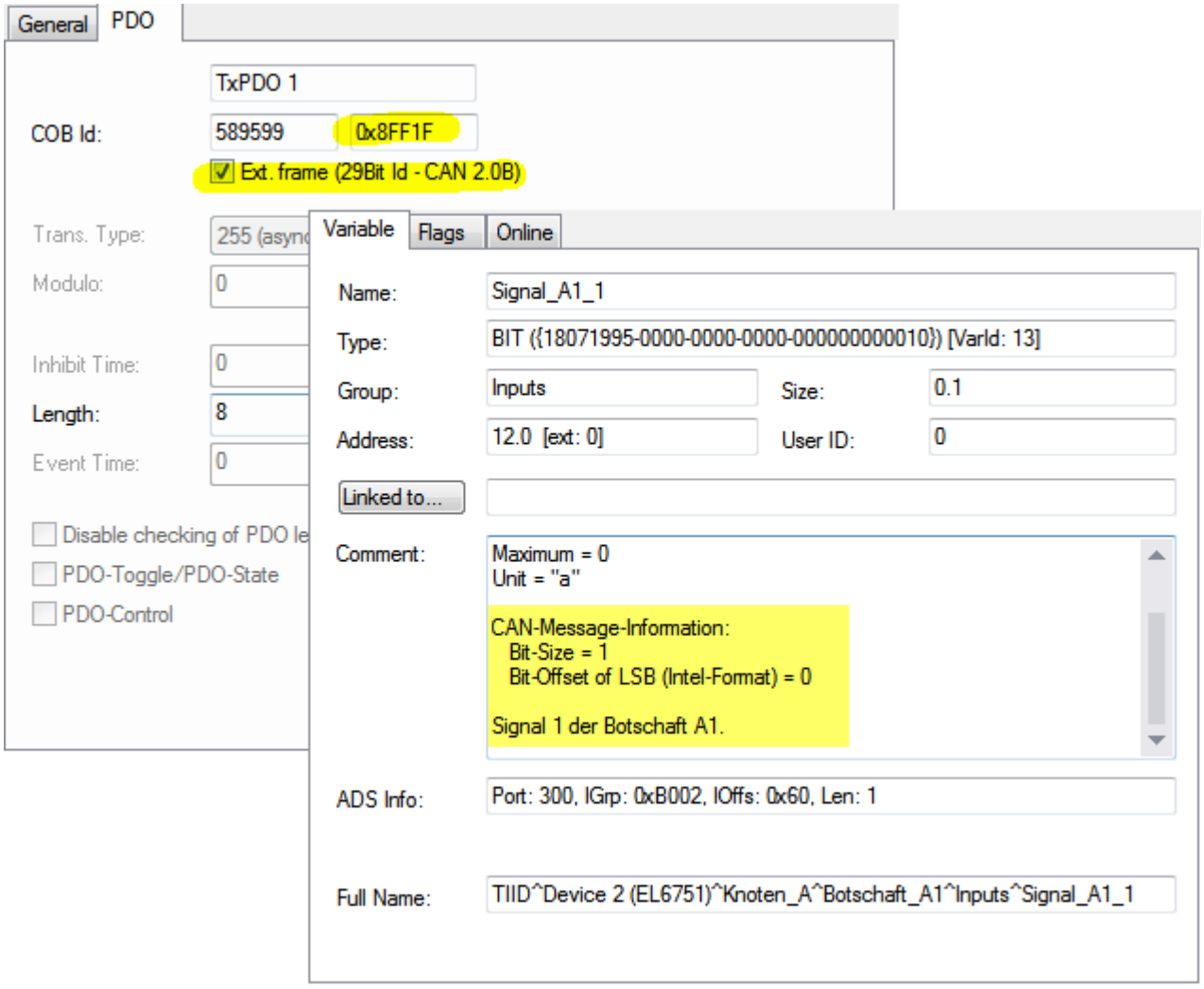
If messages with extended ID (29-bit) are imported, then the PDO contains a corresponding note.

A comment with structure information from the DBC file is appended to the imported variables.

The data types of the CAN node variables correspond to the specification by the DBC file:

- signed / unsigned integer
- Bit size

The Motorola/Intel bit-counting method or the swapping of the bytes in the case of integers is observed. Messages are mapped as PDOs and signals as variables:



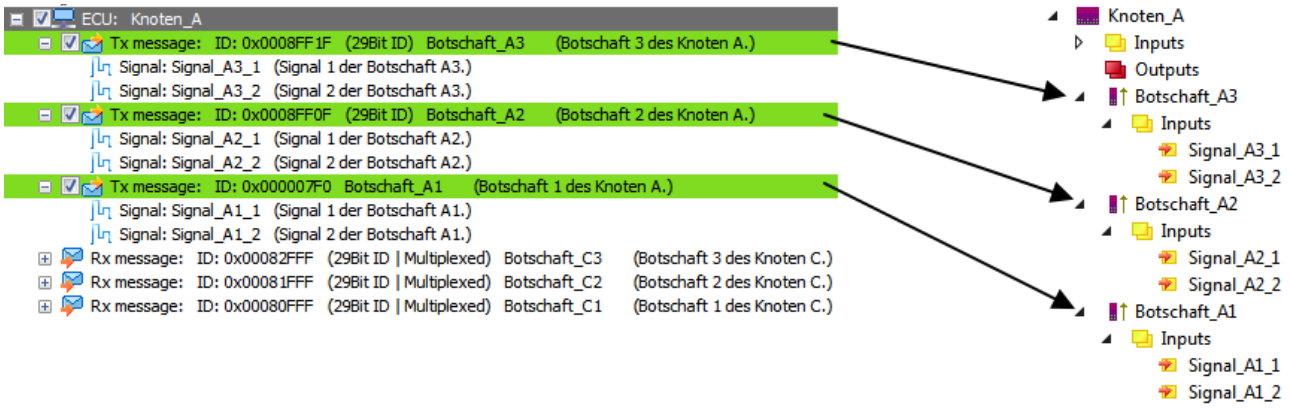
5.2 Prepared DBC Signals

The most significant extension in connection with the DBC file import is the additionally inserted TCOM module "Prepared DBC Signals"  .

This module performs the scaling given by the DBC format and the (de)multiplexing of the signals.

5.2.1 Structure (hierarchy)

Parallel structure of the CAN nodes underneath the **Prepared DBC Signals** node:



In parallel with the CAN nodes, the following are inserted in accordance with the structure of the DBC file underneath **Prepared DBC Signals**: nodes, messages and signals.

5.2.2 Scaling

Parameters for scaling are added to each signal defined in the DBC file. These are

- Factor
- Offset
- Minimum / Maximum.

In this context the DBC format refers to **raw** and **physical** values. The raw values are always transmitted on the CAN bus. The Prepared DBC Signals module shows the **physical values**, while the CAN Layer2 nodes show the **raw values**

Physical values and raw values are converted to each other in the following context:

Equations for the scaling

$$physical\ value = raw\ value * factor + offset$$

$$raw\ value = \frac{(physical\ value - offset)}{factor}$$

The **physical values** are limited to the specified range (Minimum/Maximum) after the calculation.

