

BECKHOFF New Automation Technology

Manual | EN

TE1410

TwinCAT 3 | Interface for MATLAB®/Simulink®

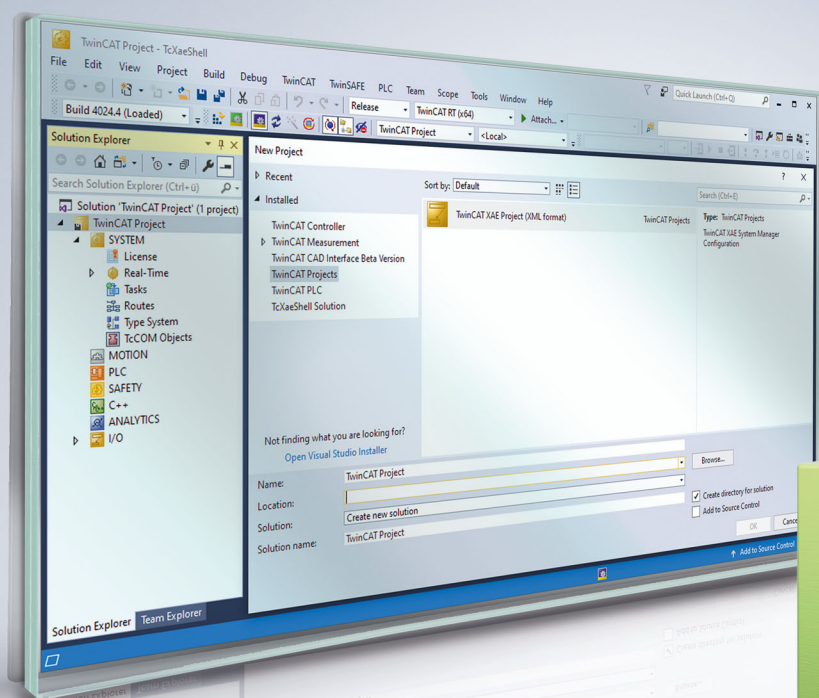


Table of contents

1 Foreword	5
1.1 Notes on the documentation	5
1.2 Safety instructions	6
1.3 Notes on information security.....	7
2 Up to version 1.1.xxxx.x	8
2.1 Overview	8
2.2 Installation	9
2.3 Licenses	10
2.4 TE1410 Simulink Library	10
2.4.1 Asynchronous Blocks.....	11
2.4.2 Synchronous Blocks.....	15
2.4.3 Utilities.....	24
3 From version 2.x.xxxx.x	25
3.1 Overview	25
3.2 Installation	25
3.3 Licenses	26
3.4 Short introduction: TwinCAT ADS.....	26
3.5 ADS in MATLAB®.....	28
3.5.1 API Documentation	28
3.5.2 Basic concepts	31
3.6 ADS in Simulink®.....	31
3.6.1 ADS blocks.....	31
3.7 Samples	41

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

© Beckhoff Automation GmbH & Co. KG, Germany.

The reproduction, distribution and utilization of this document as well as the communication of its contents to others without express authorization are prohibited.

Offenders will be held liable for the payment of damages. All rights reserved in the event of the grant of a patent, utility model or design.

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>.

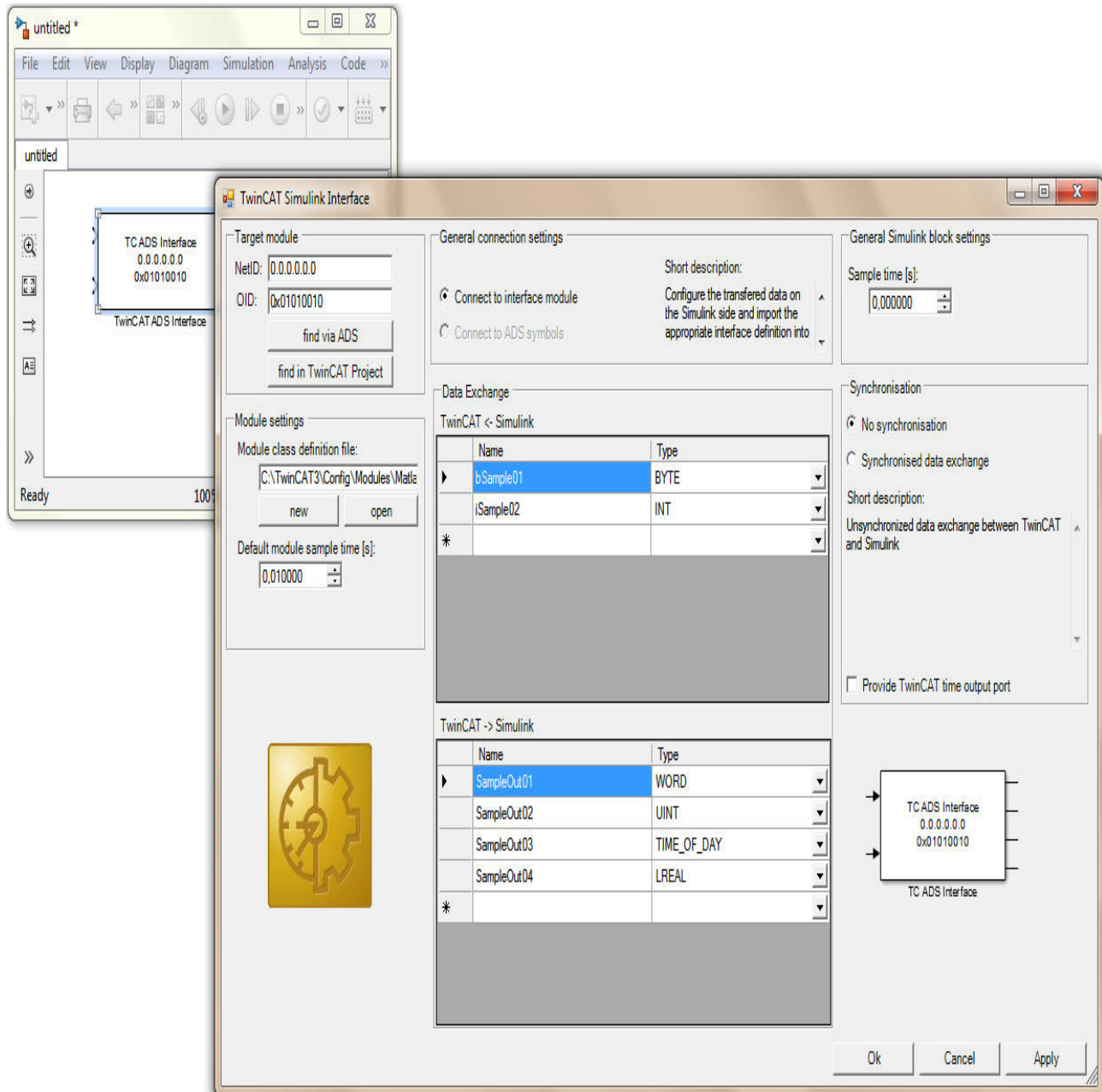
2 Up to version 1.1.xxxx.x

2.1 Overview

TE1410 interface for MATLAB®/Simulink®

TE1410 interface for MATLAB®/Simulink® can be used for data exchange between TwinCAT3 and MATLAB®/Simulink®.

The data is exchanged via ADS using Simulink blocks, which are provided in a simulink library.



Webinars concerning TE1400 and TE1410

Date	Topic	Referent
01.10.2013	TwinCAT 3 Matlab®/Simulink® -integration: Introduction, application examples, TC3 Interface for Matlab®/Simulink® (german)	Dr. Knut Güttel

[Overview of the Beckhoff webinars \(german\)](#)

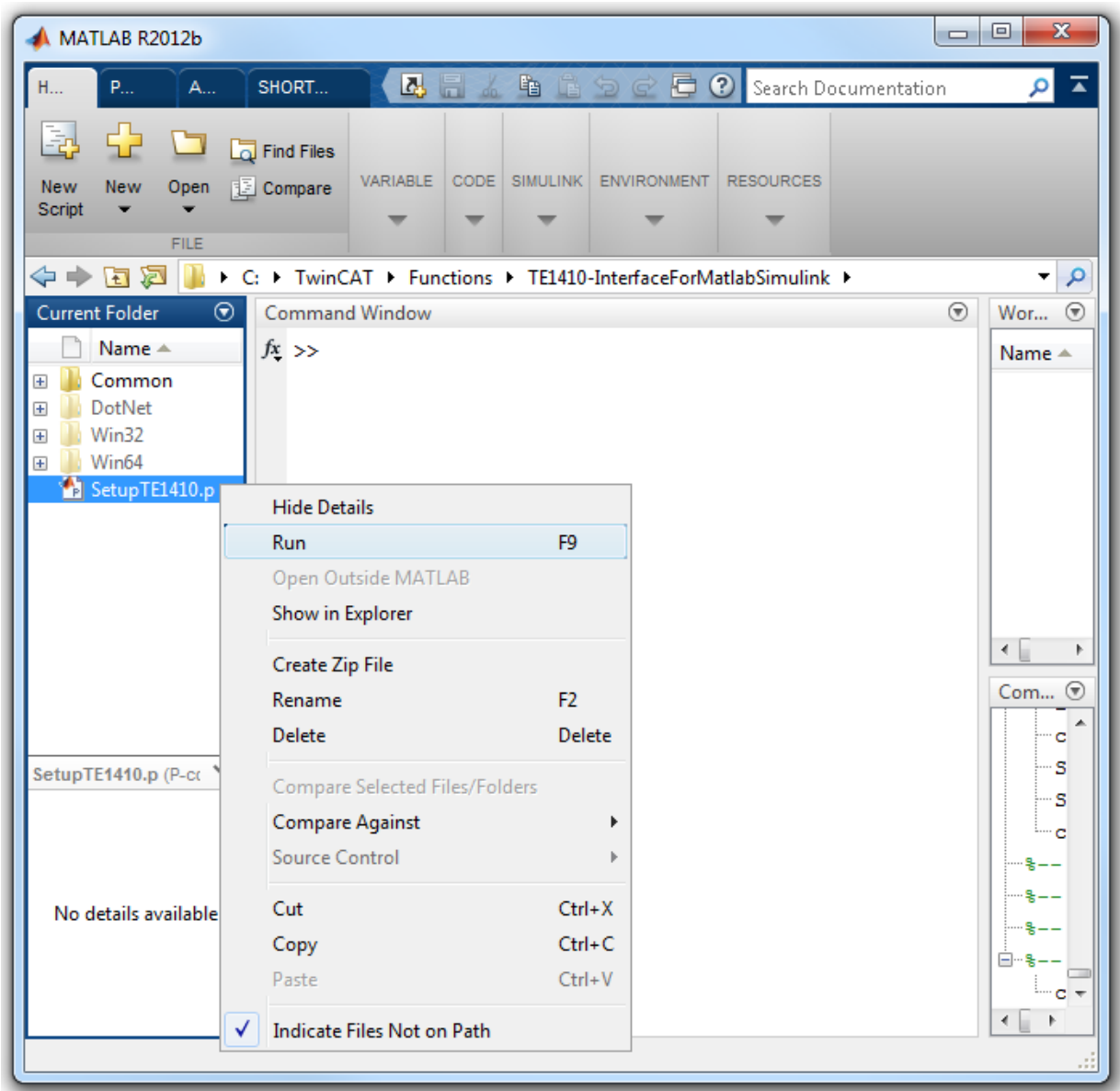
2.2 Installation

System requirements

- MATLAB®/Simulink® R2010a or newer.
- TwinCAT 3.0 or newer.

Installation instructions

1. Run the TwinCAT 3 Setup
2. Run the Setup "TE1410-InterfaceForMatlabSimulink"
3. Start MATLAB® as administrator and execute "%TwinCAT3Dir%..\Functions\TE1410-InterfaceForMatlabSimulink\SetupTE1410.p" in MATLAB®.



NOTE**Files cannot be found**

When executing MATLAB® without Administrator permission, on a system with User Account Control (UAC) enabled, the MATLAB® path can't be saved durable. In this case "SetupTE1410.p" has to be executed after each restart of MATLAB®, otherwise some files cannot be found, which are necessary to generate TwinCAT modules.

2.3 Licenses

To use the functions of the TE1410 interface for MATLAB®/Simulink®, a TE1410 license is required on the system on which MATLAB®/Simulink® is operated with the TE1410 Simulink function blocks. In addition, at least a TwinCAT ADS installation is required on this computer, in order to make the ADS router available.

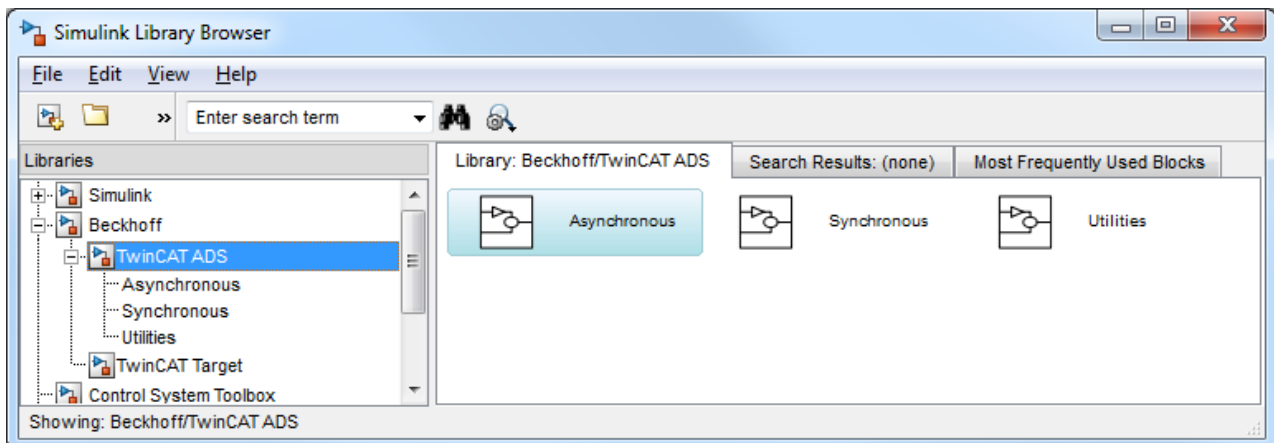
From this system, connections to various target platforms can then be established. No additional runtime licenses are required for using the TE1410 on the target platforms.

Demo version

A 7-day trial license is not available for TE1410, since it is an engineering product. However, you can test the product with limited functionality (demo version). The limitations of the demo version are automatically triggered if no valid TE1410 license is found on the system. The functionality is limited to a total of 5 ADS variables and 5 ADS function blocks per Simulink model.

