EtherCAT® Set-Up a Network Configuration

How-To

Created by: EtherCAT Technology Group
Contact: info@ethercat.org

Date: 2016-07-05

EtherCAT® Technology Group

LEGAL NOTICE

Trademarks and Patents

EtherCAT® and Safety over EtherCAT® are registered trademarks and patented technologies, licensed by Beckhoff Automation GmbH, Germany. 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.

Disclaimer

The documentation has been prepared with care. The technology described is, however, constantly under development. For that reason the documentation is not in every case checked for consistency with performance data, standards or other characteristics. In the event that it contains technical or editorial errors, we retain the right to make alterations at any time and without warning. 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.

Copyright

CONTENTS

FIGURES

1 Scope of Configuration Tool

As example of the functionalities provided by a Configuration Tool, the TwinCAT 3.1 development environment (also called XAE) will be used in this document.

In TwinCAT the EtherCAT master device and Configuration Tool functionalities are integrated in the same software suite: an internal data format is therefore used to transfer the configuration from the TwinCAT development environment to the master device.

The TwinCAT development environment can additionally be used to generate a standard, XML-based ENI file according to the ETG.2100 specification, which can be provided to third-party EtherCAT master devices which do not have an own Configuration Tool (see How-To: Generate and Export ENI File).

The installation of TwinCAT 3.1 should take care of the following points:

a. TwinCAT 3.1 can be installed on both 32-bit and 64-bit Windows Operating Systems.

b. Although the TwinCAT 3.1 XAE is integrated in the Microsoft Visual Studio development environment, a previous installation of the latter software is not necessary: in case no Visual Studio installation is available on a PC, the TwinCAT 3.1 setup will install the Visual Studio shell.

c. TwinCAT 3.1 “Engineering” installation package has to be chosen, and the in installation should be performed by selecting “Run as administrator” (Figure 1).

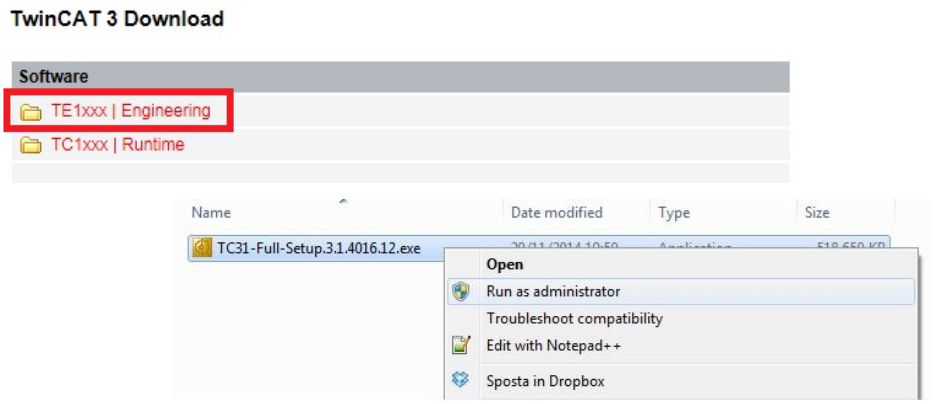

Figure 1: installation of TwinCAT 3.1

d. The TwinCAT Real-Time Ethernet Driver has to be installed on the network card which will be used as EtherCAT master (Figure 2).