5.2.3 Data type conversion

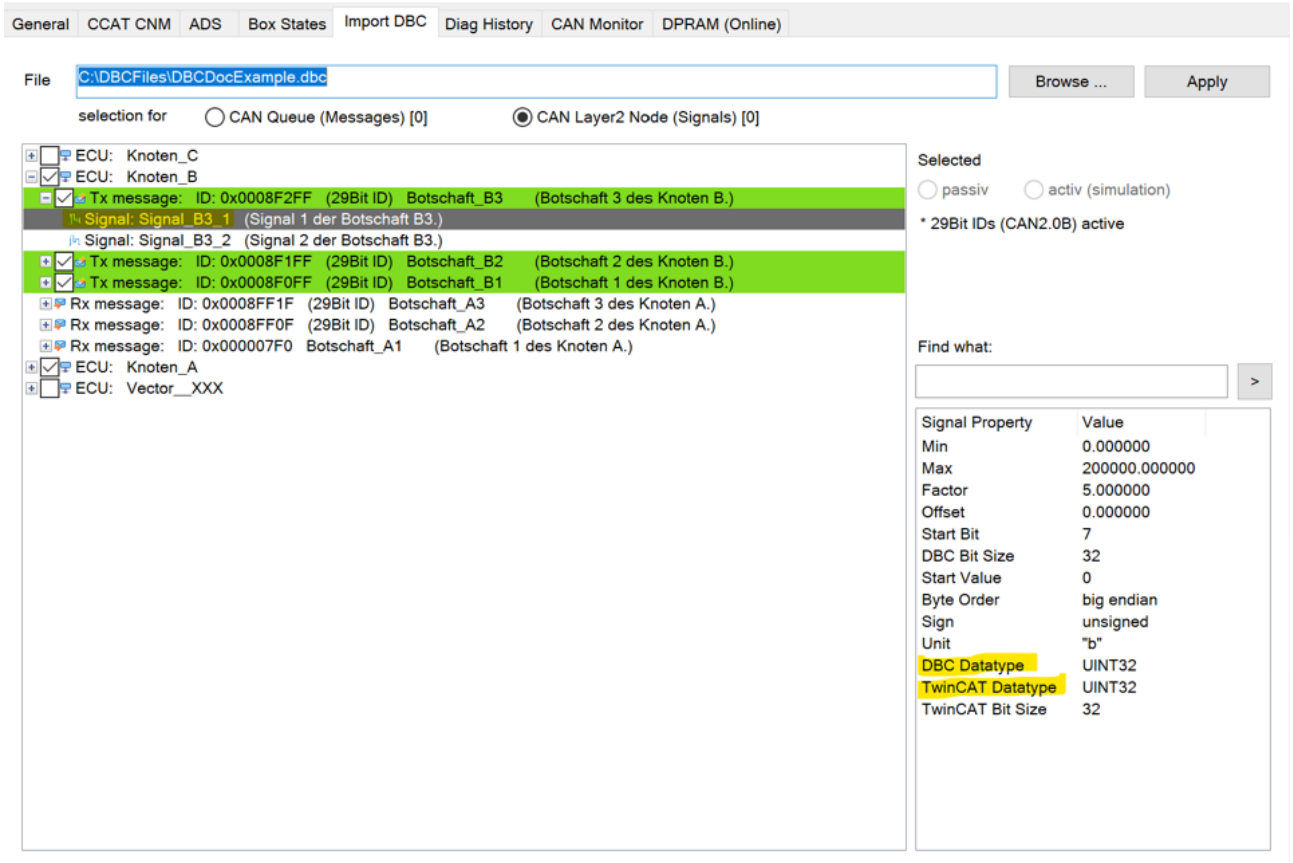
Due to the scaling performed by the Prepared BDC Signals module, it is necessary to adapt the data types of the signals so that accuracy is not lost when scaling. In principle, all signals in the Prepared BDC Signals module are of the data type DOUBLE (64-bit (*double precision*)).

In the case of signals having an integer factor and an integer limit value (Minimum/Maximum), the conversion to DOUBLE is dispensed with and an integer data type is used instead.

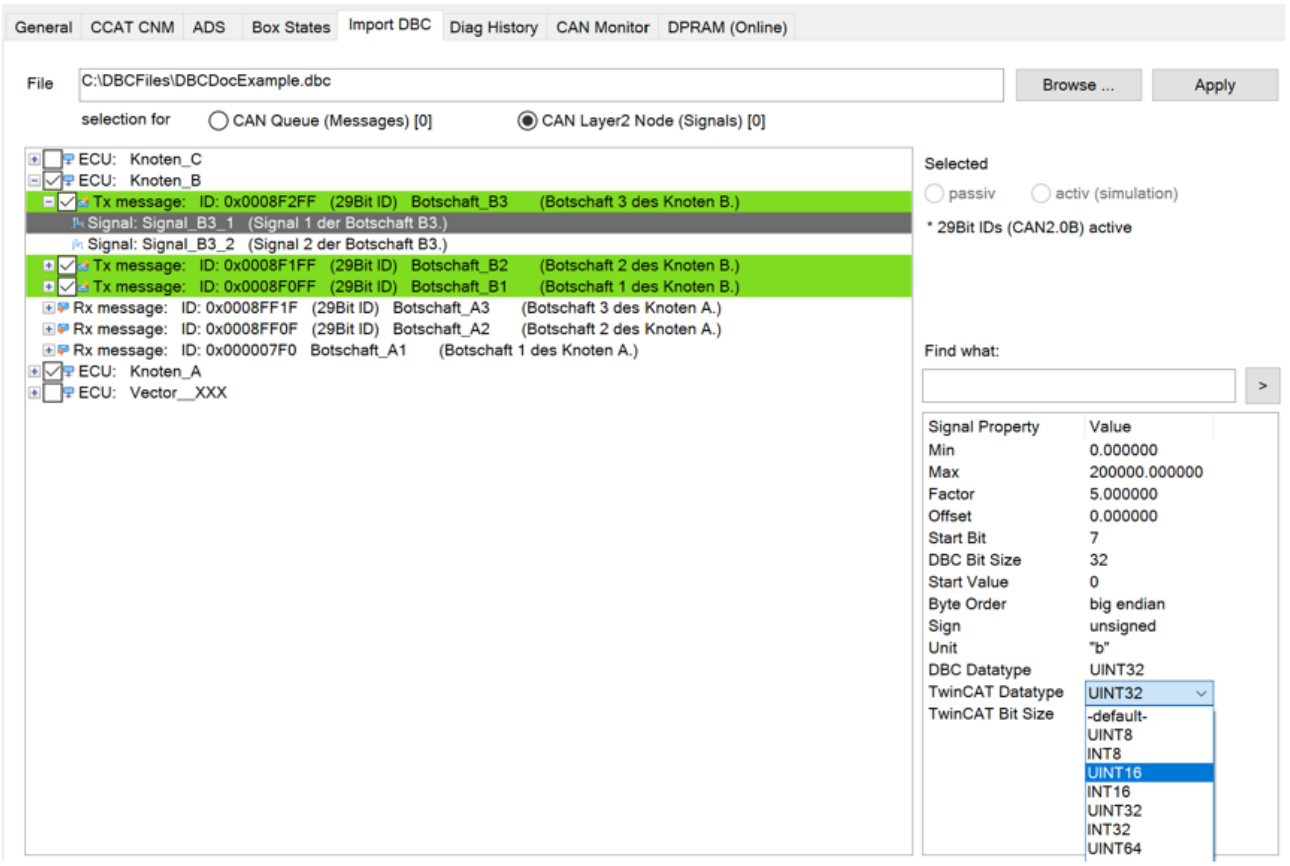
The data type conversion performed by the import can be displayed for each signal on the import page and changed if desired.

For the "Prepared DBC Signals" always exactly the data type is created that you select at **TwinCAT Datatype**.

First confirm your selection with the button **Write new Datatype,** and then with **Apply,** so that the change is applied.

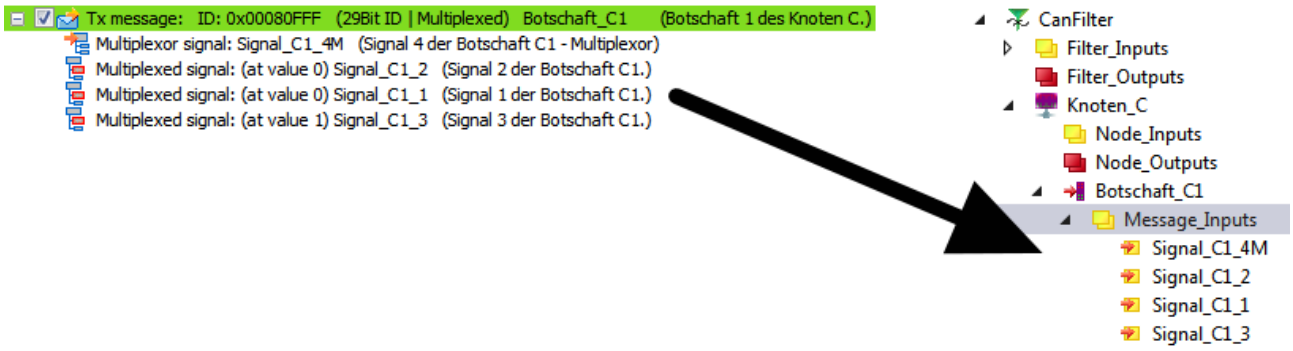


The signal properties **TwinCAT Datatype** and **TwinCAT Bit Size** can be adapted prior to the import. If a different data type is set, a new BitSize is ignored.



5.2.4 Multiplexed data

Multiplexed signals defined in the DBC file (see fig. Multiplexed Signals [▶ 29]) are accepted into the Prepared DBC Signals node as normal signals.