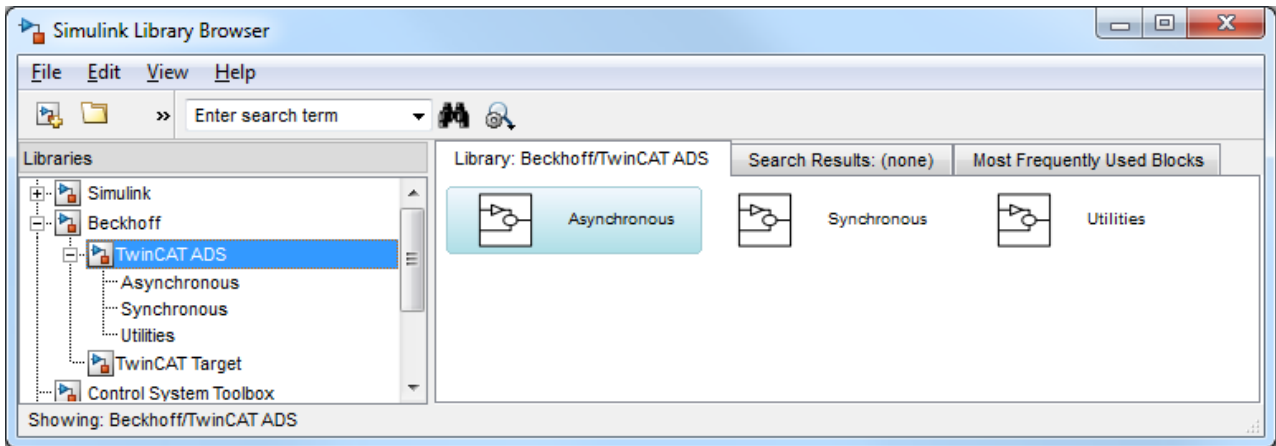
2.4 TE1410 Simulink Library

After TE1410 - Interface for MATLAB®/Simulink® was installed successfully, the Simulink library browser contains the entry Beckhoff / TwinCAT ADS:



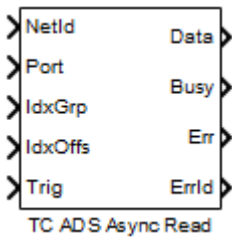
2.4.1 Asynchronous Blocks

TE1410 - Interface for MATLAB®/Simulink® was installed successfully, the Simulink library browser contains the entry Beckhoff / TwinCAT ADS:



2.4.1.1 TC ADS Async Read

The block "TC ADS Async Read" provides read access to TwinCAT ADS variables. You can find this block beneath the subfolder "Asynchronous" of the Beckhoff ADS Library inside the "Simulink Library Browser".



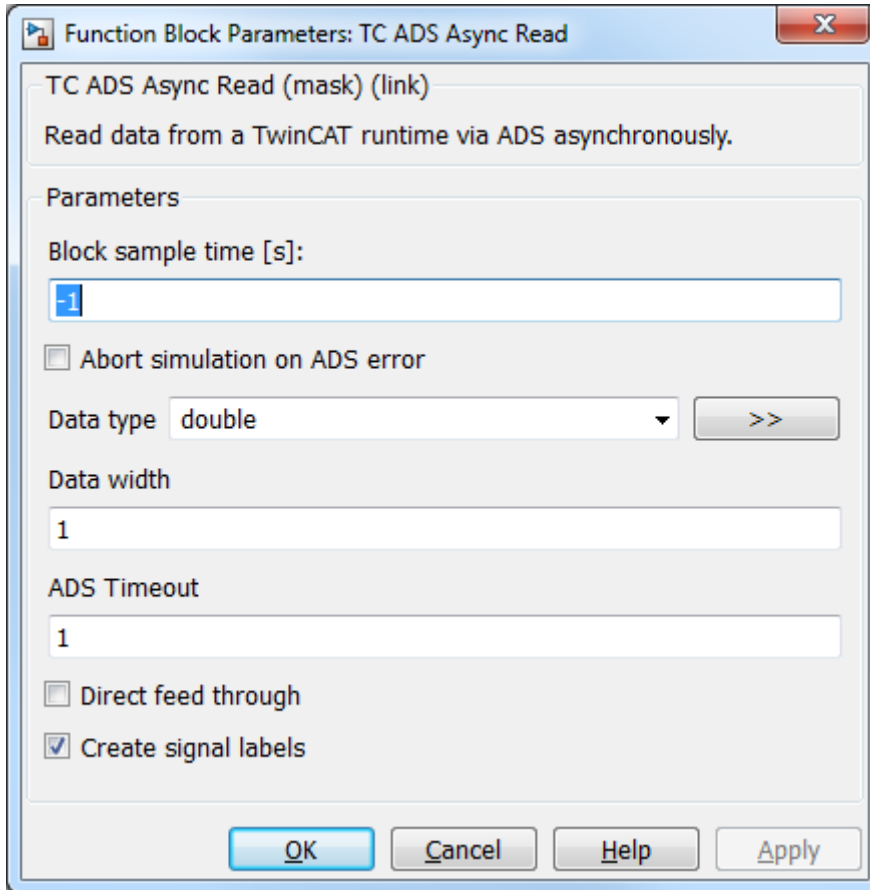
Block input ports

NetId	The target AMS NetId
Port	The target ADS Port
IdxGrp	The target ADS Index Group
IdxOffs	The target ADS Index Offset
Trig	Positive edge triggers the ADS request

Block output ports

Data	The value, read from the ADS variable
Busy	Boolean flag signaling that the block is currently waiting for an ADS response
Err	Boolean flag signaling an actual ADS error
ErrId	The ADS error code

Block parameters



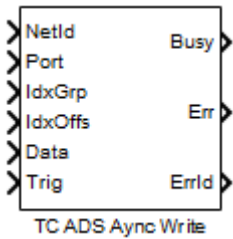
Block sample time	The Simulink block sample time, which is the ADS request interval (referred to the Simulink time but not to the real time)
Abort simulation on ADS error	Stop simulation at the first ADS error
Data type	The data type of the <i>Data</i> output signal, which should be the data type of the transmitted ADS variable
Data width	The width of the <i>Data</i> output signal. Use values greater than 1 to transmit arrays.
ADS timeout	Simulink will wait on a response of the last ADS request until this timespan is exceeded (in each time step)
Direct feed through	Evaluate the input values in the output method of the block. This ensures, that the values at the output ports in the current time step belong to the ADS parameters at the input ports of the same time step. Otherwise the output values belong to the input ADS parameters of the previous time step. Activating this option can significantly slow down the simulation.
Create signal labels	Signal labels are created for the signal lines, connected to the output ports

Code generation

Code generation is supported only for the TwinCAT Target.

2.4.1.2 TC ADS Async Write

The block "TC ADS Async Write" provides write access to TwinCAT ADS variables. You can find this block beneath the subfolder "Asynchronous" of the Beckhoff ADS Library inside the "Simulink Library Browser".



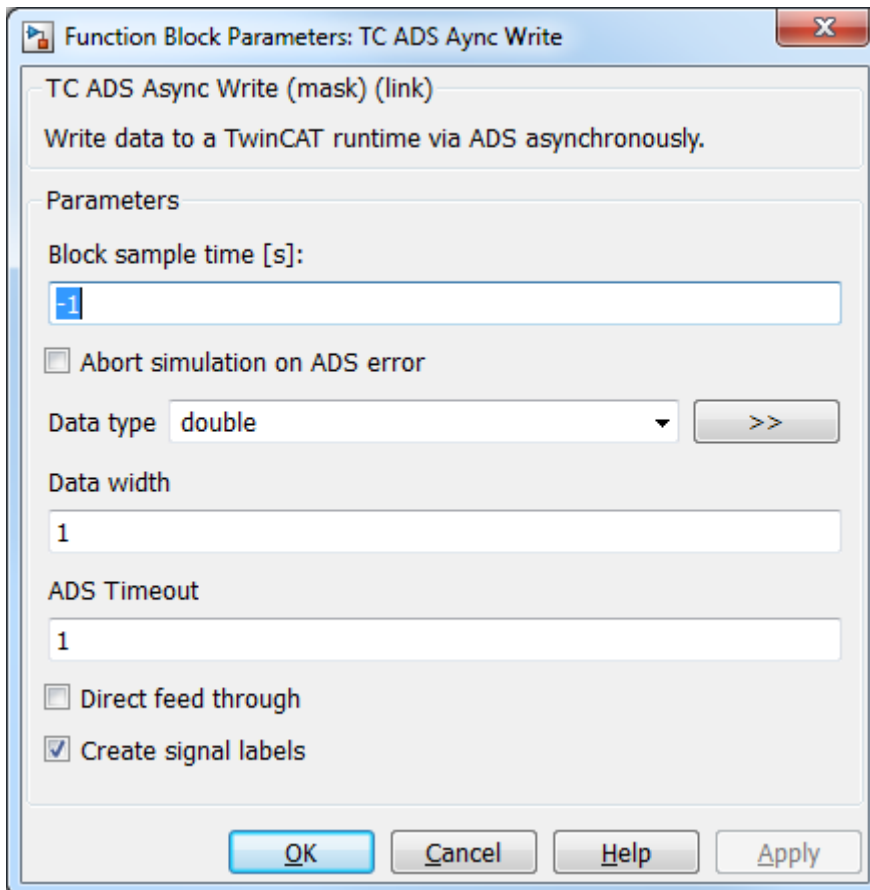
Block input ports

NetId	The target AMS NetId
Port	The target ADS Port
IdxGrp	The target ADS Index Group
IdxOffs	The target ADS Index Offset
Data	The value to be written to the ADS variable
Trig	Positive edge triggers the ADS request

Block output ports

Busy	Boolean flag signaling that the block is currently waiting for an ADS response
Err	Boolean flag signaling an actual ADS error
ErrId	The ADS error code

Block parameters



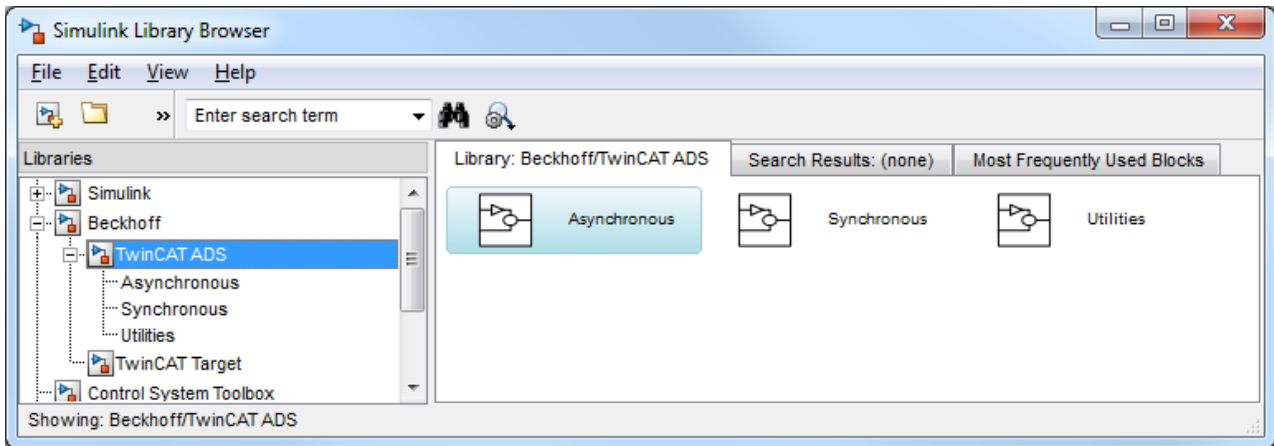
Block sample time	The Simulink block sample time, which is the ADS request interval (referred to the Simulink time but not to the real time)
Abort simulation on ADS error	Stop simulation at the first ADS error
Data type	The data type of the <i>Data</i> input signal, which should be the data type of the transmitted ADS variable
Data width	The width of the <i>Data</i> input signal. Use values greater than 1 to transmit arrays.
ADS timeout	Simulink will wait on a response of the last ADS request until this time span is exceeded (in each time step)
Direct feed through	Evaluate the input values in the output method of the block. This ensures, that the values at the output ports in the current time step belong to the ADS parameters at the input ports of the same time step. Otherwise the output values belong to the input ADS parameters of the previous time step. Activating this option can significantly slow down the simulation.
Create signal labels	Signal labels are created for the signal lines, connected to the output ports

Code generation

Code generation is supported only for the TwinCAT Target.

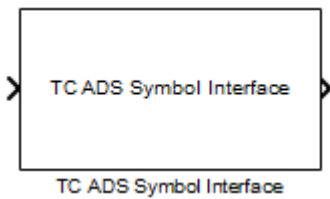
2.4.2 Synchronous Blocks

TE1410 - Interface for MATLAB®/Simulink® was installed successfully, the Simulink library browser contains the entry Beckhoff / TwinCAT ADS:



2.4.2.1 TC ADS Symbol Interface

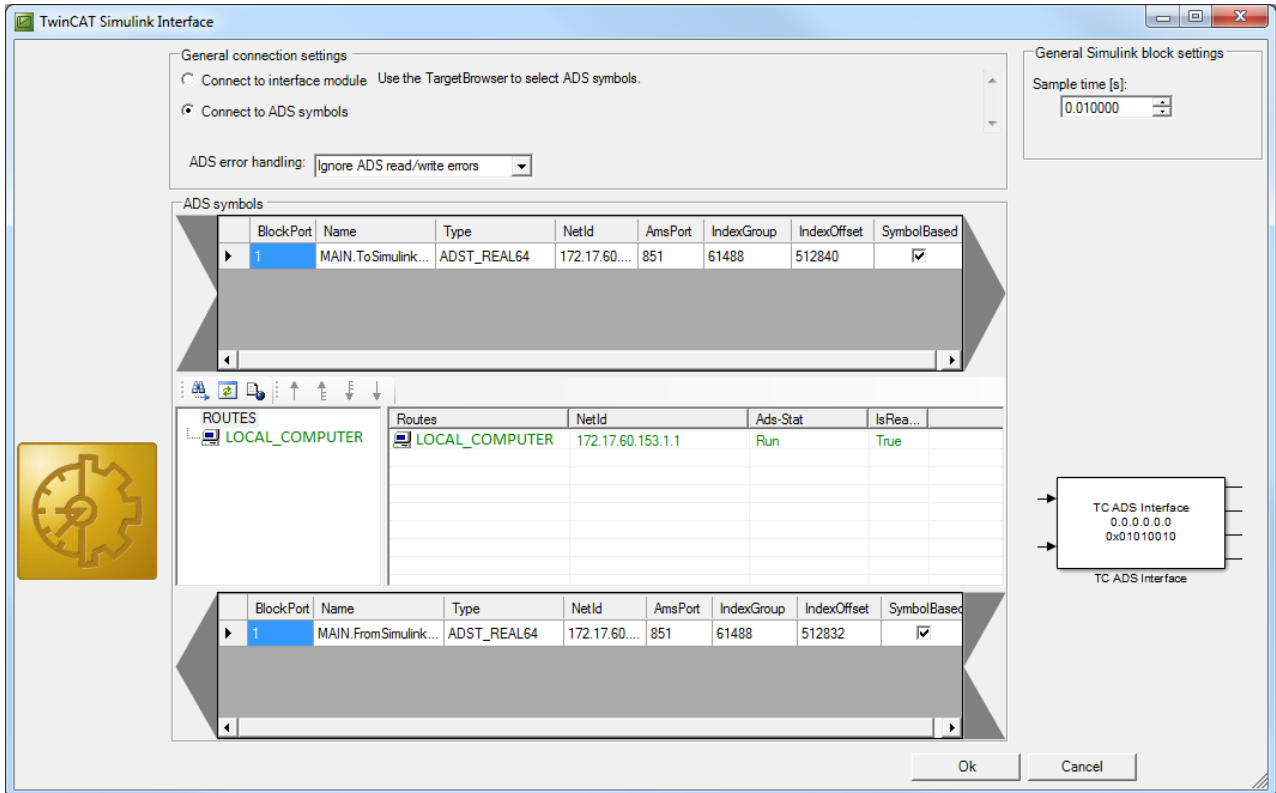
The block "TC ADS Symbol Interface" provides synchronous read and write access to a TwinCAT ADS symbols. You can find this block beneath the subfolder "Synchronous" of the Beckhoff ADS Library inside the "Simulink Library Browser".



Block input and output ports

The number of ports and the data types of the ports depend on the block configuration. There are no fix input or output ports.

Block parameters



ADS symbols

Define the input and output ports of the Simulink block. You can use the target browser to search for ADS symbols and put the symbols to the symbol list of the desired transmit direction. This can be done via the appropriate buttons of the target browser or via Drag and Drop.