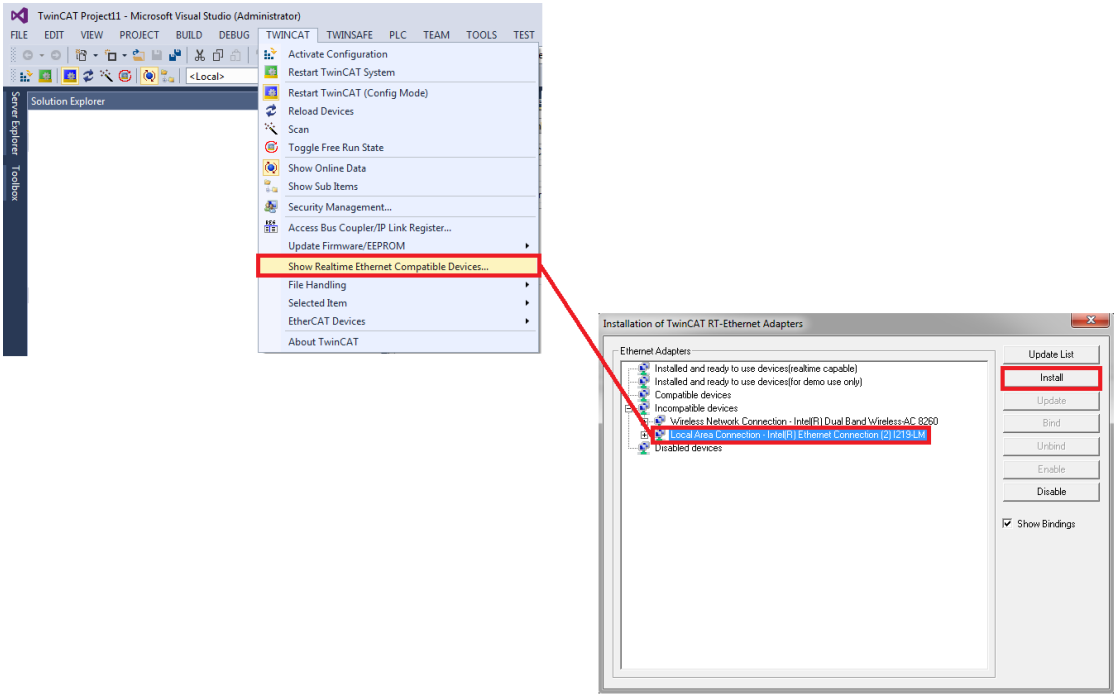

Figure 2: installation of the Real-Time Ethernet Driver

Note: If a network card is listed under “Incompatible devices”, this does not mean that it cannot be used to test the EtherCAT communication: it only means that this card will provide only weak real-time capabilities. For most of the testing purposes this is sufficient, therefore the driver can be installed. If the installation was successfully completed, the network card will be moved under the “Installed and ready to use devices” list:

- Compatible devices → Installed and ready to use devices (realtime capable)
- Incompatible devices → Installed and ready to use devices (for demo use only)

e. New TwinCAT projects can be created based on the custom template (Figure 3).

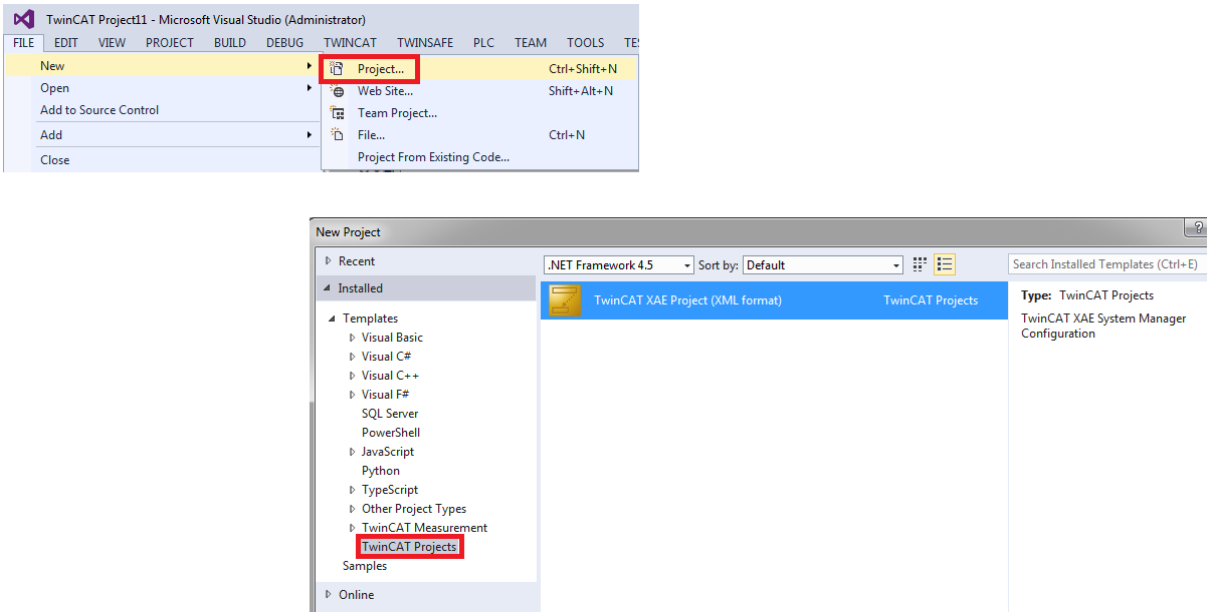

Figure 3: create a new TwinCAT project

2 Creating an EtherCAT Configuration

An EtherCAT network configuration can be created both offline (i.e. by inserting devices manually) and online (i.e. by scanning the connected devices).