The demultiplexing procedure is carried out as follows on arrival of the corresponding CAN message:

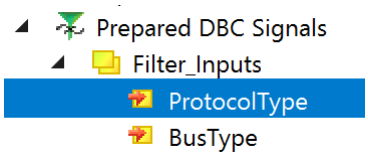
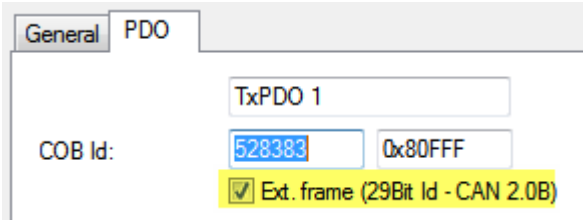
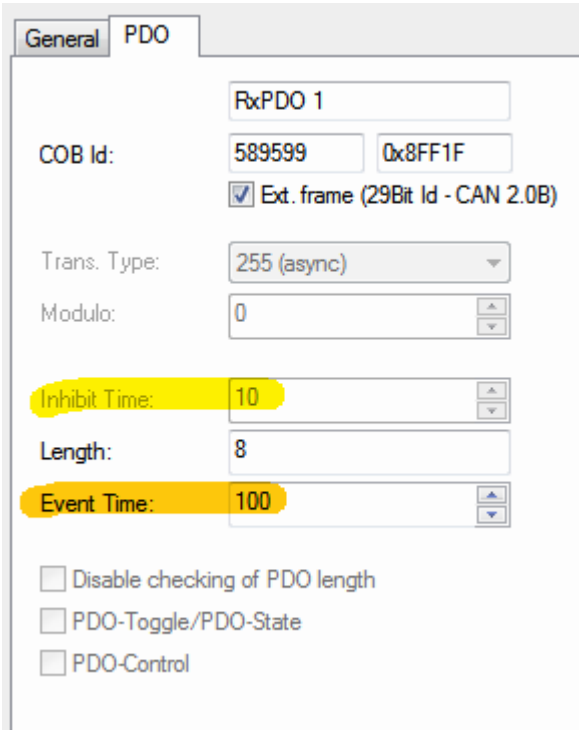
- if the message contains multiplexed signals, the multiplexer of the message is evaluated and only the signals that the frame contains are copied according to the multiplexer.
- Multiplex signals not contained in the frame are not updated

The multiplexing procedure (when transmitting messages), is initiated by changing a multiplexed signal before transmitting the message:

- If the signal that was changed is a multiplexed signal, the signal is copied to the corresponding position in the frame and the multiplexer signal is set to the multiplex value of this signal.
- The CAN frame is sent.

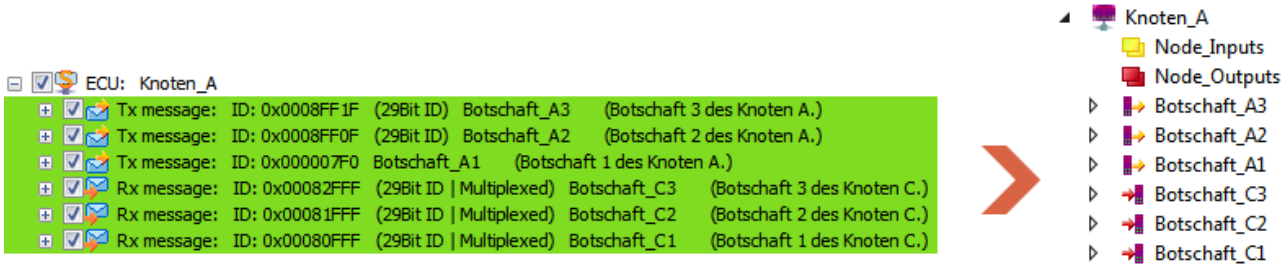
5.2.5 Further signal attributes

The networks, nodes, messages and signals defined in a DBC file can possess additional attributes. Some of these attributes are evaluated by the supplement during the import. These are:

Object type	Attribute	Application
Network	ProtocolType BusType	 The Prepared DBC Signals module additionally contains these variables: • **ProtocolType** = 1, the imported DBC file describes a J1939 network • **ProtocolType** = 0, the protocol is not defined in detail • **BusType** = 1, the bus described is a CAN bus • If **BusType** = 0, the bus type is not described in detail
Message	VFrameFormat	 Used to distinguish between • standard CAN frames (11-bit ID) and • extended frames (29-bit ID) • as well as J1939PG messages.
	GenMsgDelayTime GenMsgCycleTime GenMsgSendType	 If a message is marked as "cyclic", then the **GenMsgCycleTime** is adopted as the **Event Time** in the PDO settings. The frame is thus cyclically transmitted. The value of **GenMsgDelayTime** is adopted as the **Inhibit Time**.
Signal	GenSigStartValue	The value specified under GenSigStartValue is used as the starting value (default value) of the signal.

5.2.6 Simulation of network nodes

The Rx messages can also be imported in the case of nodes that are marked for simulation:



As described under "Marking a node for simulation [▶ 29]", individual nodes of the DBC file can be marked for simulation. Nodes with this marking also import their messages listed under RX messages (if selected). Furthermore, a direction reversal of the inputs and outputs takes place.

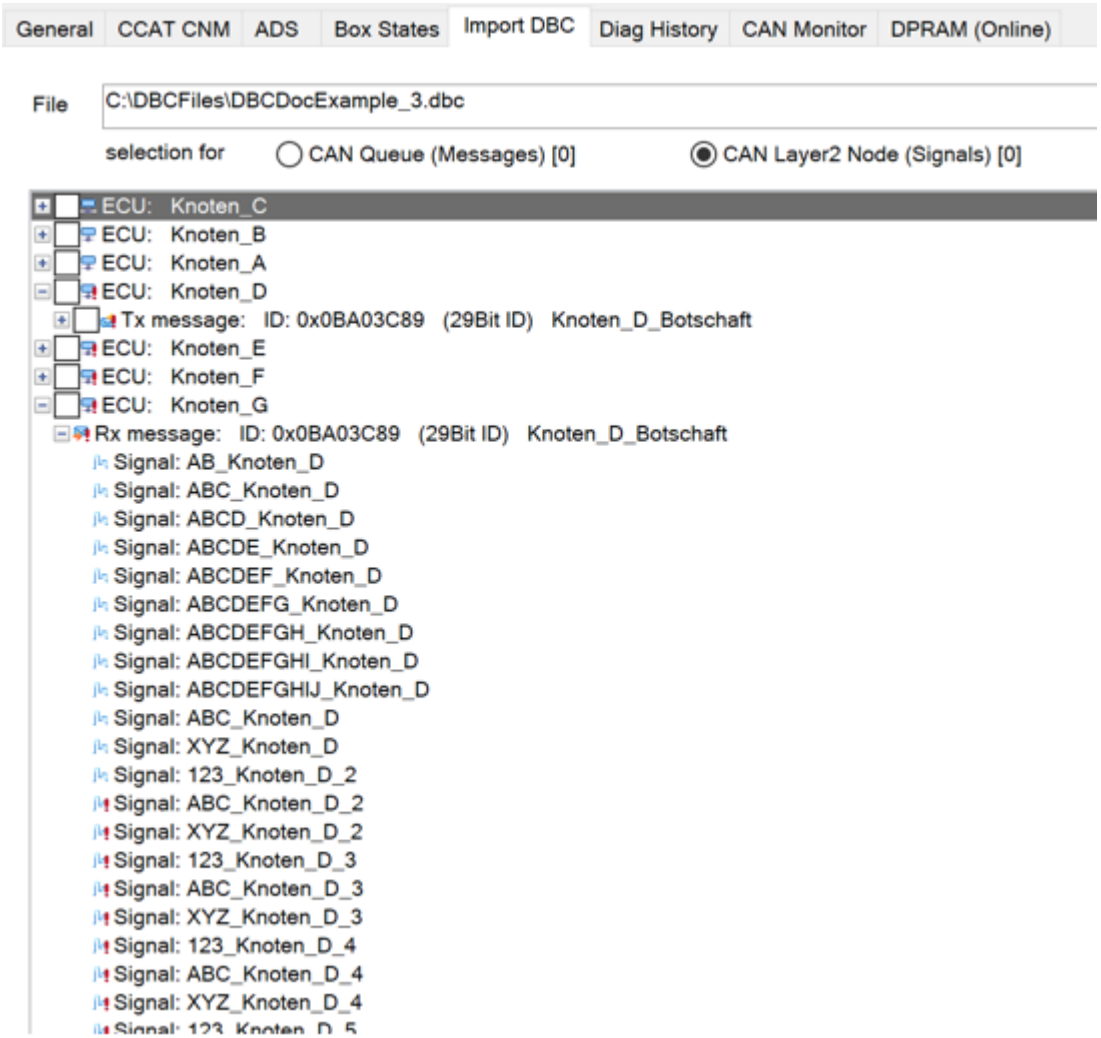
5.2.7 Reimport

If a change is necessary when selecting the message to be imported after the import of a DBC file, the DBC import dialog can be called again (see additional **Import DBC** tab of the CAN device).

The dialog displays the last loaded DBC file. The markings set beforehand are likewise set. All messages are reimported on quitting the import dialog. Existing variable links are re-established.

5.2.8 Tolerating messages with data lengths exceeding 8 bytes

ECU, messages and signals that exceed the 8-byte limit are marked with an exclamation mark:



In addition to messages that are defined by maximally 8 bytes of data, the DBC format also enables the definition of messages with data lengths exceeding 8 bytes. These cannot be transmitted with a CAN Frame. A transport protocol must be used for the transmission of such large messages (as are usual, for example, with the J1939 protocol).