General Simulink block settings

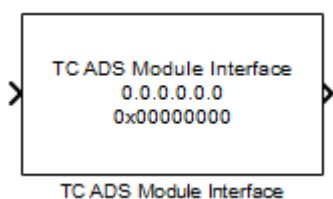
	Block sample time	The Simulink block sample time, which is the ADS request interval (referred to the Simulink time but not to the real time)
--	--------------------------	--

Code generation

Code generation from models using this block is supported for the TwinCAT Target. The generated TwinCAT module gets additional Data Areas (Process Images), which contain the input and output variables of the Simulink block.

2.4.2.2 TC ADS Module Interface

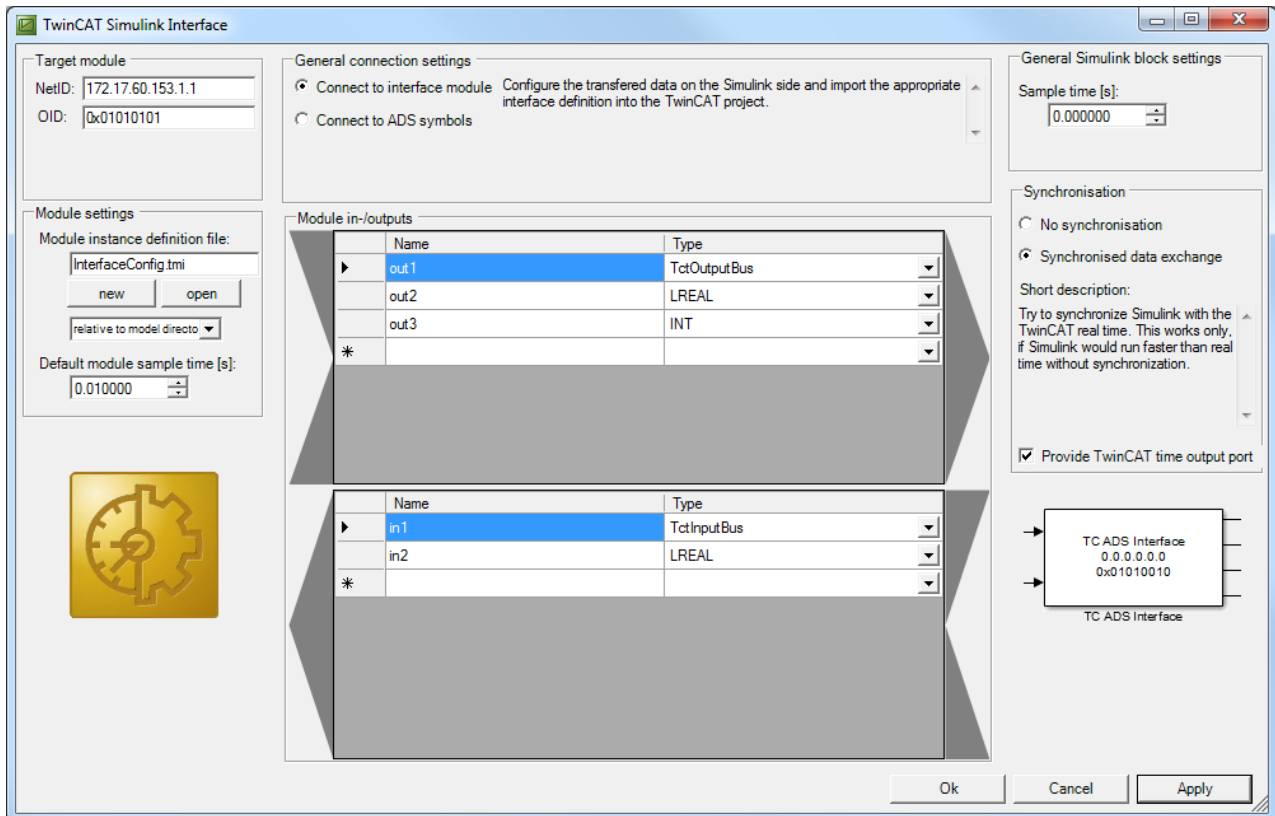
The block "TC ADS Module Interface" provides synchronous read and write access to a special TwinCAT module. You can find this block beneath the subfolder "Synchronous" of the Beckhoff ADS Library inside the "Simulink Library Browser".



Block input and output ports

The number of ports and the data types of the ports depend on the block configuration. There are no fix input or output ports.

Block parameters



Target module		
	NetId	The NetId of the target system, where the interface module instance is running.
	OID	The object ID of the module instance.

Module settings		
	Module instance definition file	Create or open a TMI (TwinCAT module instance) file, containing the interface module configuration. This file can be imported to the TwinCAT project.
	Default module sample time	The default sample time of the interface module.

Module in-/outputs		
Define the input and output ports of the Simulink block. These settings are saved to the selected TMI (TwinCAT module instance) file. The signals can be of any base type or a Simulink Bus Object, which is defined in the MATLAB workspace.		

General Simulink block settings		
---------------------------------	--	--

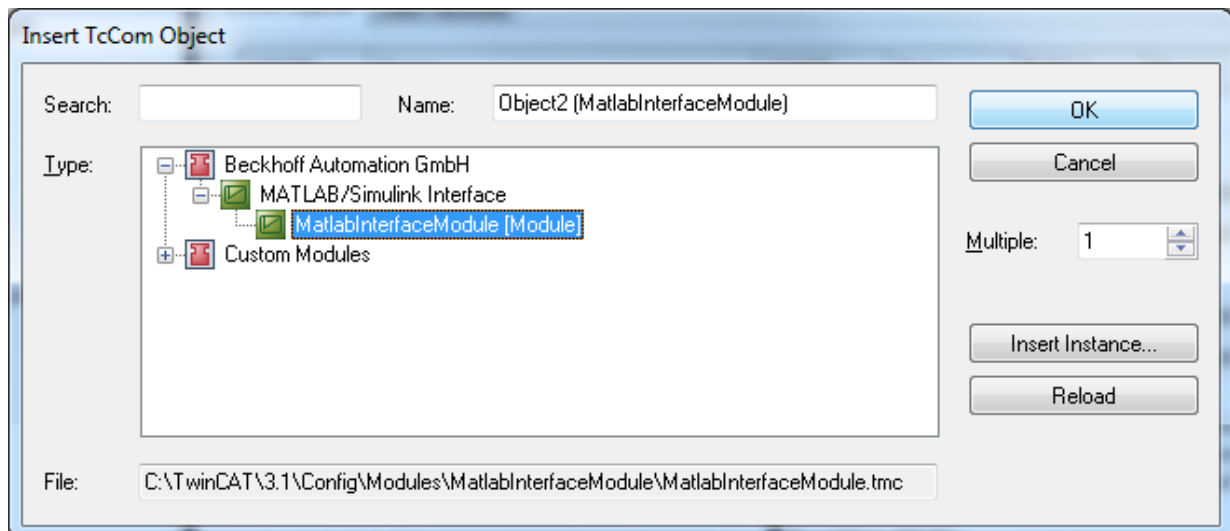
	Block sample time	The Simulink block sample time, which is the ADS request interval (referred to the Simulink time but not to the real time)
--	--------------------------	--

Synchronisation

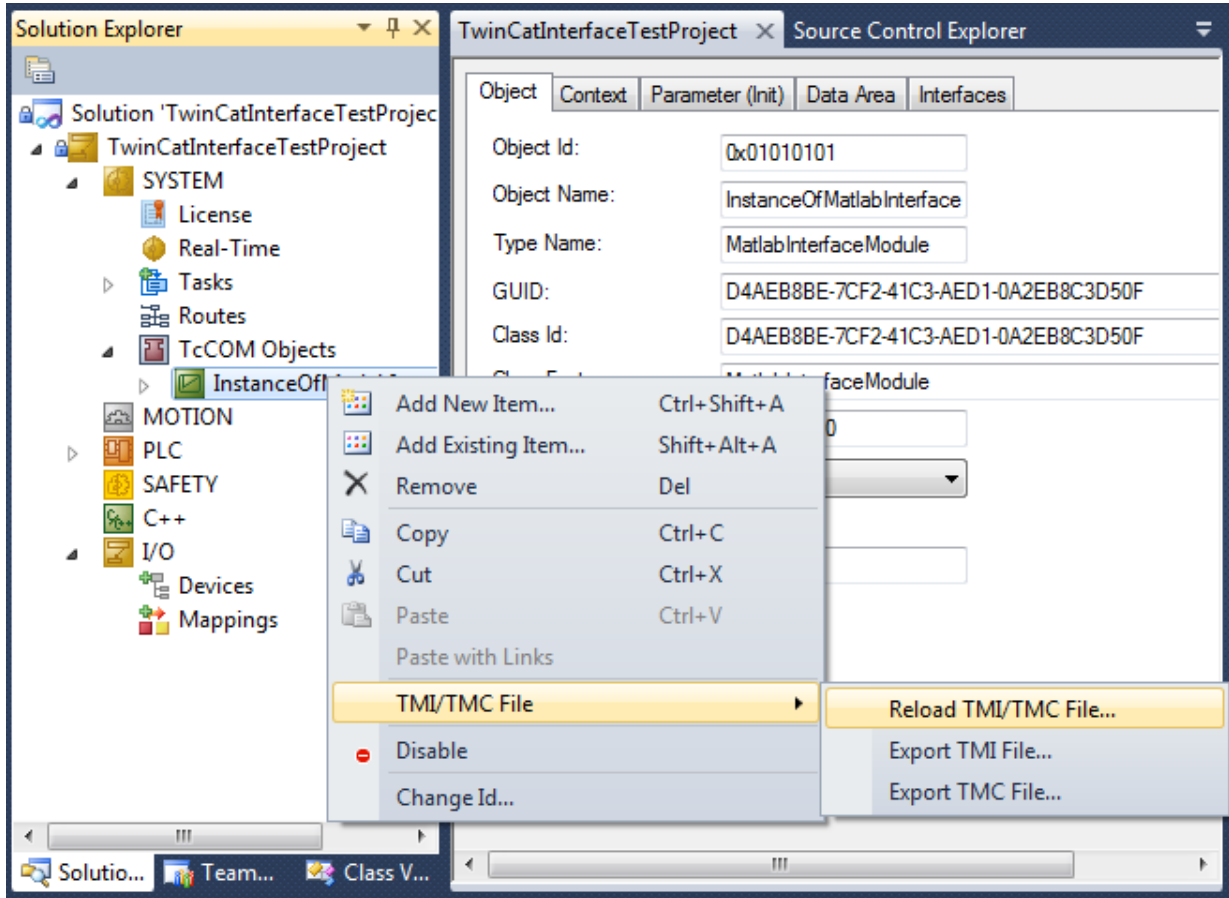
	No synchronization / Synchronized data exchange	The real time span from the start of the simulation is transmitted from the interface module to Simulink and can optionally be used to slow down the simulation time to the real time.
	Provide TwinCAT time output port	Provide the real time span from the start of the simulation as an output port.

Interface module configuration

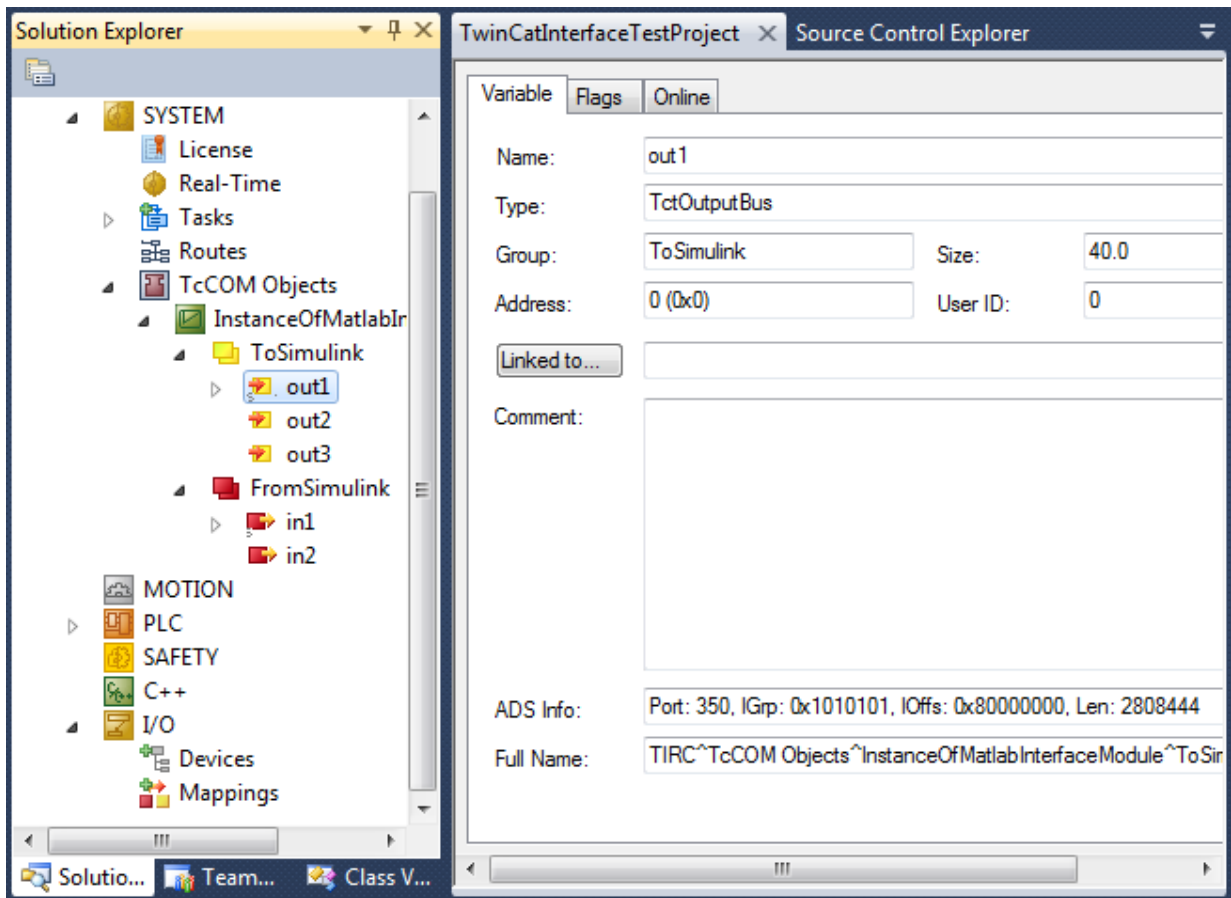
1. Open the TwinCAT project you want to connect to
2. Create a new instance of the **MatlabInterfaceModule** at the TwinCAT projects node "**System / TcCom Objects**" via its context menu item "**Add new item...**"



- Open the context menu of the newly created object node and **Reload the TMI file**. Open the TMI file that was configured in the Simulink block configurator.



4. Link the process image variables of the interface module to the desired process images of other TwinCAT modules (PLC, C++, MATLAB/Simulink, IOs). The objects input Data Area contains the Simulink block outputs and vice versa.

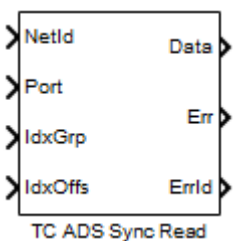


Code generation

Code generation from models using this block is supported for the TwinCAT Target. The generated TwinCAT module gets additional Data Areas (Process Images), which have the same structures as the Data Areas of the interface module instance.

2.4.2.3 TC ADS Sync Read

The block "TC ADS Sync Read" provides read access to TwinCAT ADS variables. You can find this block beneath the subfolder "Synchronous" of the Beckhoff ADS Library inside the "Simulink Library Browser".