2.1 Manual configuration

a. Click with the right mouse button on I/O → Devices, select “Add New Item…” and choose the “EtherCAT Master” device (Figure 4). In case the TwinCAT Real-Time Ethernet Driver is installed on more than one network card, a “Device Found At” pop-up will allow to select the adapter which will be used as EtherCAT master; otherwise, the only network card on which the driver is installed will be automatically configured as master device.

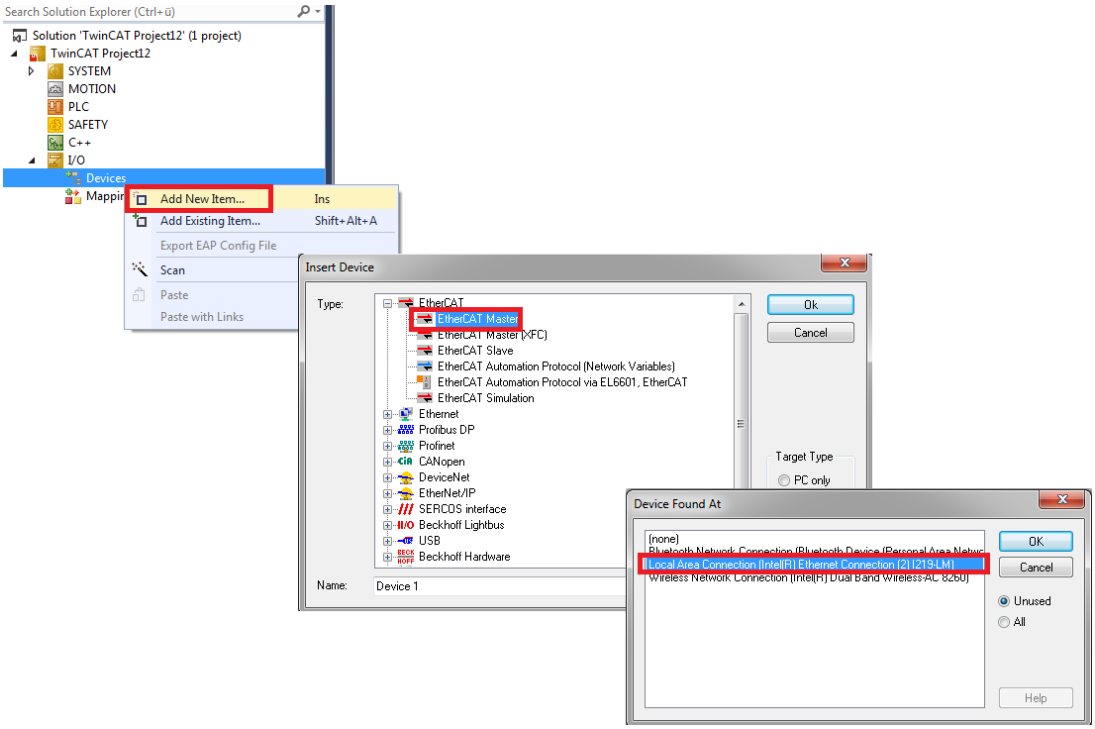

Figure 4: insert the EtherCAT Master device

The assignment of a network card to the EtherCAT master functionality can be checked, and in case changed, at any time in the “Adapter” tab of the master device (Figure 5).