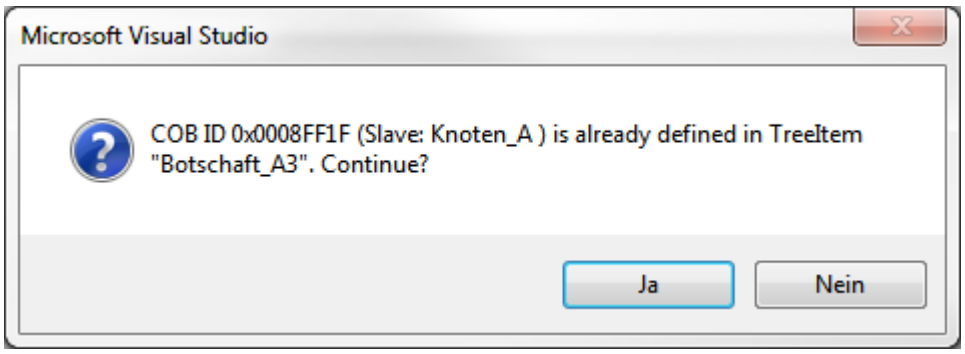
Such a high protocol is not implemented. However, such messages are read and marked with an exclamation mark. For a better overview, the ECU and the signals that exceed the 8-byte limit are also marked in the same way.

5.2.9 General notes

CAN interface and CANopen nodes on a single CAN device are not supported. Conversely, the CAN interface and CAN Layer2 nodes are supported.

If the CAN interface and CAN-Layer2 nodes are inserted under a CAN device, the CAN-Layer2 nodes always have priority when receiving CAN messages (even if the ID is set in the acceptance filters of the CAN queue). All other IDs that are also entered in the acceptance filter are routed to the CAN queue.

Note on double-assigned CAN IDs:



An Extended ID in the range of the 11-bit ID (< 0x7FF) is not an 11-bit ID. This is therefore not received by the CAN Layer2 nodes. The CAN Queue, conversely, does not make this distinction, but instead displays this extended or standard ID in bit 31.

Before the activation of the configuration, an additional check is performed to ascertain whether CAN IDs have been assigned twice. If so, this will be reported.

5.3 Necessary firmware version

The support of the 29-bit ID for the PDOs is possible with the EL6751 from this firmware version:

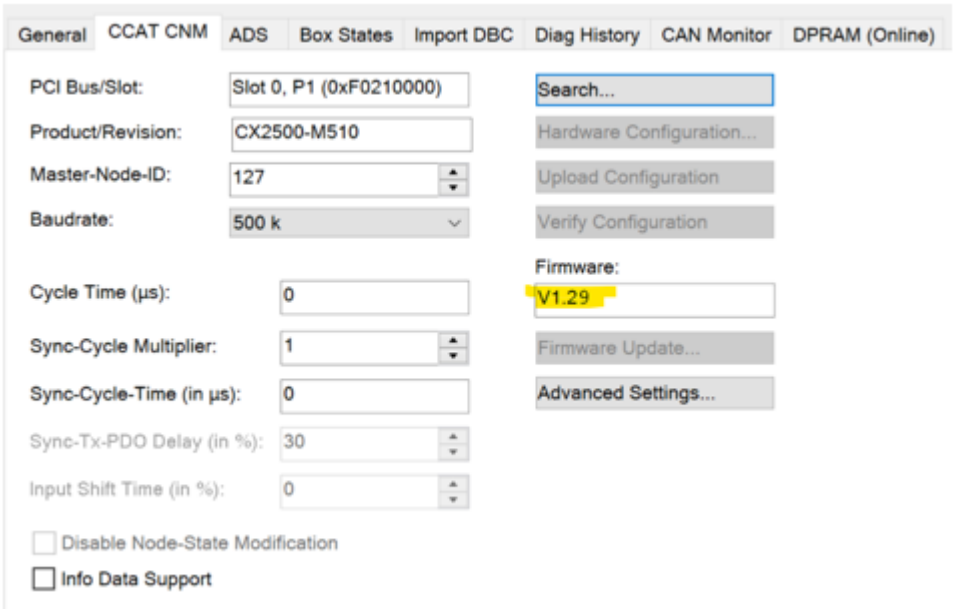
- FW16(V1.14)

If an EL6751 is to be used that uses an older firmware, the following error message (Emergency) will be shown when activating the configuration:

```
(COE-Emergency (Hex:ff00,02,"10 0607 0620") )
```

PDOs with 11-bit identifier already function with older firmware versions.

TwinCAT Version 4024.12 or higher is to be used when using the M510 CAN optional interface with the Embedded PCs or the FC512x. The version of the CCAT driver should be V1.29 or higher:



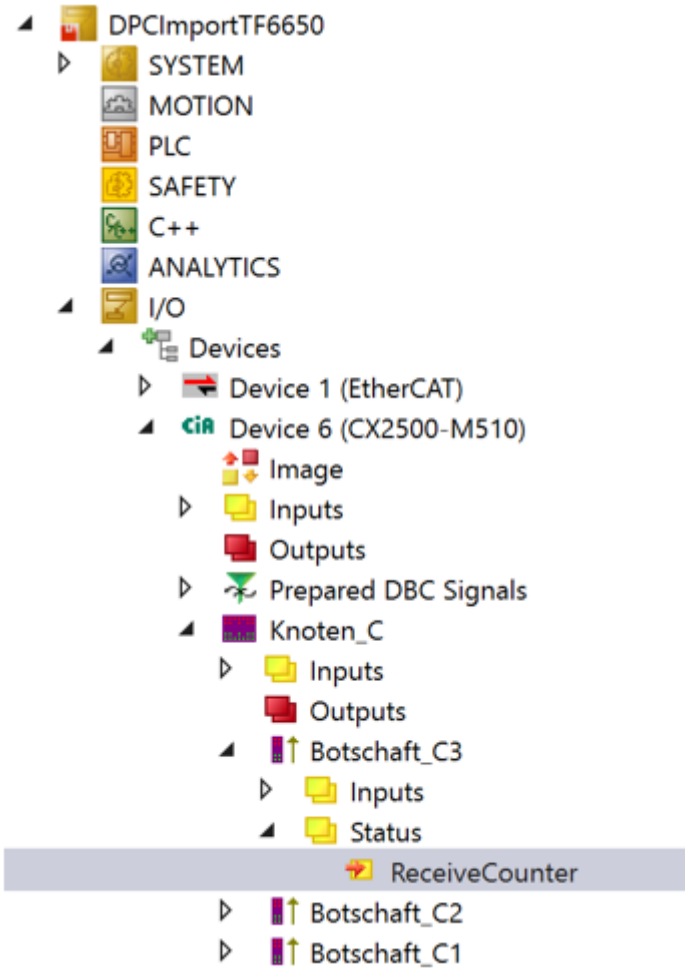
6 Monitoring messages with CCAT-based CAN interfaces

If messages from a network device are sporadically not sent cyclically, it may be necessary to monitor the arrival of a message.

The CCAT-CAN device offers the option here to perform monitoring with the **ReceiveCounter** in the IO tree.

The **ReceiveCounter** is located in the DBC - RAW data underneath the associated message.

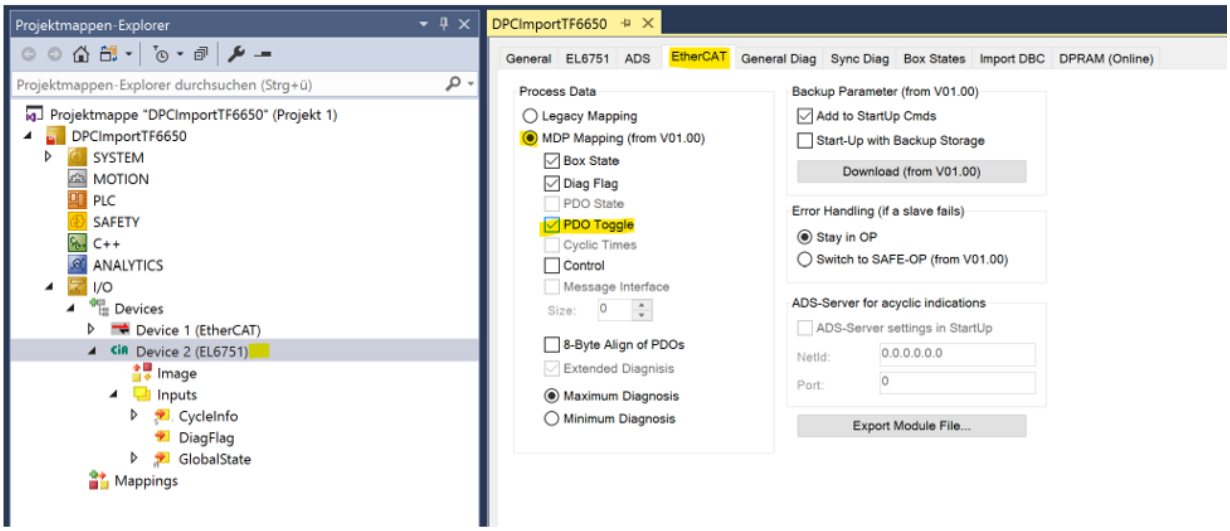
If the message arrives, the counter is incremented.