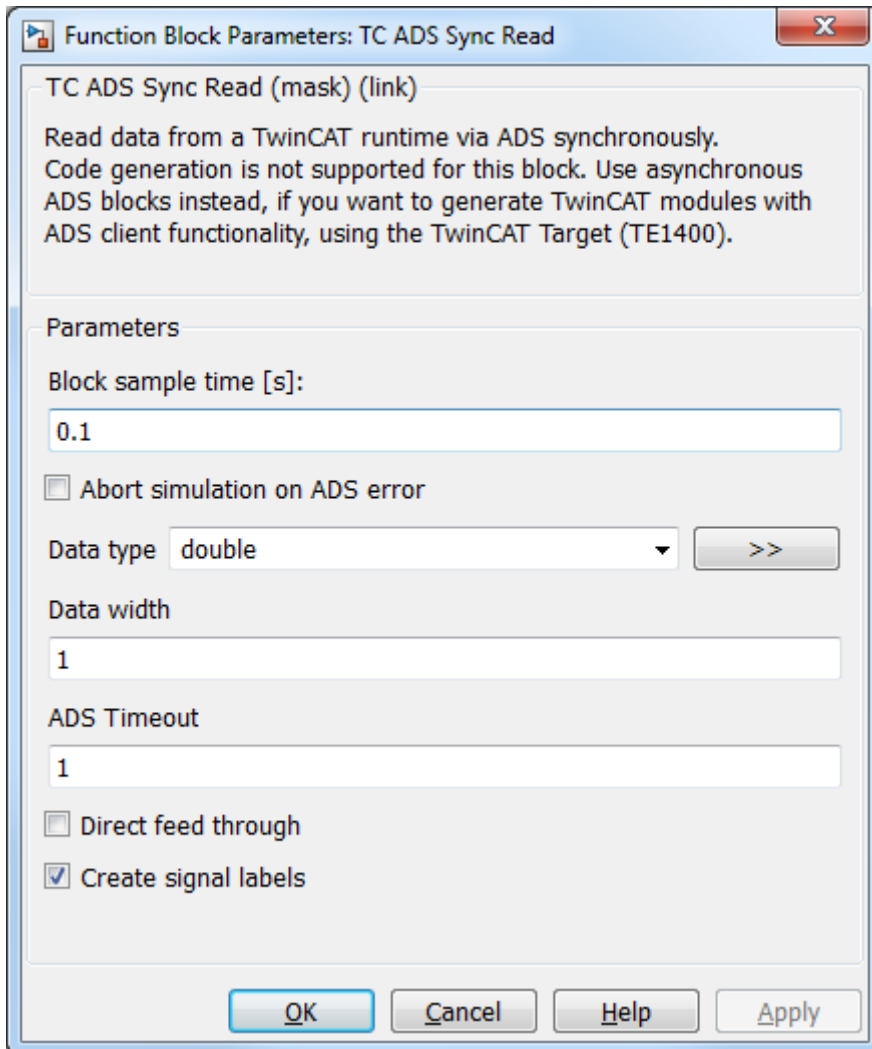
Block input ports

NetId	The target AMS NetId
Port	The target ADS Port
IdxGrp	The target ADS Index Group
IdxOffs	The target ADS Index Offset

Block output ports

Data	The value, read from the ADS variable
Err	Boolean flag signaling an actual ADS error
ErrId	The ADS error code

Block parameters



Block sample time	The Simulink block sample time, which is the ADS request interval (referred to the Simulink time but not to the real time)
Abort simulation on ADS error	Stop simulation at the first ADS error
Data type	The data type of the <i>Data</i> output signal, which should be the data type of the transmitted ADS variable
Data width	The width of the <i>Data</i> output signal. Use values greater than 1 to transmit arrays.
ADS timeout	Simulink will wait on a response of the last ADS request until this time span is exceeded (in each time step)
Direct feed through	Evaluate the input values in the output method of the block. This ensures, that the values at the output ports in the current time step belong to the ADS parameters at the input ports of the same time step.

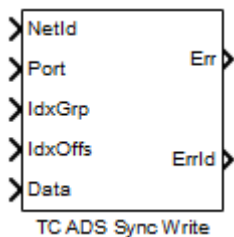
	Otherwise the output values belong to the input ADS parameters of the previous time step. Activating this option can significantly slow down the simulation.
Create signal labels	Signal labels are created for the signal lines, connected to the output ports

Code generation

Code generation is not supported. The reason is, that synchronous ADS communication requires to wait for the ADS responses. Due to nondeterministic ADS response times, this could lead to big jitter and cycle time exceeding in real time applications. Use asynchronous ADS blocks for use in real time applications instead.

2.4.2.4 TC ADS Sync Write

The block "TC ADS Sync Write" provides write access to TwinCAT ADS variables. You can find this block beneath the subfolder "Synchronous" of the Beckhoff ADS Library inside the "Simulink Library Browser".



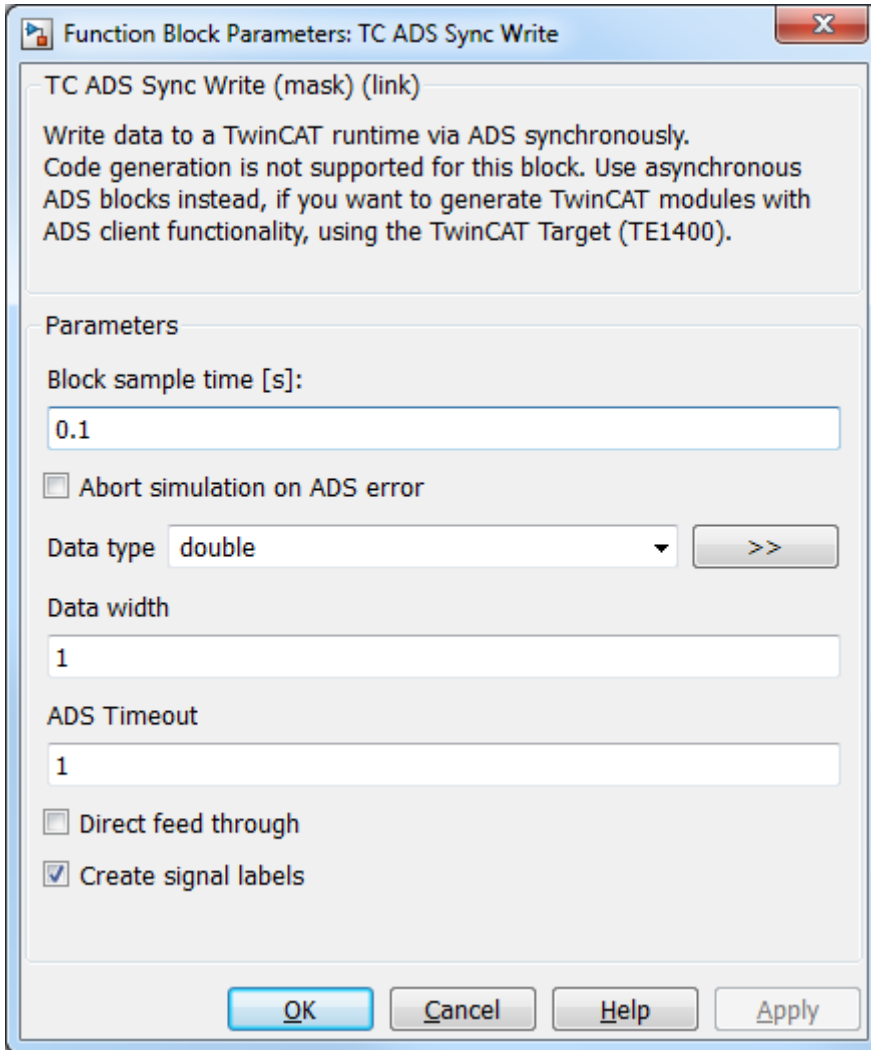
Block input ports

NetId	The target AMS NetId
Port	The target ADS Port
IdxGrp	The target ADS Index Group
IdxOffs	The target ADS Index Offset
Data	The value to be written to the ADS variable

Block output ports

Err	Boolean flag signaling an actual ADS error
ErrId	The ADS error code

Block parameters



Block sample time	The Simulink block sample time, which is the ADS request interval (referred to the Simulink time but not to the real time)
Abort simulation on ADS error	Stop simulation at the first ADS error
Data type	The data type of the <i>Data</i> input signal, which should be the data type of the transmitted ADS variable
Data width	The width of the <i>Data</i> input signal. Use values greater than 1 to transmit arrays.
ADS timeout	Simulink will wait on a response of the last ADS request until this time span is exceeded (in each time step)
Direct feed through	Evaluate the input values in the output method of the block. This ensures, that the values at the output ports in the current time step belong to the ADS parameters at the input ports of the same time step. Otherwise the output values belong to the input ADS parameters of the previous time step. Activating this option can significantly slow down the simulation.
Create signal labels	Signal labels are created for the signal lines, connected to the output ports

Code generation

Code generation is not supported. The reason is, that synchronous ADS communication requires to wait for the ADS responses. Due to nondeterministic ADS response times, this could lead to big jitter and cycle time exceedings in real time applications. Use asynchronous ADS blocks for use in real time applications instead.

2.4.3 Utilities

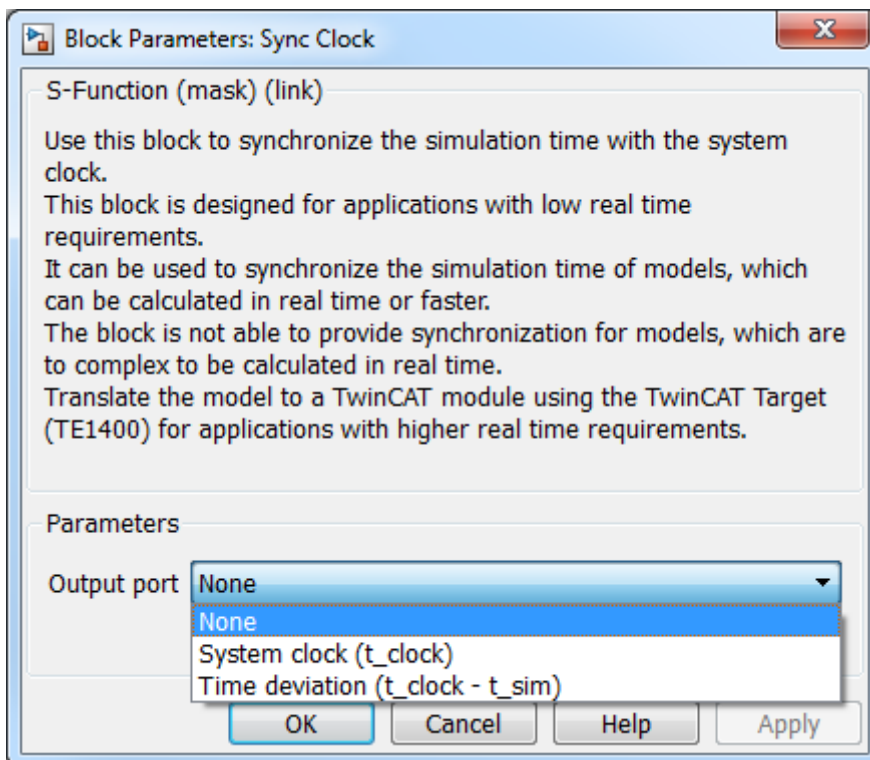
2.4.3.1 Sync Clock

The block "Sync Clock" can be used to slow down the simulation time to the system time. You can find this block beneath the subfolder "Utilities" of the Beckhoff ADS Library inside the "Simulink Library Browser".



Sync Clock

Block parameters



Output port

Activates an optional block output port, which can be used to monitor the deviation between the simulation and the system time.

Code generation

Code generation is supported for the TwinCAT Target but no code is generated for this block, because real time modules do not require any system time synchronization.

3 From version 2.x.xxxx.x

- TE1410 Interface for MATLAB®/Simulink® versions lower than 1.2.xxxx.x support MATLAB R2010b to MATLAB R2019a.
- TE1410 Interface for MATLAB®/Simulink® versions higher than 2.x.xxxx.x support MATLAB R2019a and higher.
- The installations for both versions can be used in parallel on one engineering system.
- Compatibility: the versions are not compatible with each other. When version 2.x is set up for a MATLAB® version, the version 1.2.x functions are removed from the MATLAB® path.

3.1 Overview

TE1410 TwinCAT Interface for MATLAB®/Simulink®

The TwinCAT 3 Interface for MATLAB®/Simulink® enables data exchange between MATLAB® and the TwinCAT Runtime and between Simulink® and the TwinCAT Runtime.

For Simulink®, ADS client blocks that can be simply integrated and configured in Simulink® projects are provided in the Simulink® library. It is thus possible, from the Simulink® environment, to write data to a TwinCAT runtime or to read data from it. An application example of the use of ADS blocks in Simulink® is the software-in-the-loop simulation, in which a regulating or control algorithm executed in the TwinCAT runtime is connected with a model executed in Simulink®, see description [ADS in Simulink® \[▶ 31\]](#).

An ADS Port object that can be used in the script environment is provided for MATLAB®. The object offers various methods for the exchange of data between MATLAB® and the TwinCAT runtime. In addition to an ADS client in MATLAB® – i.e. the MATLAB® environment initiates the data exchange – the use of an ADS server in MATLAB® is also possible. The latter offers the option of writing MATLAB® functions within the MATLAB® environment and calling them from TwinCAT via an ADS command. An application example for the use of the ADS client in MATLAB® is the creation of a graphical user and monitoring interface, e.g. based on a [MATLAB® app](#). The ADS server can be used, for example, to perform non-real-time relevant tasks, such as components of a predictive maintenance system or a parameter optimization. The [MATLAB® Compiler™](#) can be used to execute the programs created as stand-alone applications in the field, e.g. on an edge device or directly on the Industrial PC, see description [ADS in MATLAB® \[▶ 28\]](#).

Further Information

- <https://www.beckhoff.com/TE1410>
- <https://www.beckhoff.com/matlab>

3.2 Installation

System requirements

A distinction is made between the system running MATLAB®, Simulink® or the MATLAB® Compiler Runtime (hereinafter MATLAB® PC) and the system running the TwinCAT Runtime (runtime PC).

MATLAB PC

- Windows 10 or equivalent
- MATLAB R2019a or higher
- TwinCAT 3 ADS or TwinCAT 3 XAE Setup, at least version 3.1.4024.7

Runtime PC

- TwinCAT 3 XAR, at least version 3.0
- Operating system of the runtime PC is not relevant

Installation

The installation only needs to be run on the MATLAB® PC.

- If there is no TwinCAT on the system, install TwinCAT 3 ADS or TwinCAT 3 XAE.
- Start TwinCAT Tools for MATLAB® and Simulink® setup to install the TE1410.
- The TE1410 is installed within the TwinCAT folder structure, i.e. it is separate from the MATLAB® installation.
- If there is no MATLAB® on the system, install MATLAB® at the latest now.
- Start MATLAB® as administrator and run `%TwinCAT3Dir%..\Functions\TE14xx-ToolsForMatlabAndSimulink\SetupTE14xx.p` in MATLAB® to link the functions from TE1410 with this MATLAB® version.

3.3 Licenses

To use the functions of the TE1410 interface for MATLAB®/Simulink®, a **TE1410 license** is required on the **MATLAB® PC**.

From the MATLAB® PC, you can connect to an unlimited number of target platforms. No additional runtime licenses are required for using the TE1410 on the target platforms.

If you do not have a TE1410 license, you can use the software as a demo version.

● No 7 days trial license

i A 7-day trial license with full functionality is not available for this product.

Restrictions in the demo version

When using the demo version, appropriate warnings are displayed to the user.

For [ADS in Simulink® \[► 31\]](#): after 30 minutes, the simulation ends with an error message. The simulation must be restarted. In addition, the number of ADS symbols for reading or writing is limited to 4.