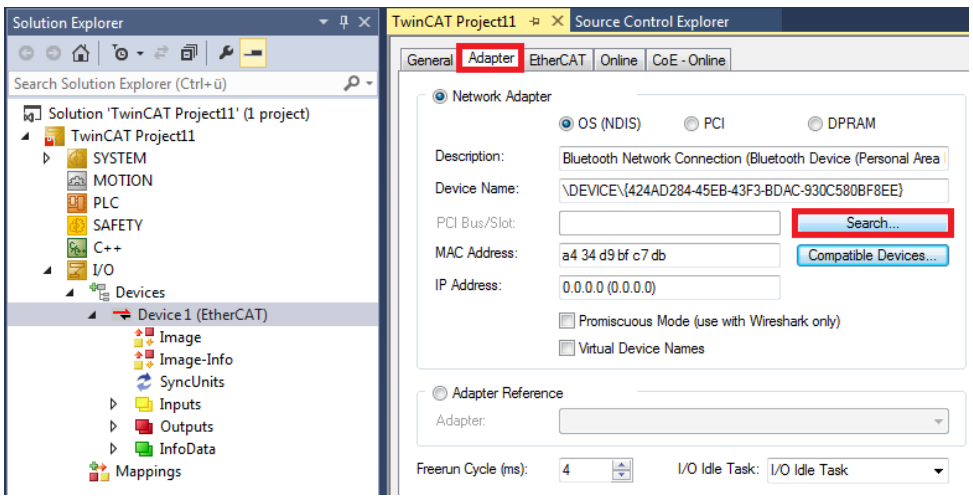

Figure 5: check and change assignment of EtherCAT master to a network card

b. Click with the right mouse button on the master device (or on an already configured slave), and add a new slave device (Figure 6).

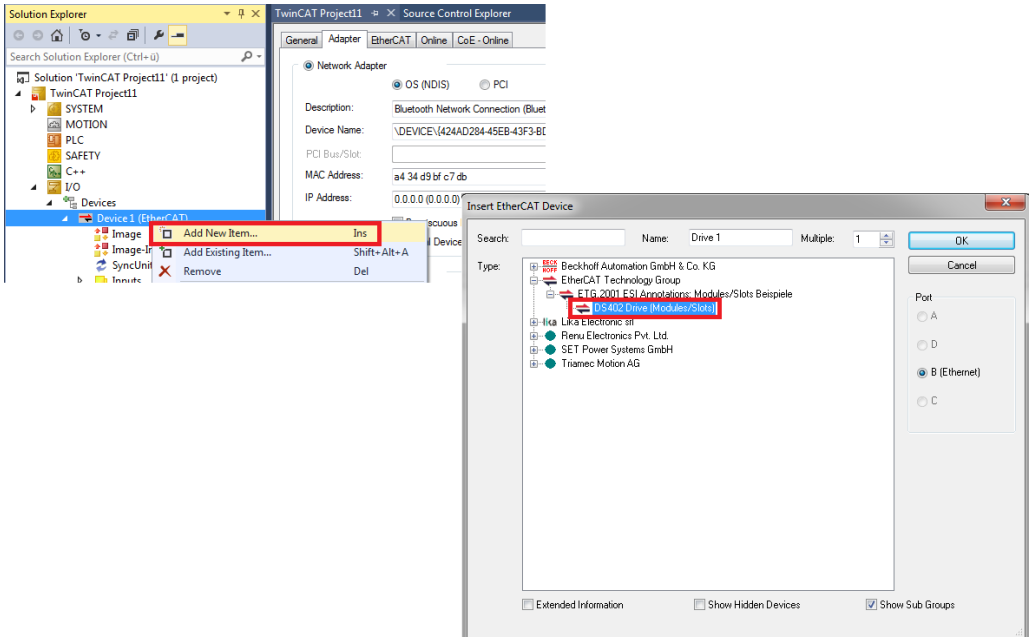

Figure 6: insert an EtherCAT slave device

Note. In order to be recognized by the TwinCAT development environment, ESI descriptions of slave devices shall be saved in the default directory C:/TwinCAT/3.1/Config/Io/EtherCAT/.

If the content of the TwinCAT default folder is changed (new files are added, old files are deleted, files overwritten, content of one or more files is changed), the ESI database shall be reloaded in order to make the changes available (Figure 7).