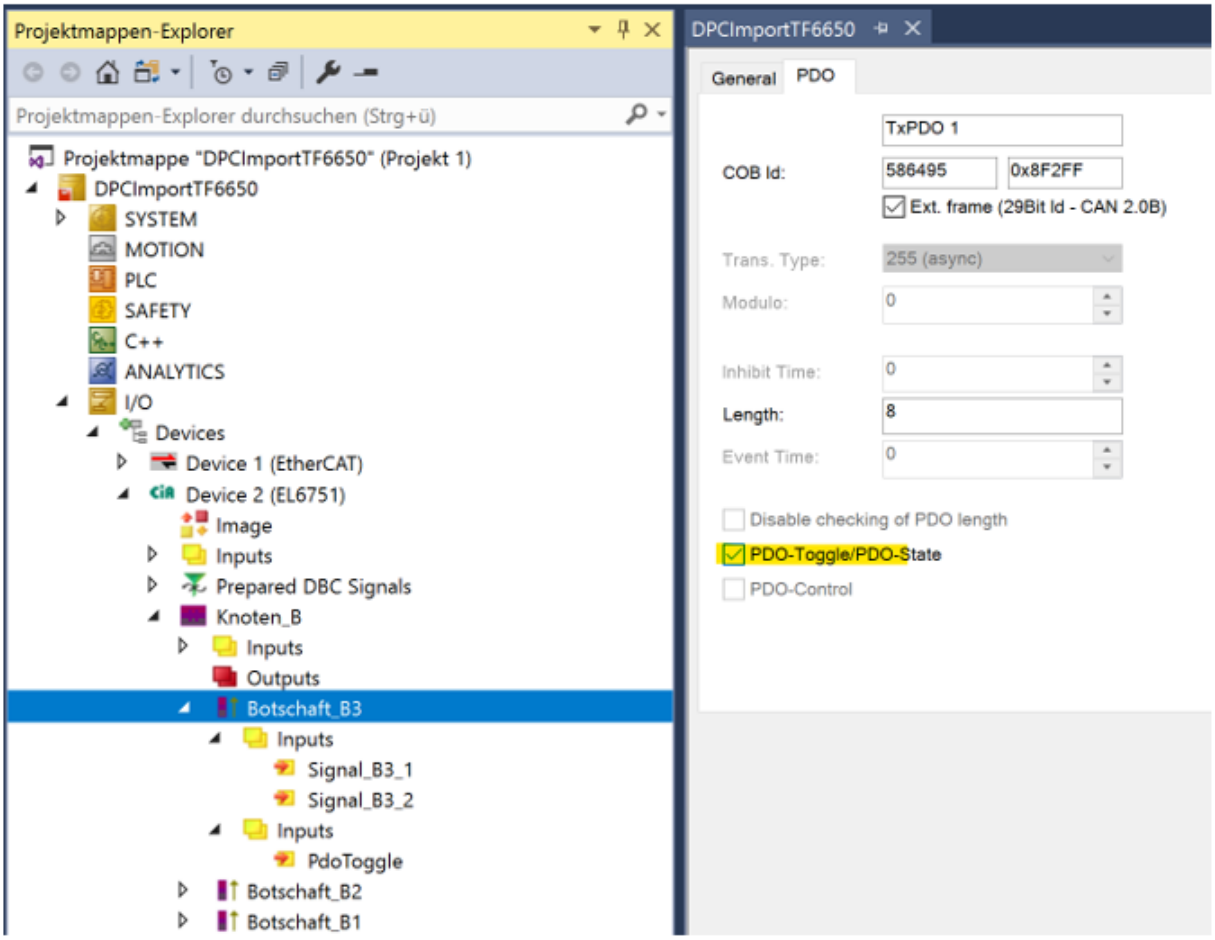
7 Monitoring of messages with the EL6751

The following settings need to be made in order to monitor the arrival of messages with the EL6751:

1. Switch the mapping of the terminal to **MDP Mapping** and select **PDO Toggle**

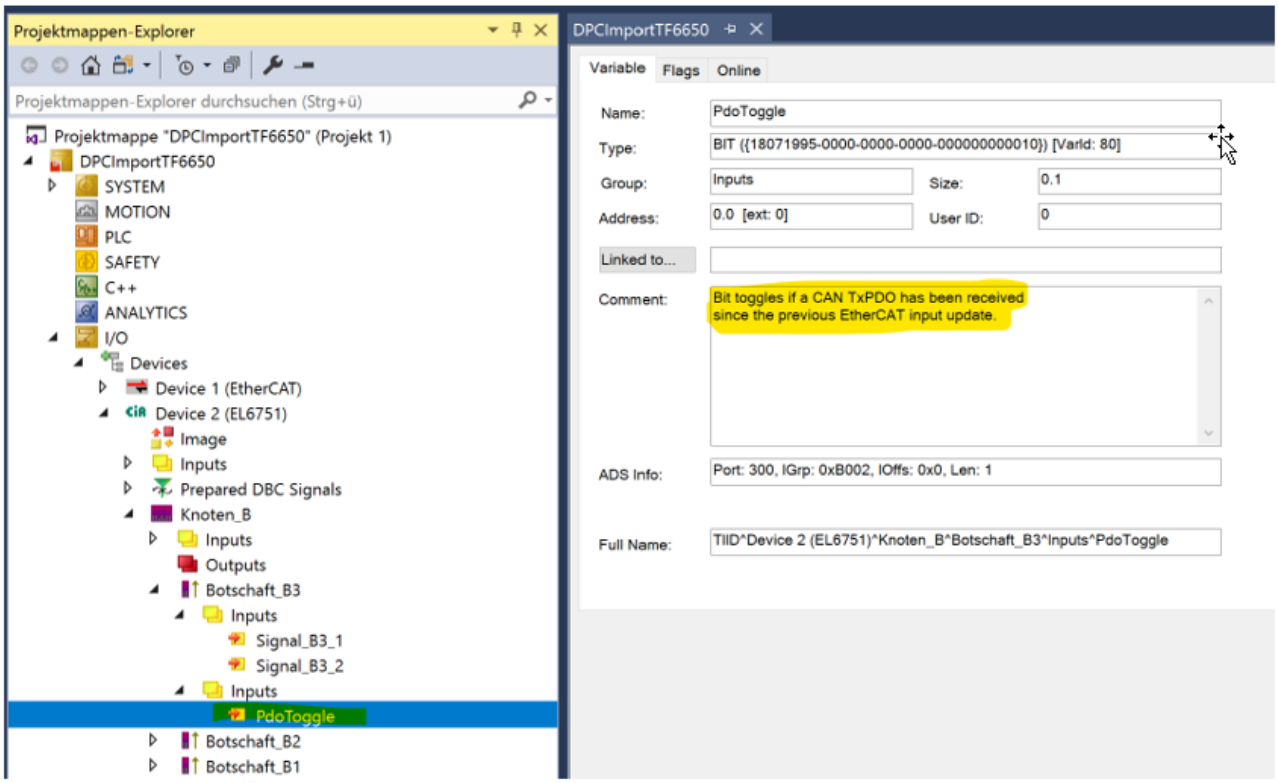


2. Activate the **PDO-Toggle/PDO State** option of the corresponding message with the RAW data



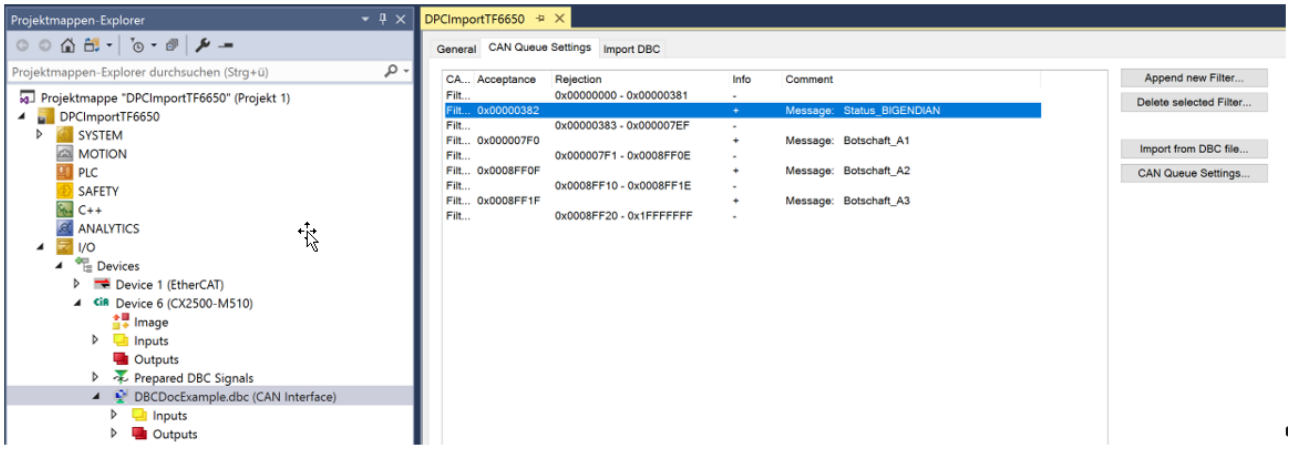
⇨ The Boolean variable **PDO-Toggle** is appended to the configuration.