For [ADS in MATLAB® \[► 28\]](#): after 30 minutes of runtime, the created port object goes into an error state. MATLAB® must be restarted to resume 30 minutes of testing. There is no limitation regarding the ADS symbols in this case.

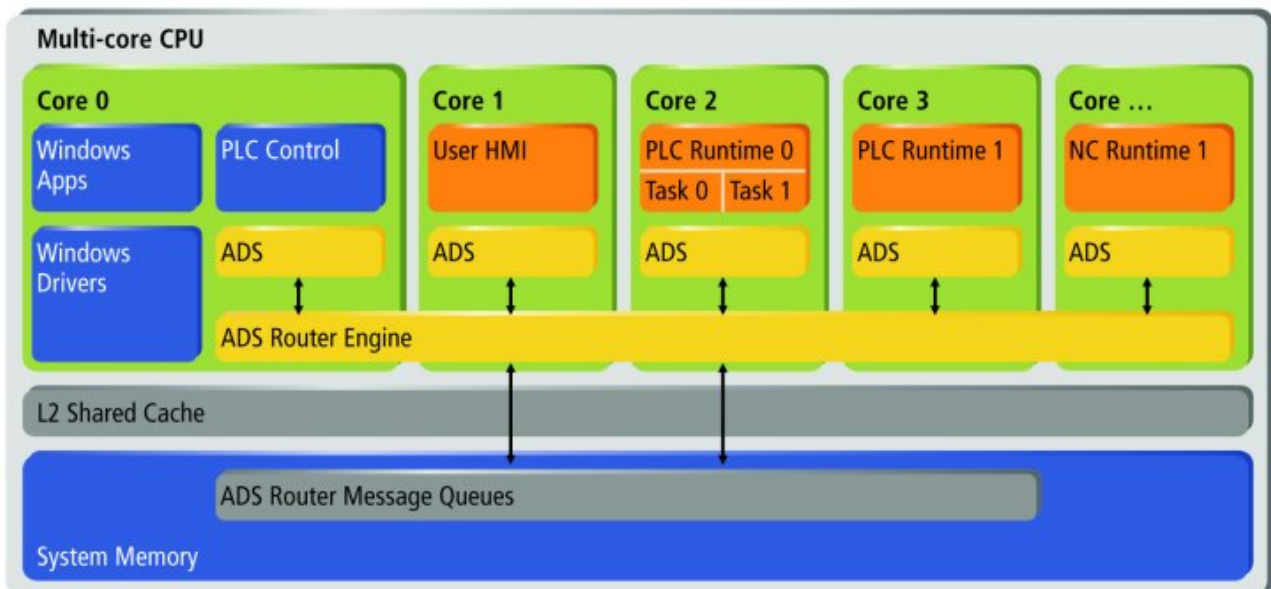
3.4 Short introduction: TwinCAT ADS

Basic structure of ADS devices and ADS symbols

Automation Device Specification (ADS) is the basis for the Interface for MATLAB®/Simulink®. ADS describes a device- and fieldbus-independent interface and enables communication between ADS devices.

The ADS device concept and the identification of an ADS device are explained below.

The modular system architecture of TwinCAT allows the individual parts of the software (e.g. TwinCAT PLC, TwinCAT NC ...) to be regarded as independent devices: there is a software module for each individual task. The servers in the system are the executing devices that provide certain services. The clients are programs that request the services of the servers. A client initially establishes a connection to the server and requests a service: for example, it requests reading the value of a variable or it requests writing a variable.



The Interface for MATLAB®/Simulink® provides an ADS client interface for Simulink® and MATLAB®, which enables data exchange (read and write) with TwinCAT runtimes. The TwinCAT runtime, or its ADS devices, thus provide their services as ADS servers and can be used from LabVIEW™.

In addition, the interface for MATLAB®/Simulink® provides an ADS server for MATLAB® so that a MATLAB® environment can provide services to ADS clients.

ADS data exchange between ADS devices takes place via the ADS router. As shown in the diagram above, data exchange between ADS devices implemented on the same system takes place via the system memory. If two ADS devices, e.g. MATLAB® and the TwinCAT runtime are on different systems, a route can be created between two ADS routers. When creating the ADS route, the transport type (usually TCP/IP) for communication between the two ADS routers can be defined. Accordingly, an ADS device identifies itself via the AMS NetId of the ADS router and a port number, which then specifies the ADS device on the system. For example, port 851 is the default port for the first PLC instance in the TwinCAT runtime. ADS services of an ADS device are then specified by two parameters, the index group and the index offset. For example, a PLC variable is accessible for reading or writing under a specific index group and index offset.

Summary

- AMS NetId: identifies the ADS router, i.e. the system.
- ADS Route: specifies the connection between two ADS routers.
- Port: identifies an ADS device.
- Index Group/Offset: specifies the ADS system service, e.g. a variable for reading and writing.

In order to make the addressing of variables in a TwinCAT runtime more convenient for the user, TwinCAT creates ADS symbols, which can be searched with the Target Browser, for example. The Target Browser is also integrated in the interface for MATLAB®/Simulink® in the TC Symbol Interface, so that ADS symbols can be selected easily and quickly. An ADS symbol for a variable in TwinCAT then contains the information mentioned above: AMS NetId, Port, Index Group and Index Offset and furthermore the Bit Size as well as a symbol name and the data type of the variable.

For more information on ADS, please see the following links:

- AMS NetId and Port: [ADS device identification](#)
- Index Group and Index Offset: [Specification for ADS devices](#)
- ADS Routes: [System Node "Routes"](#) and [Add Route](#)
- Connecting Devices with same AMS NetId: [AmsNAT](#)
- Using MQTT and Message Broker with ADS: [ADS-over-MQTT](#)
- TLS encrypted ADS: [Secure ADS](#)
- Target Browser: [Target Browser](#)

- Record ADS communication: [ADS Monitor](#)

Basic ADS data communication

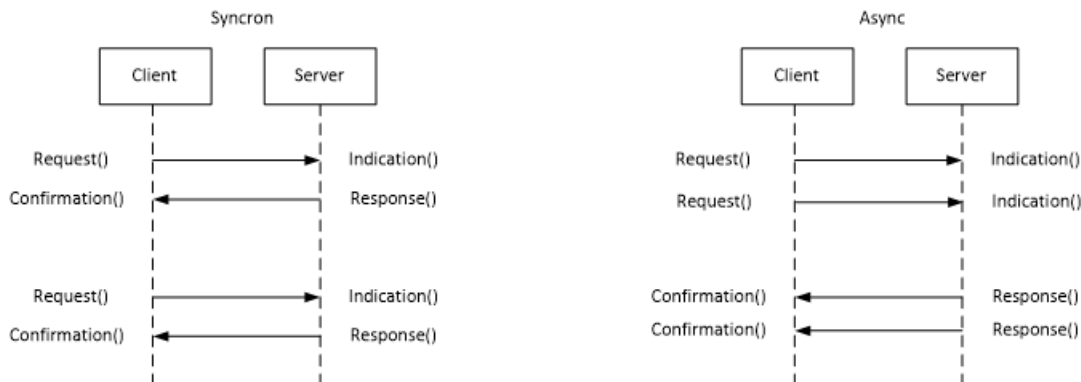
Basically, ADS has three different ways/modes of communicating data:

Synchronous reading or writing

- The ADS client waits for response from the ADS server before proceeding with the execution of the code.

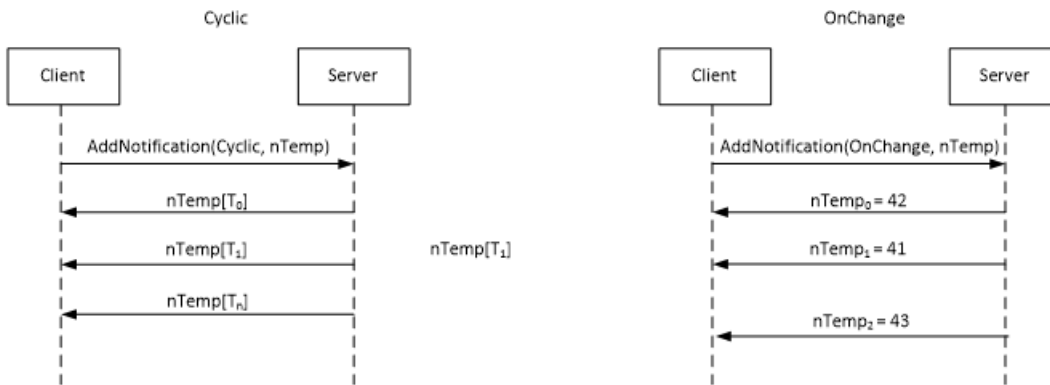
Asynchronous reading or writing

- The ADS client sends a read or write request to the ADS server, but does not wait for a response; instead, it continues to execute other parts of the program (e.g., requests to other variables).



Event-based communication (read only)

- An ADS notification is only registered once on the server. The notification can be registered "on change" or "cyclic". After a notification has been registered at the server, it is sent to the server without any further requests from the client.

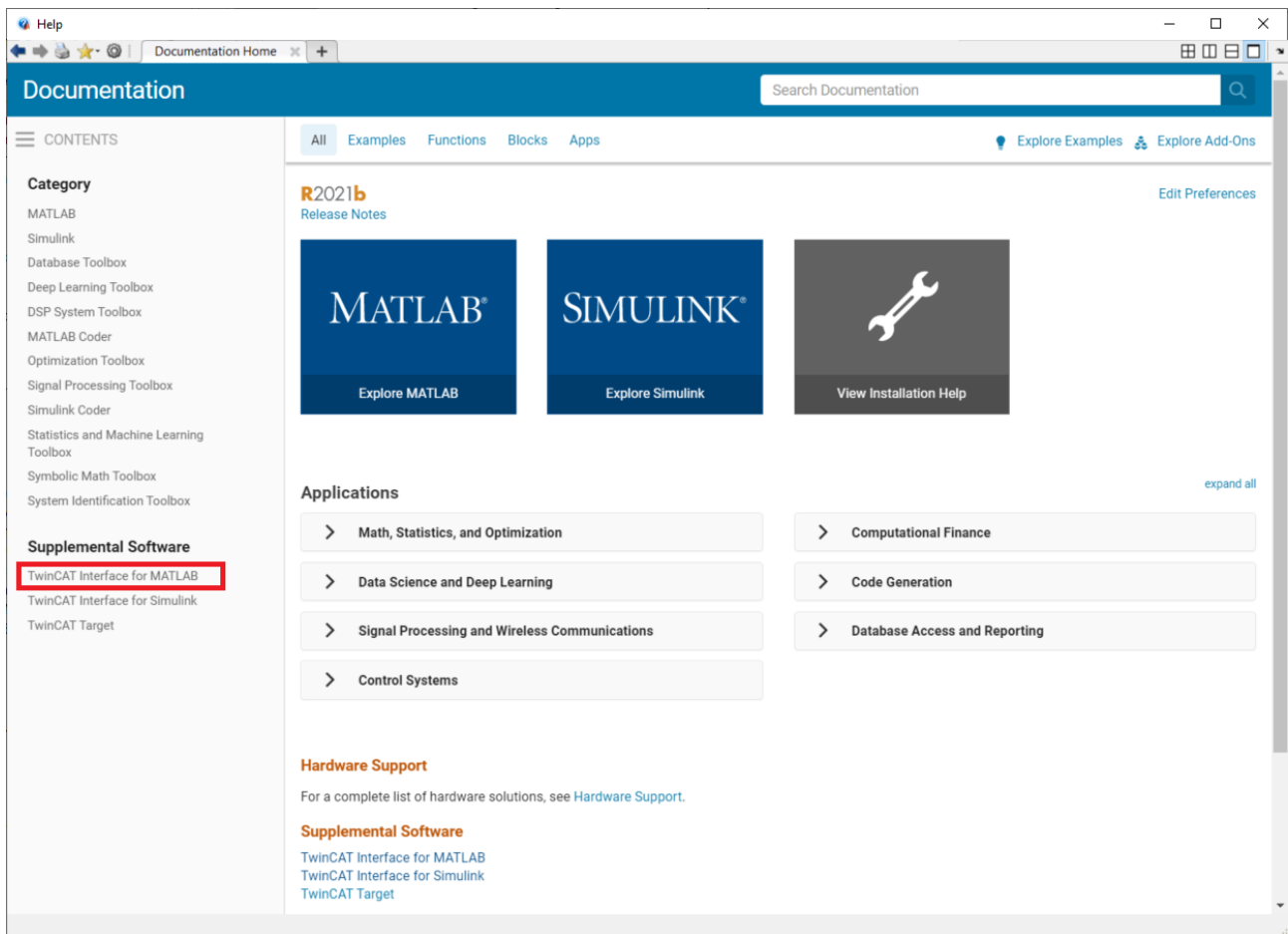


3.5 ADS in MATLAB®

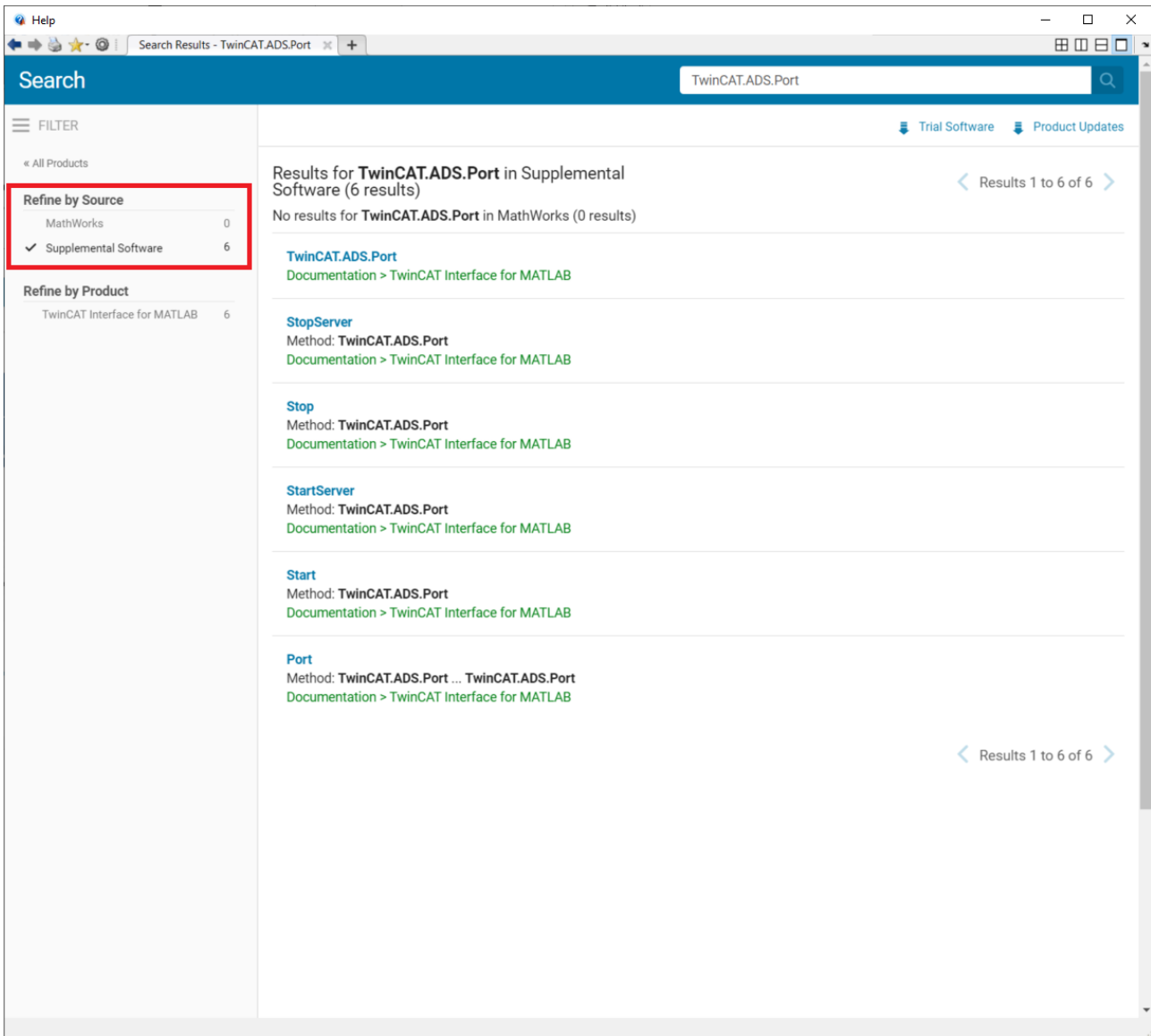
3.5.1 API Documentation

Documentation in MATLAB® Help

- ✓ The API is fully documented in the MATLAB® help.
- 1. Go to MATLAB® help and select **TwinCAT Interface for MATLAB** from **Supplemental Software**.