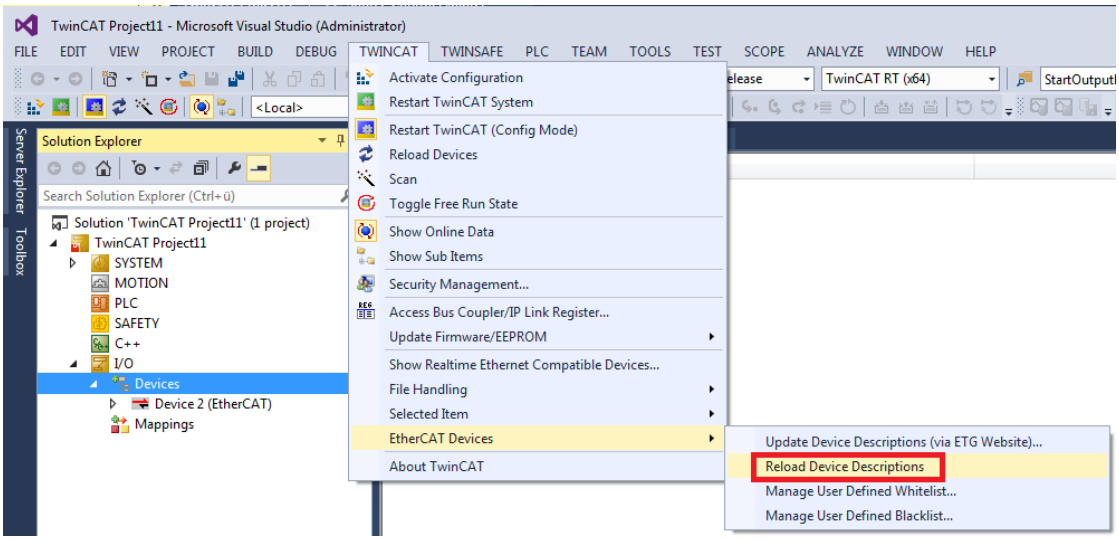

Figure 7: reload ESI database

Note. In case ESI descriptions corresponding to different Revision Numbers of a device are available, by enabling “Extended Information” and “Show Hidden Devices” it is possible to select a specific device revision (Figure 8). Otherwise, the device with the highest Revision Number will be added per default.

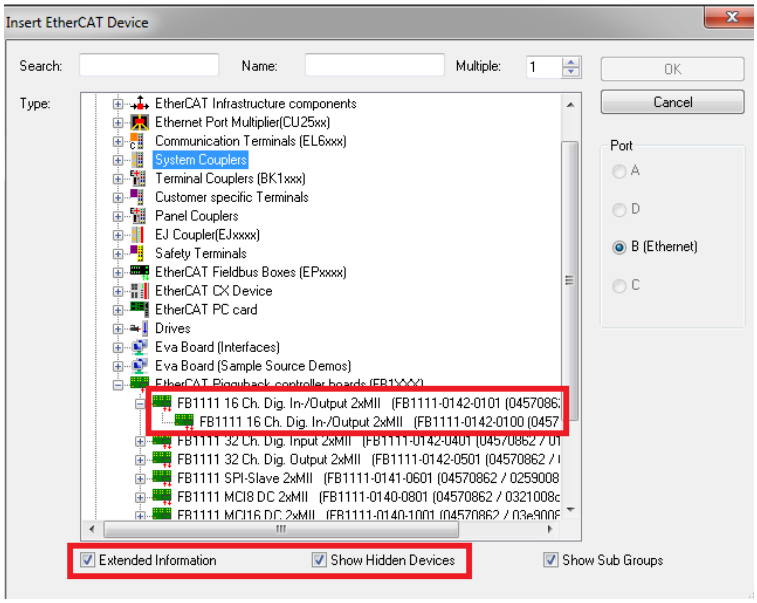

Figure 8: show different Revision Numbers of a device

Note. For devices supporting more than two EtherCAT ports, it is possible to select on which port the next device shall be appended (Figure 9).