PDO-Toggle always changes its state when the message has arrived at least once per EtherCAT cycle.


Projektmappen-Explorer
Projektmappen-Explorer durchsuchen (Strg+ü)
Projektmappe "DPCImportTF6650" (Projekt 1)
DPCImportTF6650
SYSTEM
MOTION
PLC
SAFETY
C++
ANALYTICS
I/O
Devices
Device 1 (EtherCAT)
Device 2 (EL6751)
Image
Inputs
Prepared DBC Signals
Knoten_B
Inputs
Outputs
Botschaft_B3
Inputs
Signal_B3_1
Signal_B3_2
Inputs
PdoToggle
Botschaft_B2
Botschaft_B1
DPCImportTF6650
Variable
Flags
Online
Name:
PdoToggle
Type:
BIT ({18071995-0000-0000-0000-000000000010}) [VarId: 80]
Group:
Inputs
Size:
0.1
Address:
0.0 [ext: 0]
User ID:
0
Linked to...
Comment:
Bit toggles if a CAN TxPDO has been received since the previous EtherCAT input update.
ADS Info:
Port: 300, IGrp: 0xB002, IOffs: 0x0, Len: 1
Full Name:
TIID^Device 2 (EL6751)^Knoten_B^Botschaft_B3^Inputs^PdoToggle

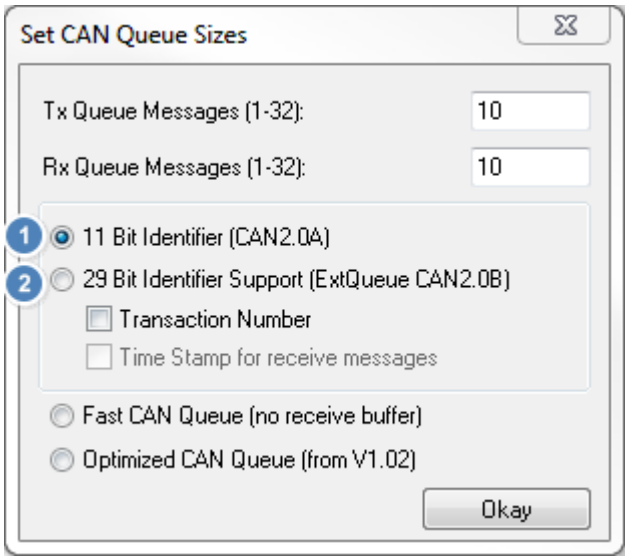
8 CAN interface

For simpler parameterization of the CAN filter with the CAN interface, the import of Message IDs from a DBC file is possible.



CAN interface of a CX2500-M510 – CAN queue settings with DBC Import button

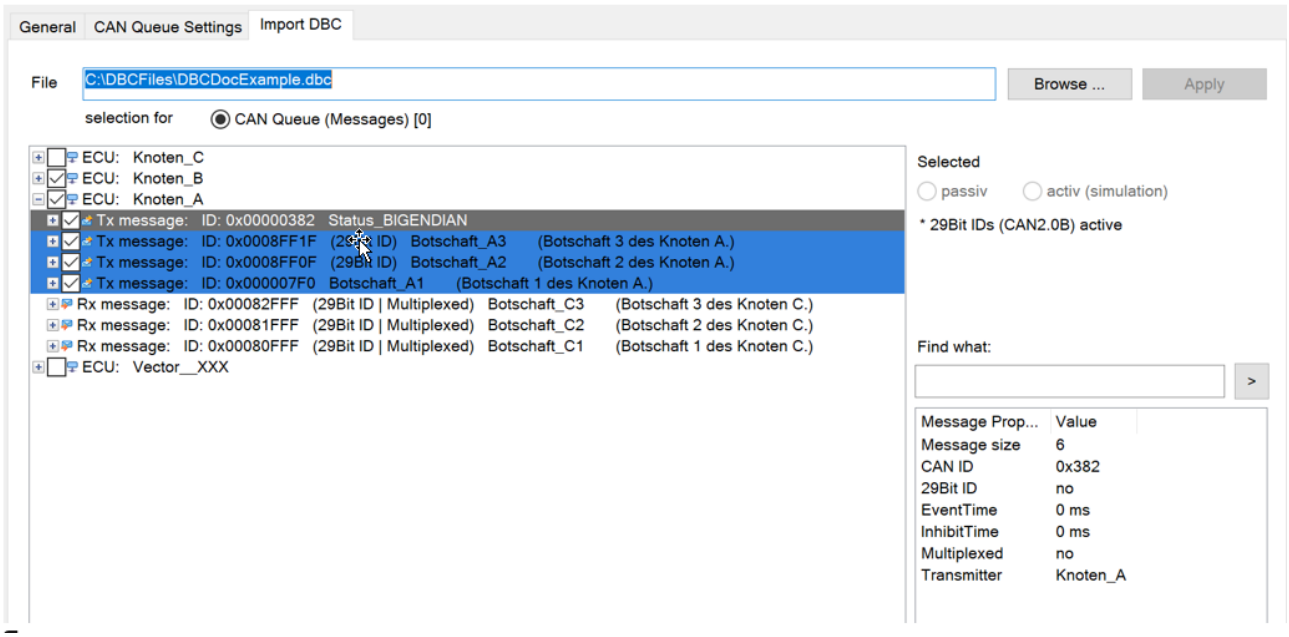
8.1 Acceptance of Message IDs from DBC files for the filter



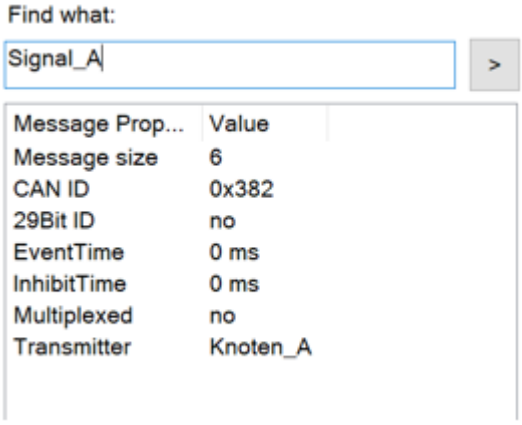
In the 11-bit (CAN2.0A) mode 1 **as well as** in the 29-bit (CAN2.0B) mode 2 of the CAN interface, Message IDs can be imported via the property page "CAN Queue Settings" under the **Import from DBC File** button.

8.2 Import tab

The Import tab implements the selection of the DBC file to be imported as well as the display of the contents and the selection of the messages. In 11-bit mode of the queue, only messages with standard CAN IDs are displayed. In 29-bit mode, Extended CAN IDs are also shown.



You can search for any text (sub-text) within the loaded DBC file via the **Find what:** field.

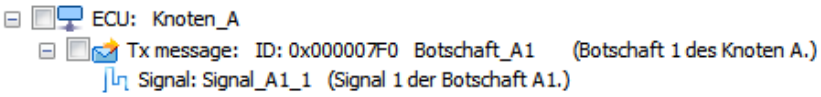


The selection bar jumps to the line with the first match and to further matches if the search is repeated with the same search text.