⇒ The TE1410 documentation can also be accessed via the search function of MATLAB®.



Intellisense in MATLAB®

During programming in MATLAB®, you are shown help to capture the transfer parameters directly and not have to search in the help every time.

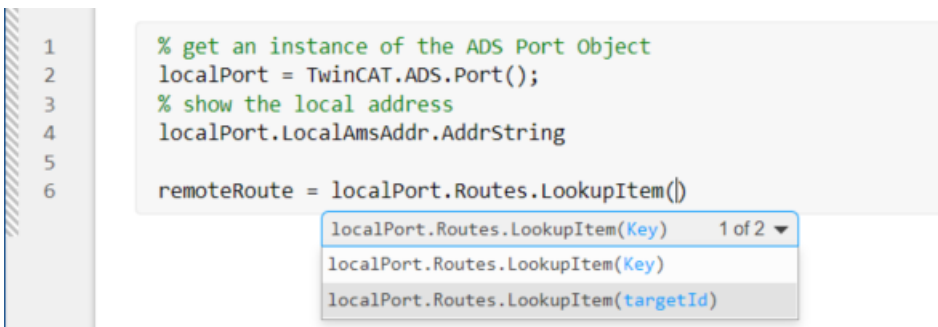


Fig. 1:

```
1 % get an instance of the ADS Port Object
2 localPort = TwinCAT.ADS.Port();
3 % show the local address
4 localPort.LocalAmsAddr.AddrString
5
6
7
8
9 localPort.Write(
    localPort.Write(vAdsVarAddress,vWriteData,timeout_ms)
    e.g. {'Local' 851 'MAIN.counter'}, 'Local:851/MAIN.counter', {'Loc...
    e.g. {'Local' 851 'MAIN.counter'}, 'Local:851/MAIN.counter', {'Local' 851 hex2dec('80000000') 456}
```

This functionality is available for all supported MATLAB® releases when using MATLAB® Live Scripts. When using MATLAB® scripts, at least a MATLAB version R2021b is required.

3.5.2 Basic concepts

The basic concepts of using the ADS port object in MATLAB® are addressed in various [samples](#) [▶ 41].

For an introduction to using an ADS client in MATLAB®, see the following sample:

```
TwinCAT.ADS.Samples.Start('ADS Client Basics')
```

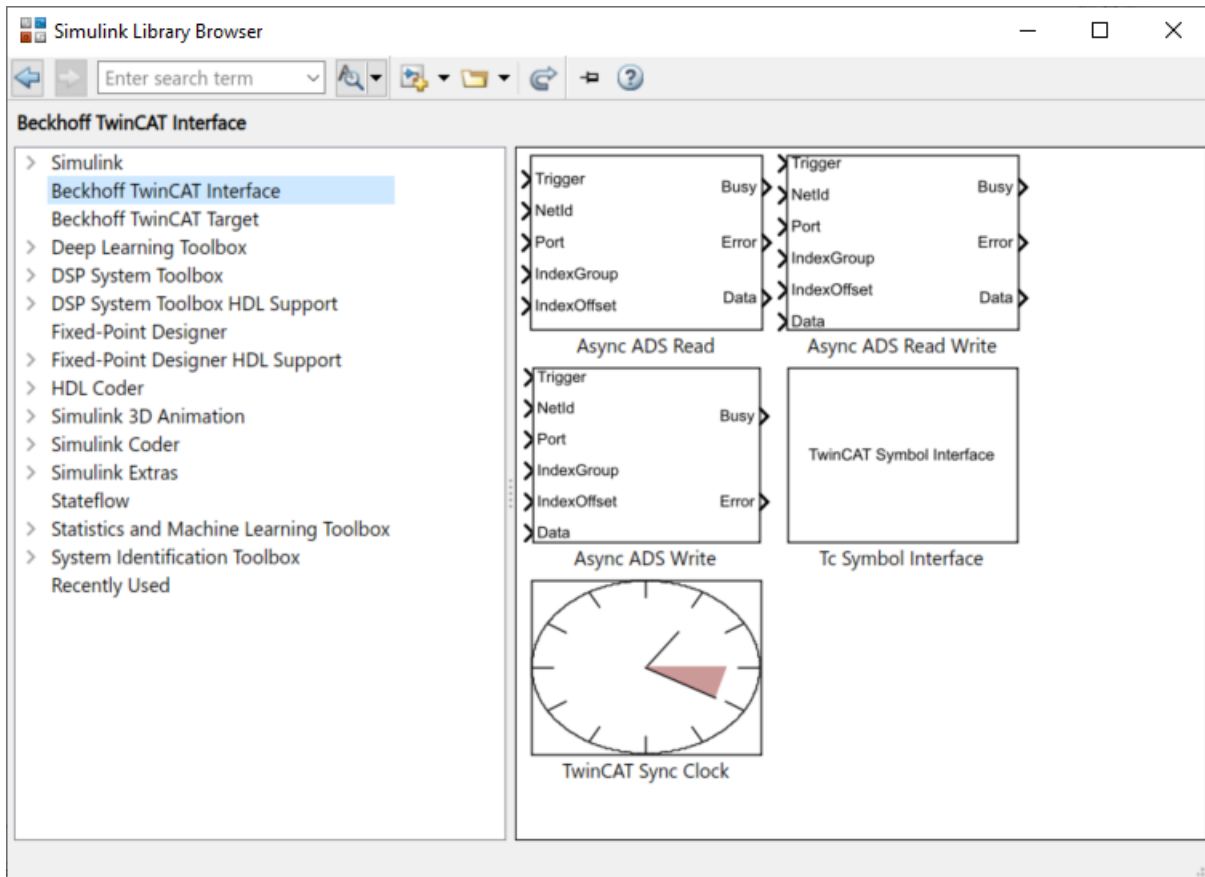
For an introduction to using an ADS server in MATLAB®, see the following sample:

```
TwinCAT.ADS.Samples.Start('ADS Server Basics')
```

3.6 ADS in Simulink®

3.6.1 ADS blocks

The ADS blocks in Simulink® can be inserted as usual via the Library Browser.



i Synchronous ADS communication between Simulink® and TwinCAT is also possible. See the following sample: `TwinCAT.ADS.Samples.Start('ADS Client Simulink Simulation')`

3.6.1.1 Tc Symbol Interface

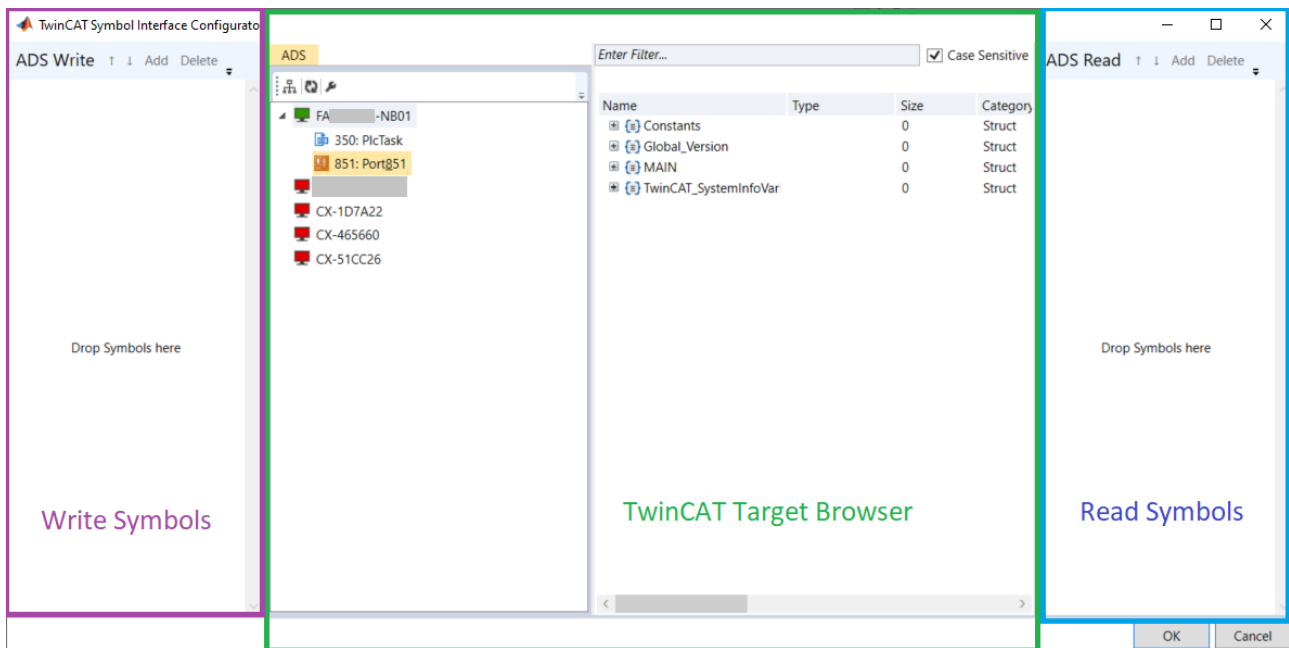
Tc Symbol Interface is the central block for communication between Simulink® and TwinCAT via ADS. The configuration of the ADS symbols to be read and written is done via the integrated Target Browser.



Tc Symbol Interface

TwinCAT Symbol Interface Configurator

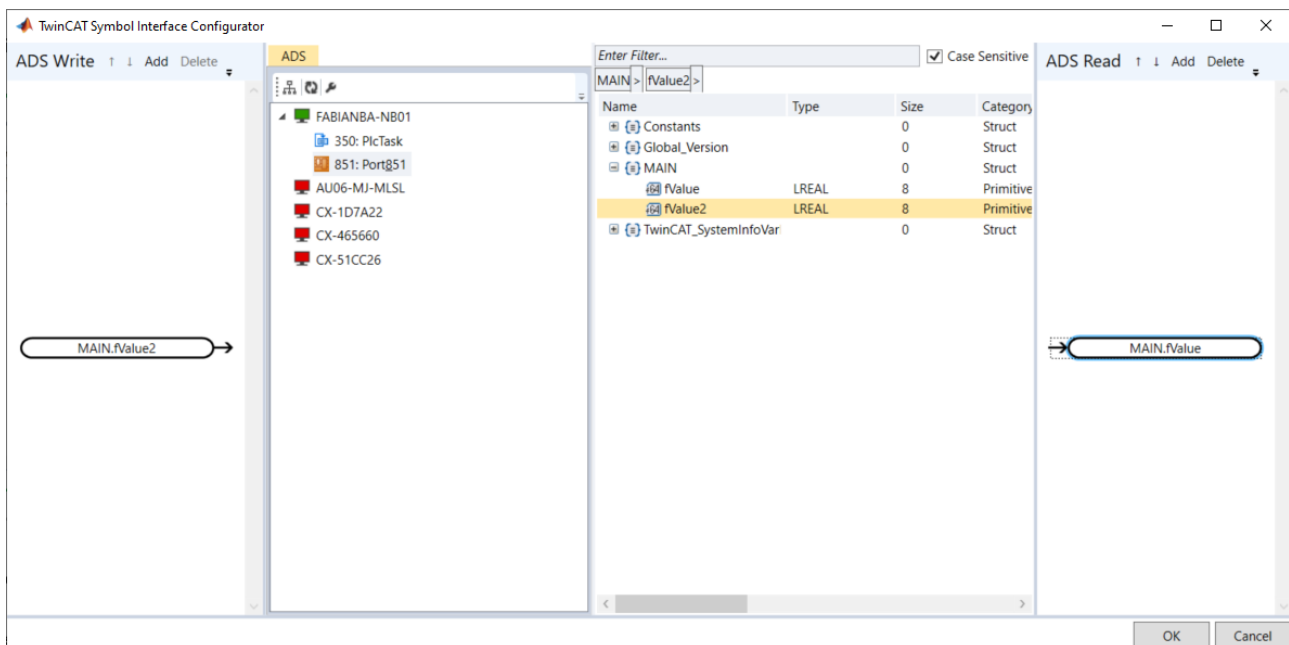
Without configuration, no input or output ports exist on the block. These are first configured by the user via the TwinCAT Symbol Interface Configurator. The configurator opens by double-clicking on the block.



In the middle part the TwinCAT Target Browser is shown. For its use, see [Scope View \(Extension ADS\)](#) documentation.

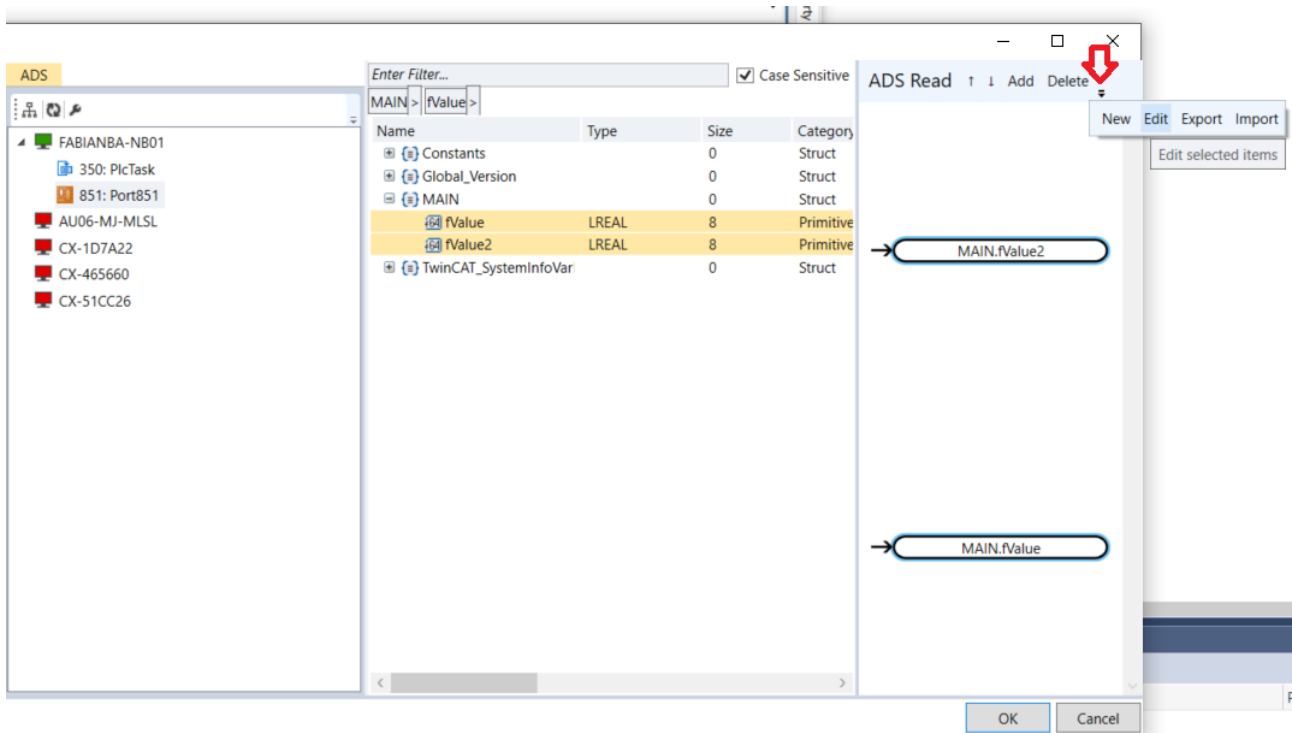
With the TwinCAT Target Browser you can browse into the target systems connected with routes and search the ADS symbols.

To select symbols for reading or for writing (multi-select is possible), you just have to drag-and-drop the symbols to the right for reading (blue box) and to the left for writing (purple box).



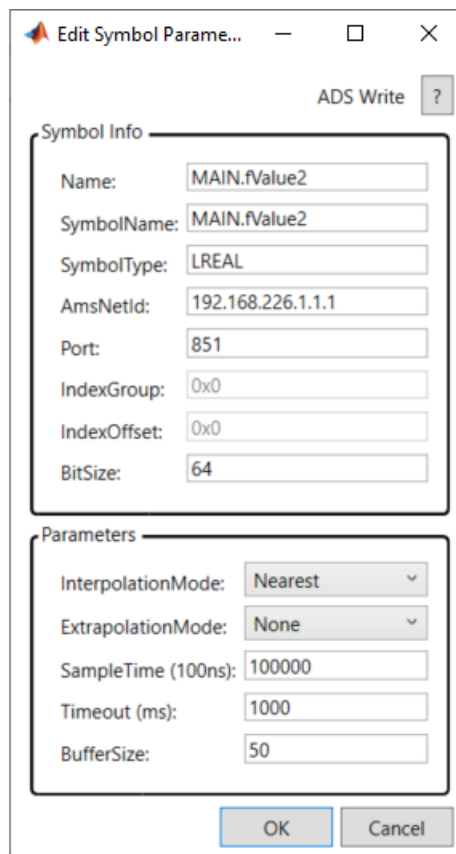
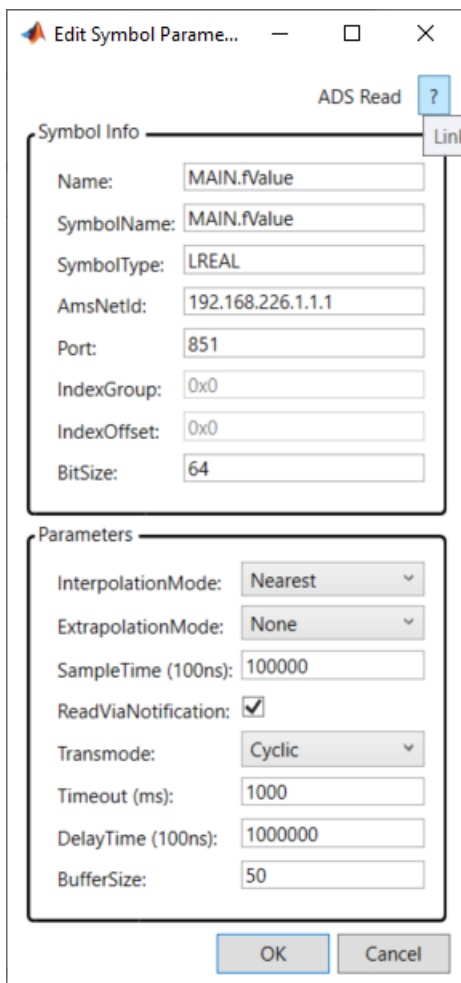
A node will then appear in the left and right areas for each ADS symbol. By double-clicking on a node, you can specify the read or write properties of the node.

You can also select multiple nodes (Ctrl+Click) and use the button **Edit** to edit all selected nodes simultaneously with regard to the read or write properties.



Likewise, you can export the configuration here and import it again.

Double-clicking on a node or selecting the button **Edit** opens a new window.



The symbol information is entered in the upper area. These are not relevant for the type of data transport. With Multi-Select, this area is grayed out because it can only be assigned to one symbol and not several at the same time. In the lower area you can specify the property of reading and writing.

Symbols Reading

Parameter	Default value	Description
Sample Time (100 ns)	TwinCAT task cycle time	Period value with which the ADS symbol is to be read.
Read Via Notification	Yes	Yes: ADS Notification is logged in. No: read by polling with SampleTime
Transmode	Cyclic	Only for notifications. Cyclic: value is transferred with SampleTime to Simulink®. OnChange: value is transferred to Simulink® when the value changes.
Timeout (ms)	1000	ADS time-out period
DelayTime (100 ns)	10*SampleTime	Buffer size on TwinCAT XAR side. Default are 10 values of the ADS symbol.
BufferSize	50	Buffer size in read samples on Simulink® page.
Interpolation mode	Nearest	None: no interpolation Previous: uses the previous value. Next: uses the next known value. Nearest: uses the previous or next value, whichever is closer [*].
Extrapolation mode	None	None: no extrapolation Constant: new value is set to the last known value.

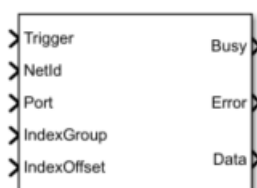
[*] If the Simulink® SampleTime does not match the TwinCAT task cycle time of an ADS symbol, values for a specific time stamp may not be available. Further it can happen that during simulation with several SampleTimes a value with time stamp between existing values of Simulink® is expected. Accordingly, if there is enough data in the buffer (see BufferSize), you can interpolate or not interpolate between two values.

Symbols Writing

Parameter	Default value	Description
Sample Time (100 ns)	TwinCAT task cycle time	Period value with which the ADS symbol is to be written.
Timeout (ms)	1000	ADS time-out period
BufferSize	50	Buffer size in samples on Simulink® page.
Interpolation mode	Nearest	see above
Extrapolation mode	None	see above

3.6.1.2 Async ADS Read

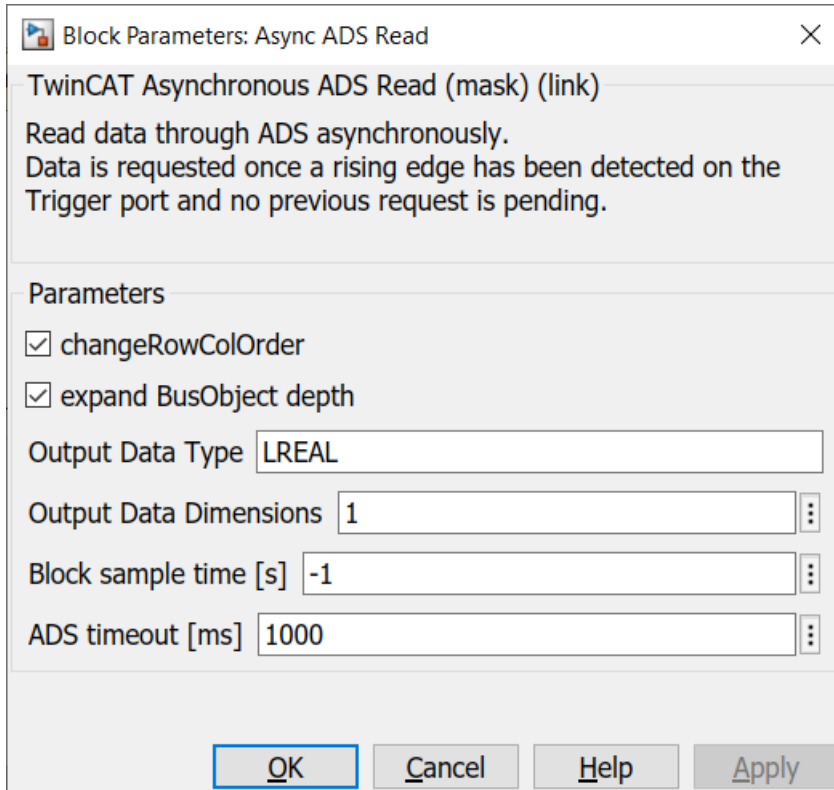
Asynchronous reading of an ADS symbol described at the input via AMS Net Id, Port, Index Group and Index Offset [► 26]. The read command is sent from Simulink® with a positive edge at the trigger port.



Port description

Port Name	Data type	Description
Trigger	boolean	Block sends read command on positive edge.
NetId	uint8	AMS Net Id of the target system
Port	uint16	Port of the ADS device
IndexGroup	uint32	Index group
IndexOffset	uint32	Index offset
Busy	boolean	TRUE from sending the request until receiving the response from the server
Error	int32	Errorcode. Important: <i>Pending</i> is displayed as an error code.
Data	<i>configurable</i>	See Output Datatype and Dimension in Block Parameter

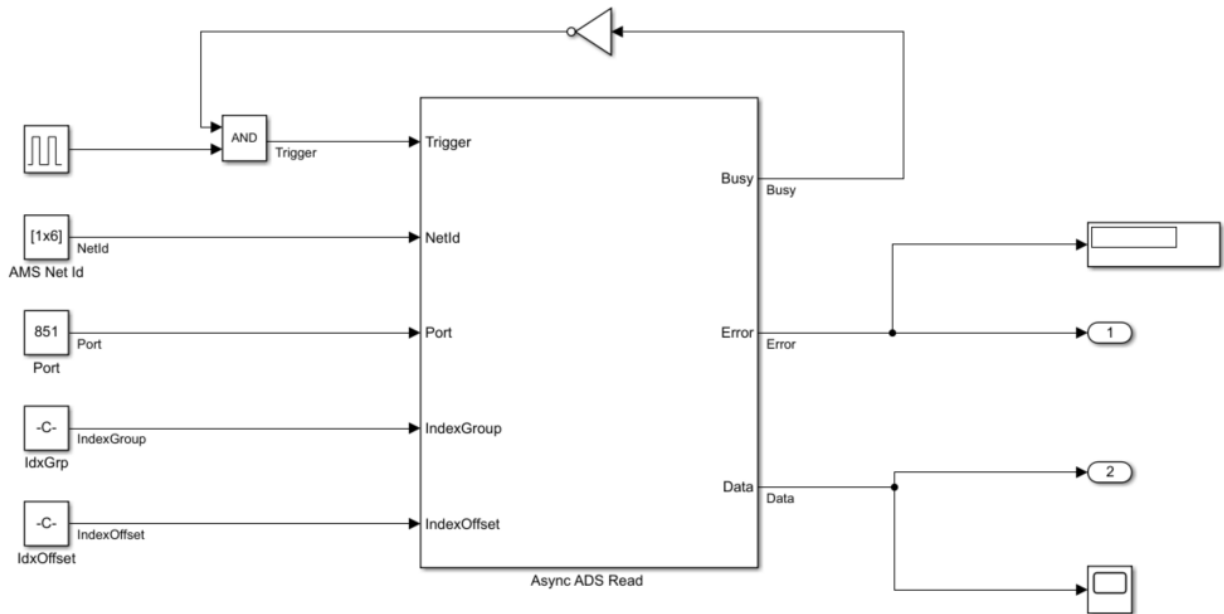
Block Parameter



Parameter	Description
"Change Row Col Order" or "Change row major to column major matrix element order"	Transposes the data to the ordinary representation in Simulink®.
Output Data Type	TwinCAT primary data type, e.g. UINT, LINT, REAL, ...
Output Data Dimensions	Dimension, e.g. 5, [2.3], ...
Block sample time [s]	Sample Time of the block (Default -1 = Simulink Sample Time)
ADS timeout [ms]	Timeout for ADS call

Sample

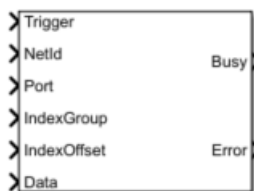
An example model can be built as follows:



- The Const blocks carry the ADS address information here.
- The AMS Net Id is passed as an array, e.g. [192,168,226,1,1,1].
- For Index Group and Index Offset please note that usually in TwinCAT (for example in the Target Browser) the information is given in hexadecimal. It is a good idea to use the hex2dec command, e.g. hex2dec('5DFF0').
- Busy output feedback is used to prevent a next request from being made during one request.

3.6.1.3 Async ADS Write

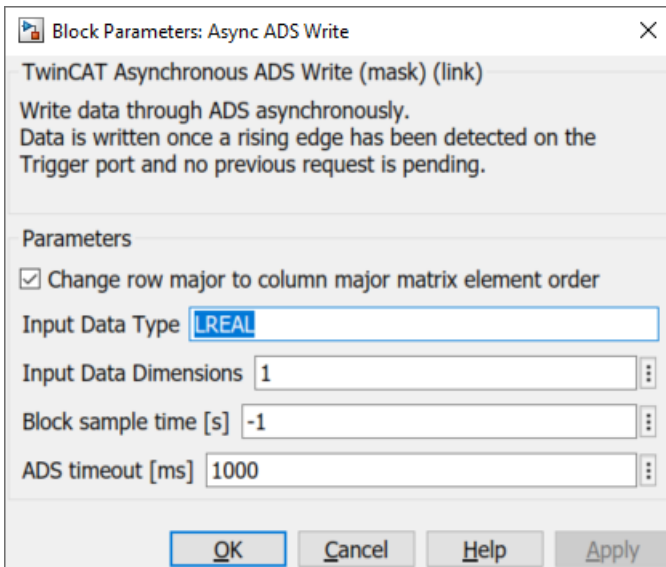
Asynchronous writing of an ADS symbol described at the input via AMS Net Id, Port, Index Group and Index Offset [▶ 26]. The read command is sent from Simulink® with a positive edge at the trigger port.