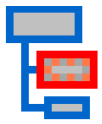
8.2.1 Representation of the DBC contents

On the **Import** tab, the nodes of the DBC file are displayed in the following hierarchy:

- Node (ECU)
  - Message



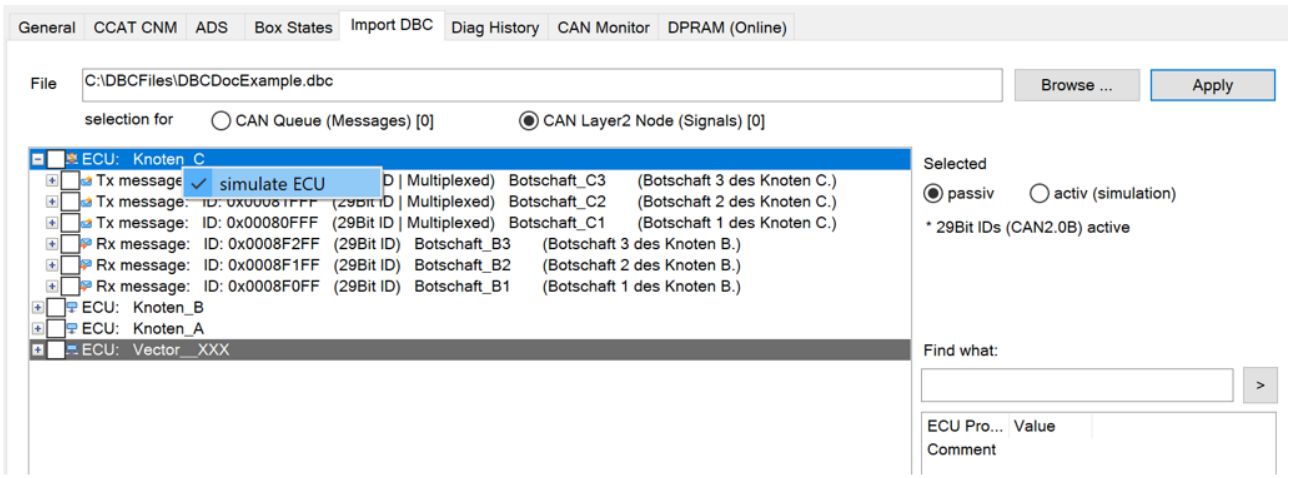
Signal

Level	Symbol	Meaning	Structure of the text
Node (ECU)		Network device	ECU: %*Name of the node*%
Message		CAN message transmitted by the associated node	Tx message: ID: %*id*% [*(29-bit ID | Multiplexed)*] %*Name of the message*% [%*(Comment)*%]
Signal		Signal within the assigned CAN message	Signal: %*Name of the signal*% [%*(Comment)*%]
Further symbols:			
Rx message		CAN message received from the associated node	Rx message: ID: %*id*% [*(29-bit ID | Multiplexed)*] %*Name of the message*% [%*(Comment)*%]
Signal (Multiplexer)		Multiplexer switch signal	Multiplexer signal: %*Name of the signal*% [%*(Comment)*%]
Signal (multiplexed)		Multiplexed signal	Multiplexed signal: (at value %*Multiplexor*%) %*Name of the signal*% [%*(Comment)*%]
Node (simulated)		Network device that can be simulated.	ECU: %*Name of the node*%

*data placed in [ ] are only displayed if they apply or exist in the DBC file.

8.2.2 Selection of messages for import

Only messages that are selected are imported when clicking the **Apply** button. The selection is made via the **checkboxes** in front of the message lines.



The checkboxes in front of a node (ECU) (de)select all selectable node messages.

A selected message is additionally marked with a blue background if it is intended to be imported into the CAN Queue.

The Property Sheet can also be used to import CAN-Layer2 nodes, in which case the selected messages are marked with a green background (see below).

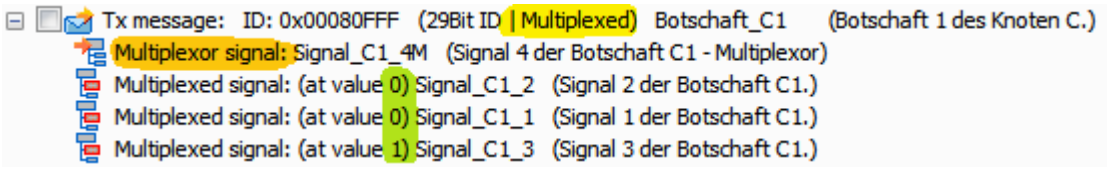
The number of selected messages is displayed in the header.

In the case of verbose comments for the messages and signals, the size of the Property Sheet can be adjusted for greater clarity (**Resize**).

The selection can take place via the mouse and the keyboard (arrow keys and space bar).

The queue messages can be displayed in filter mode, but not changed. In the case of the queue, the selection from last time remains saved as long as only the queue is deleted and not the device.

8.2.3 Multiplexed signals



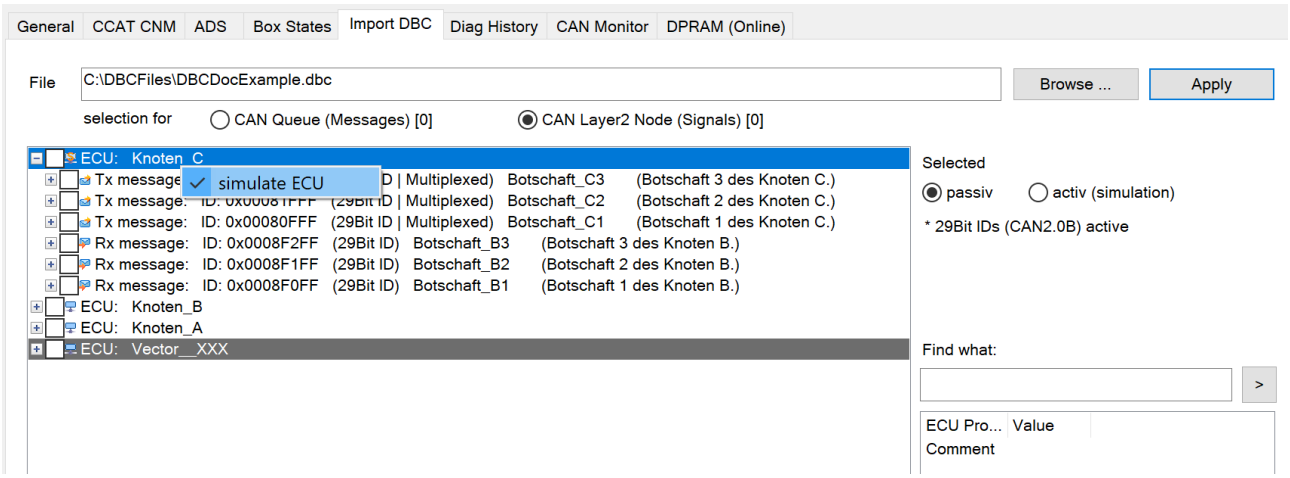
Messages containing multiplexed signals are marked with **Multiplexed**.

In the case of a received message, the value of the multiplexer signal indicates which of the listed signals the message contains (see: *(*at value x*)*).

8.2.4 Marking a node for simulation

So that a node can be simulated by the PLC, the messages that it sends must be imported as outputs. Messages that it receives must be imported as inputs. This corresponds precisely to the reverse direction of the standard case, if the configuration of the nodes is only imported for observation. In this standard case, messages that a node transmits in the network are imported as inputs. Messages that the node receives in the standard case are ignored (not imported), since they could be transmitted by any other node of the network and could thus be imported via this node.

Simulate-ECU direction reversal (input to output) and enablement of the RX signals:



A node is marked for simulation by right-clicking on it (context menu). Alternatively, the **passive** and **active (simulation**) radio buttons adjacent to the tree can be used to switch.

This node is then marked accordingly with a different symbol. The Rx messages previously marked only for information can now also be selected for the import.

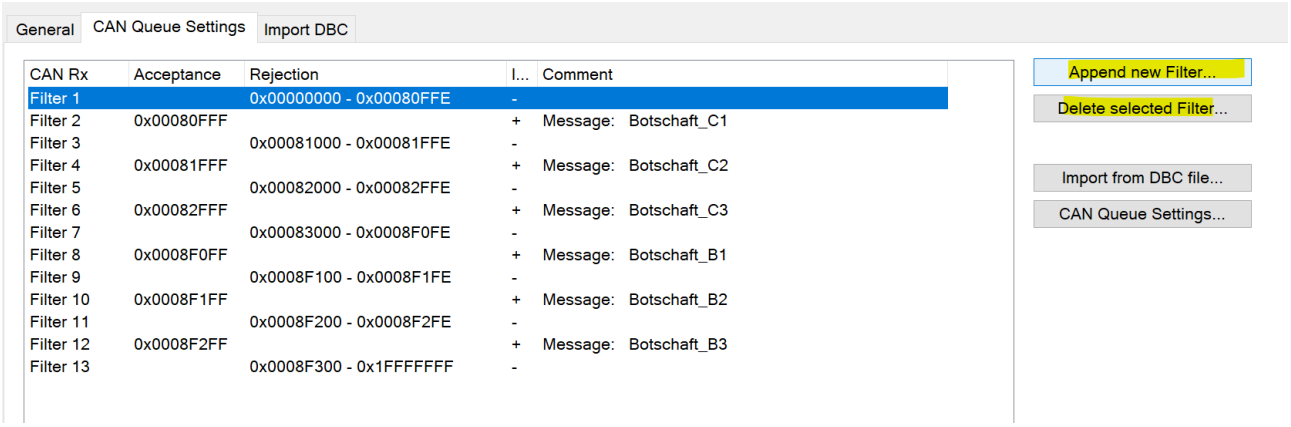
8.3 Manual parameterization of individual filters

The receiving filter of the CAN interface can also be manually set or deleted after an import.