Port description

Port Name	Data type	Description
Trigger	boolean	Block sends read command on positive edge.
NetId	uint8	AMS Net Id of the target system
Port	uint16	Port of the ADS device
IndexGroup	uint32	Index group
IndexOffset	uint32	Index offset
Data	<i>configurable</i>	See Input Datatype and Dimension in Block Parameter
Busy	boolean	TRUE from sending the request until receiving the response from the server
Error	int32	Error code

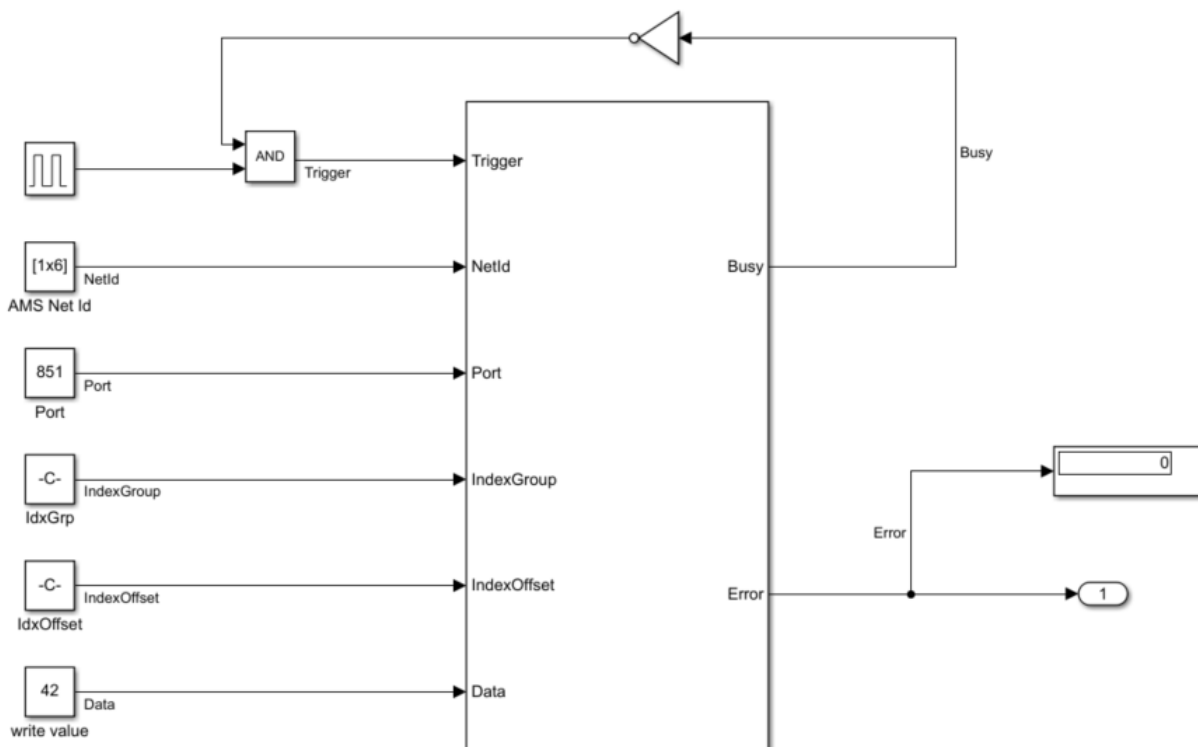
Block Parameter



Parameter	Description
Input Data Type	TwinCAT primary data type, e.g. UINT, LINT, REAL, ...
Input Data Dimensions	Dimension, e.g. 5, [2.3], ...
Block sample time [s]	Sample Time of the block (Default -1 = Simulink Sample Time)
ADS timeout [ms]	Timeout for ADS call

Example

An example model can be built as follows:

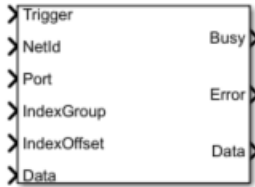


- The Const blocks carry here the ADS address information and the value 42 to transfer it to TwinCAT.
- The AMS Net Id is passed as an array, e.g. [192,168,226,1,1,1].
- For Index Group and Index Offset please note that usually in TwinCAT (for example in the Target Browser) the information is given in hexadecimal. It is a good idea to use the hex2dec command, e.g. hex2dec('5DFF0').

- Busy output feedback is used to prevent a next request from being made during one request.

3.6.1.4 Async ADS ReadWrite

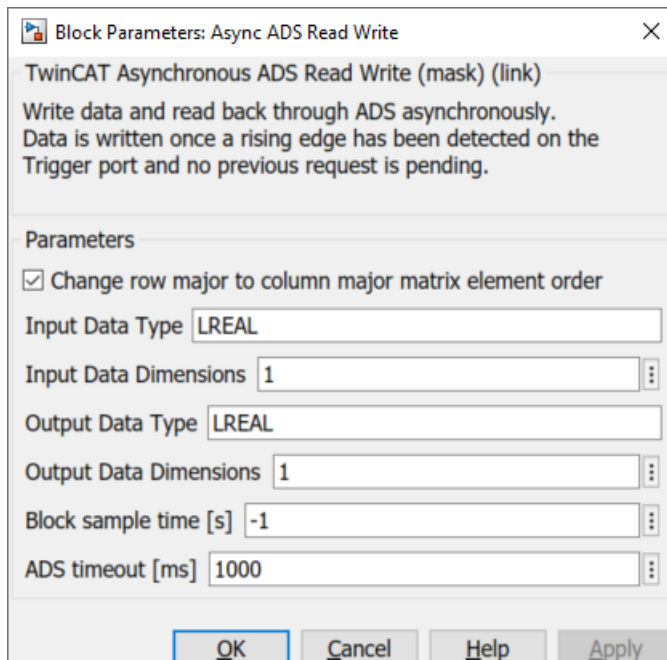
Asynchronous ReadWrite of an ADS symbol written at the input via AMS Net Id, Port, Index Group and Index Offset [▶ 26]. The read command is sent from Simulink® with a positive edge at the trigger port.



Port description

Port Name	Data type	Description
Trigger	boolean	Block sends read command on positive edge.
NetId	uint8	AMS Net Id of the target system
Port	uint16	Port of the ADS device
IndexGroup	uint32	Index group
IndexOffset	uint32	Index offset
Data	<i>configurable</i>	See Input Datatype and Dimension in Block Parameter
Busy	boolean	TRUE from sending the request until receiving the response from the server
Error	int32	Error code
Data	<i>configurable</i>	See Output Datatype and Dimension in Block Parameter

Block Parameter

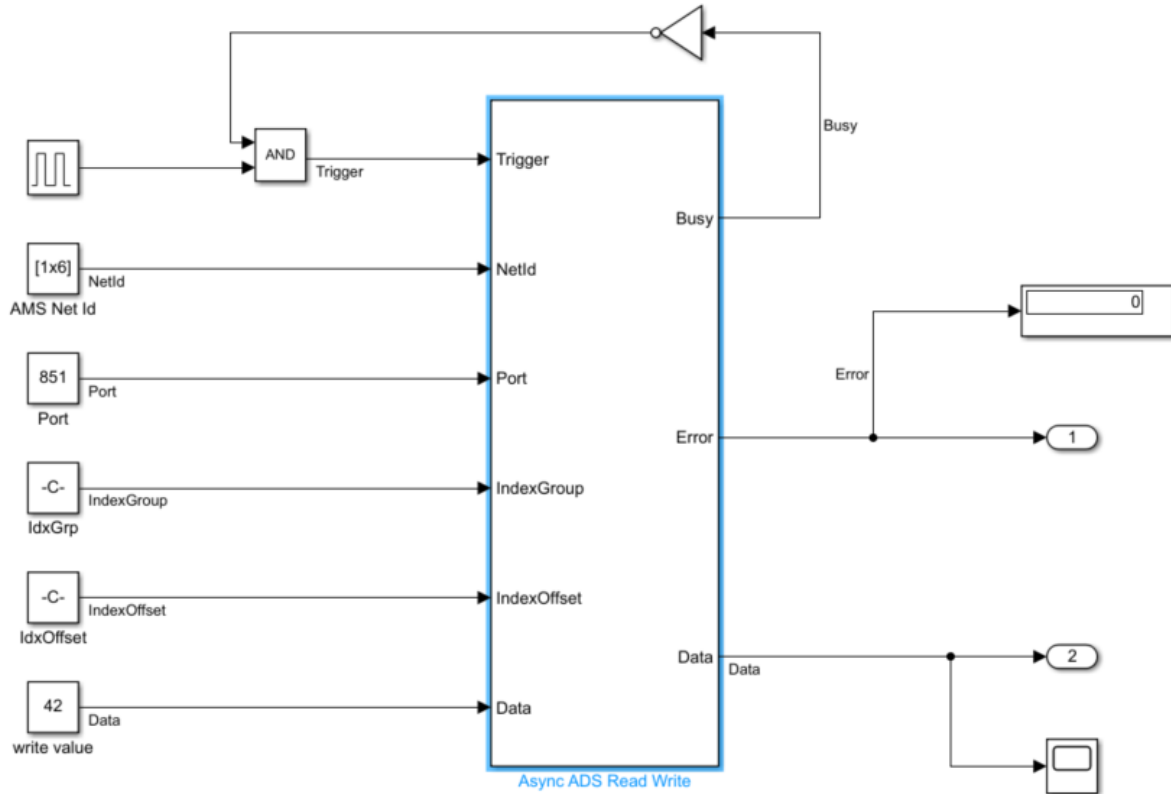


Parameter	Description
Input Data Type	TwinCAT primary data type, e.g. UINT, LINT, REAL, ...
Input Data Dimensions	Dimension, e.g. 5, [2.3], ...
Output Data Type	TwinCAT primary data type, e.g. UINT, LINT, REAL, ...
Output Data Dimensions	Dimension, e.g. 5, [2.3], ...

Parameter	Description
Block sample time [s]	Sample Time of the block (Default -1 = Simulink Sample Time)
ADS timeout [ms]	Timeout for ADS call

Sample

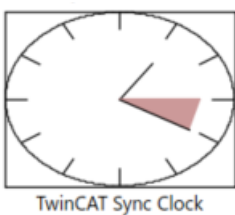
An example model can be built as follows:



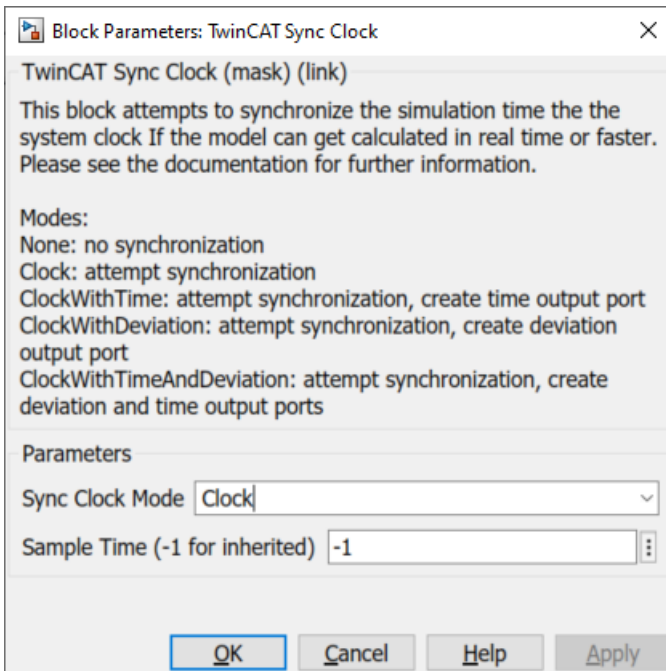
- The Const blocks carry here the ADS address information and the value 42 to transfer it to TwinCAT as input of the function.
- The AMS Net Id is passed as an array, e.g. [192,168,226,1,1,1].
- For Index Group and Index Offset please note that usually in TwinCAT (for example in the Target Browser) the information is given in hexadecimal. It is a good idea to use the hex2dec command, e.g. hex2dec('5DFF0').
- Busy output feedback is used to prevent a next request from being made during one request.
- Data at the output then carries the return value of the called function.

3.6.1.5 TwinCAT Sync Clock

Synchronization of Simulink® simulation with local system time.



Block Parameter



Sync Clock Mode	
None	Simulation is neither accelerated nor decelerated.
Clock	Simulation time and system time are synchronized.
ClockWithTime	Simulation time and system time are synchronized. In addition, an output port is created that displays the past system time.
ClockWithTimeAndDeviation	Simulation time and system time are synchronized. In addition, an output port is created that shows the past system time and the deviation between simulation time and system time.

Comment

The block does not synchronize the Simulink® simulation with the TwinCAT runtime, but with the local system time. Synchronization of the local system time is done by "braking" the Simulink® simulation; acceleration is not possible.

For a fully synchronized software-in-the-loop simulation between TwinCAT and Simulink® you need a TwinCAT 3 Usermode Runtime: External Control and the SiL Connector Block in Simulink®.

3.7 Samples

The MATLAB® Command Window provides you with a list of provided samples. To do this, use the command:

```
TwinCAT.ADS.Samples.List
```

```

Command Window
>> TwinCAT.ADS.Samples.List

TE1410 Samples:

ADS Client Basics:
  Products: TE1410 - MATLAB ADS classes
  Topics: Expore ADS routes, Read PLC variables into MATLAB workspace, Plot a series of values read from a PLC variable
  Level: 1
  Description: Shows basic ADS client functionality by connecting to a TwinCAT PLC.
  Start

ADS Sum Requests:
  Products: TE1410 - MATLAB ADS classes
  Topics: Read multiple values via ADS sum request
  Level: 2
  Description: Shows how to access multiple TwinCAT PLC variables with a single ADS sum request.
  Start

ADS Request Objects:
  Products: TE1410 - MATLAB ADS classes
  Topics: Asynchronous ADS requests using request objects
  Level: 2
  Description: Shows how to use ADS request objects for parallelization of request execution.
  Start

ADS Server Basics:
  Products: TE1410 - MATLAB ADS classes
  Topics: MATLAB ADS Server
  Level: 3
  Description: Shows a MATLAB based implementation of a simple ADS server.
  Start

BigData with ADS Notifications through MATLAB ADS client:
  Products: TE1410 - MATLAB ADS classes
  Topics: Read PLC variables through an ADS notification
  Level: 5
  Description: Shows how to transfer large volumes of data from a TwinCAT PLC to MATLAB using ADS notifications
  Start

BigData through MATLAB ADS server:
  Products: TE1410 - MATLAB ADS classes
  Topics: Transfer data from the PLC to MATLAB
  Level: 5
  Description: Shows how to transfer large volumes of data from a TwinCAT PLC to MATLAB using the ADS server functionality
  Start

ADS Client Simulink Simulation:
  Products: TE1410 - MATLAB ADS classes
  Topics: Read through ADS, Write through ADS, Simulation, Simulink
  Level: 5
  Description: Create synchronous Simulink ADS blocks through MATLAB ADS
  Start

fx >> |

```

You can access the samples by clicking on the blue Start link. To do this, the sample code is copied to your user directory so that you do not change the original sample. You can work with the copy of the sample accordingly and try it out.

Also available for displaying and starting individual samples:

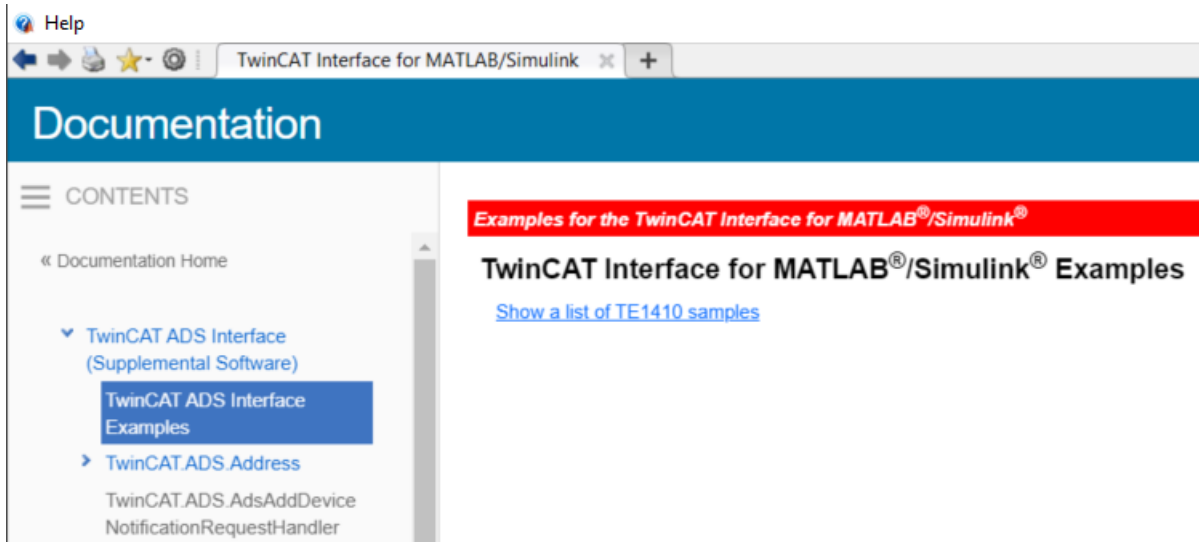
```
TwinCAT.ADS.Samples.Show(SampleName)
```

```
TwinCAT.ADS.Samples.Start(SampleName)
```

The argument *SampleName* is to be passed as a string, e.g.:

```
TwinCAT.ADS.Samples.Start('ADS Server Basics')
```

The list of samples can also be obtained from the documentation built into MATLAB® by clicking on the link **Show a list of TE1410 samples**.



More Information:
www.beckhoff.com/te1410

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