Deleting filters

Filters must first be marked in the list before they can be removed from it. A multiple selection is possible via keyboard or mouse (range selection).

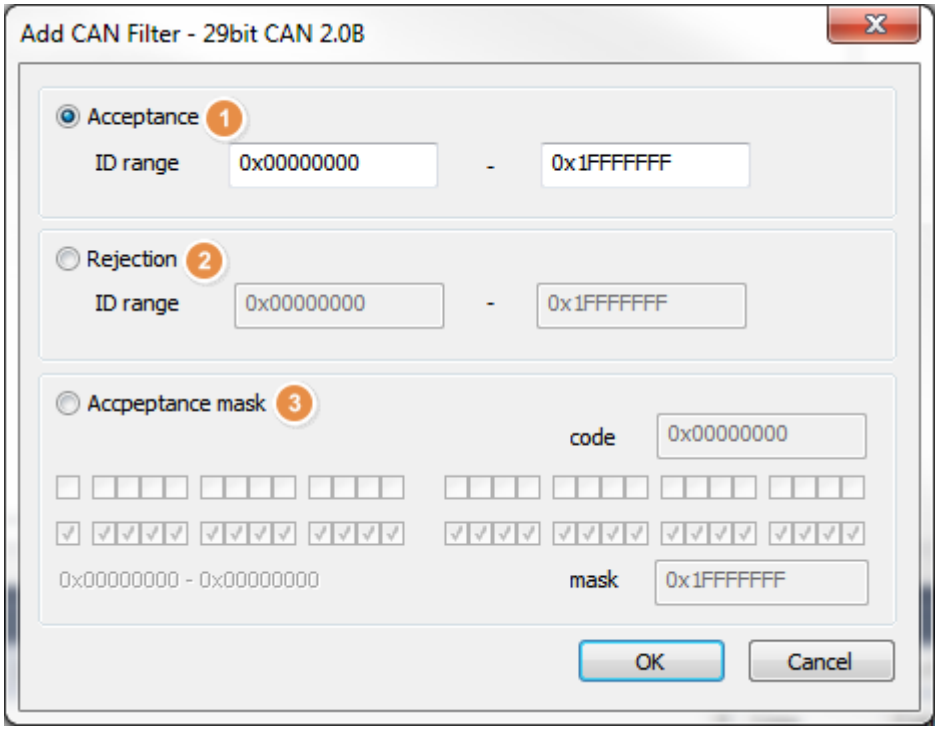
Finally, the **Delete selected Filter** button must be pressed to confirm the deletion.



Adding filters

Filters are appended via the **Append new Filter** button.

8.3.1 Filter selection



Filters can be entered as band-pass filters 1, as band-stop filters 2 or via a mask 3.

Band-pass and band-stop filters are entered as ranges.

8.3.2 The mask input

The filter mask is used to define which bits of the CAN identifier are to be compared with the assigned filter.

If a mask bit is not set, the corresponding bit of the CAN-Id is accepted irrespective of whether the filter code is 0 or 1.

If a mask bit is set, the corresponding bit of the CAN-Id is compared with the value of the filter code. If both are the same, the message is accepted, if they differ the message is rejected.

Sample 1)

Only the message with the ID 0x1200 is to be accepted.
Filter code and mask for acceptance of the ID 0x1200:

All bits must be compared, all bits of the mask to 1
The filter code must be set to the desired ID

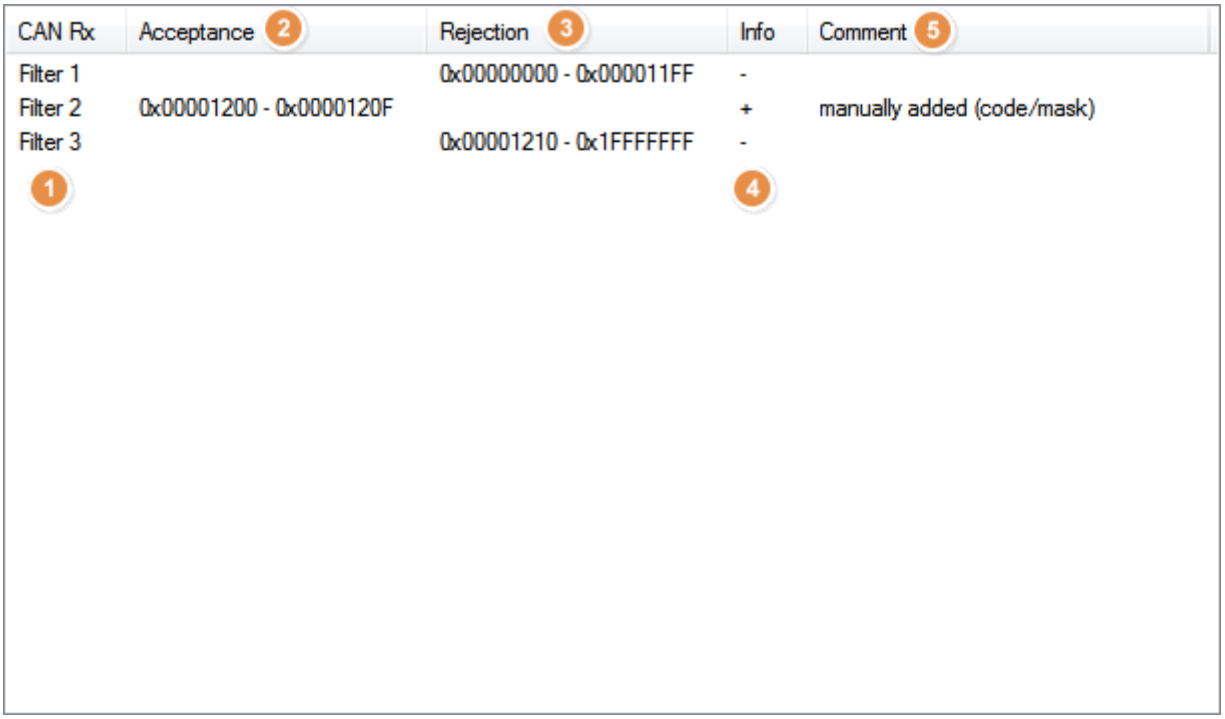
Sample 2)

Filter code and mask for acceptance of the range ID 0x1200 – ID 0x120F:

The messages from 0x1200 to 0x120F are to be accepted:
In order to permit a range, the bits of the mask that mark the range are not set.
The filter code can be set to any desired ID within the range

8.3.3 Tabular illustration of the filter IDs and comments

Tabular overview of the receiving filters:



1 CA-Rx: Sequential numbering

2 Acceptance: Band-pass filter

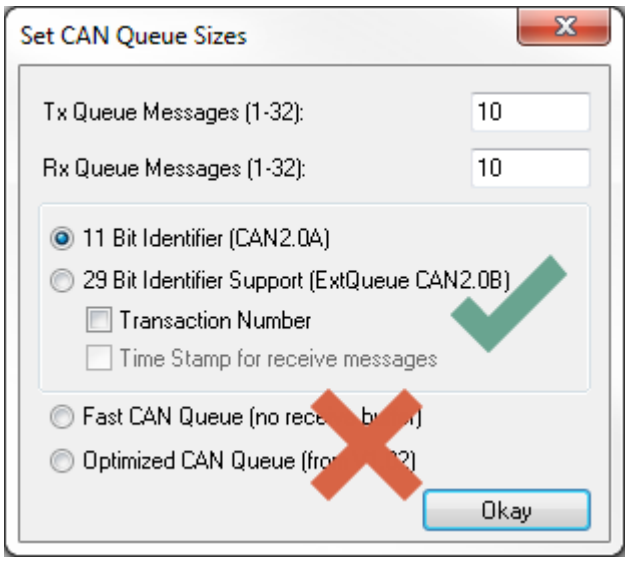
3 Rejection: Band-stop filter

4 Info: - for band-stop filter, + for band-pass filter

5 Comment

8.3.4 Application range of the filters

Receive filters can be used in the 11-bit and 29-bit modes of the queue:



The receive filter is not available in the **Fast CAN Queue** and **Optimized CAN Queue** modes.

8.3.5 Maximum number of filter entries

255 filter entries are supported.

8.4 Necessary firmware version

The EL6751 supports the 29-bit filters from the firmware version:

- FW16(V1.14)

TwinCAT Version 4024.12 or higher is to be used when using the M510 CAN optional interface with the Embedded PCs or the FC512x.
The version of the CCAT driver should be V1.26 or higher:

More Information:
www.beckhoff.com/tf6650

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