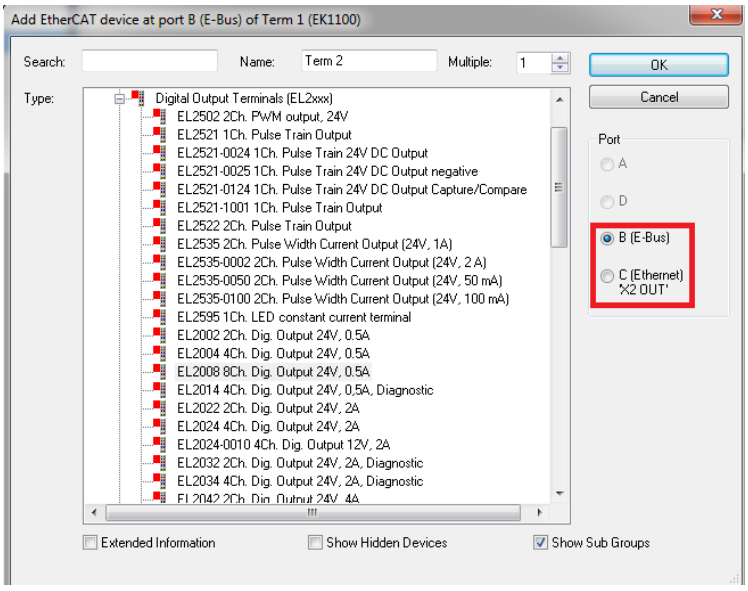

Figure 9: selection of output port

2.2 Automatic configuration

a. Click with the right mouse button on I/O → Devices, select “Scan” and confirm the hint which automatically shows-up: the device called “EtherCAT”, corresponding to the master device functionality, should be automatically selected(Figure 10). Then press “OK”.

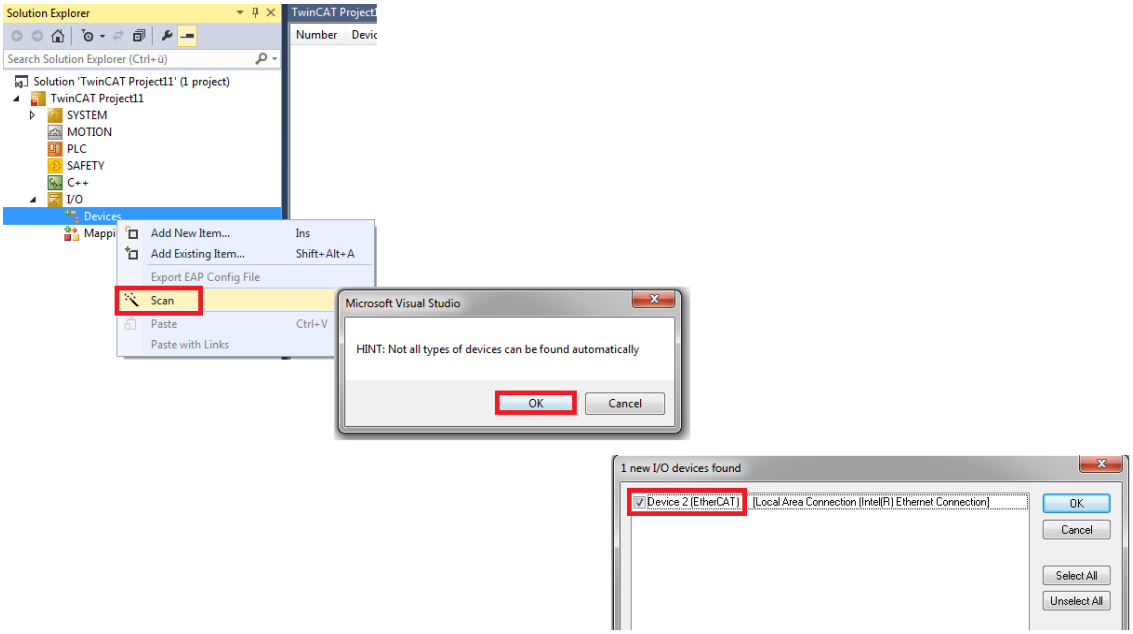

Figure 10: automatically scan devices

Note. If “No new I/O devices found” is shown when starting a scan, the TwinCAT Real-Time Ethernet Driver is not properly installed (Figure 11).

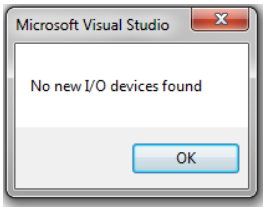

Figure 11: Real-Time driver not properly installed

In this case, it is suggested to check the driver installation first, as described above.

If the network card is shown as “EtherCAT Automation Protocol” (instead as “EtherCAT”), this means that the TwinCAT Real-Time Ethernet Driver is properly detected, but no hardware connectivity with EtherCAT slaves is detected (Figure 12).

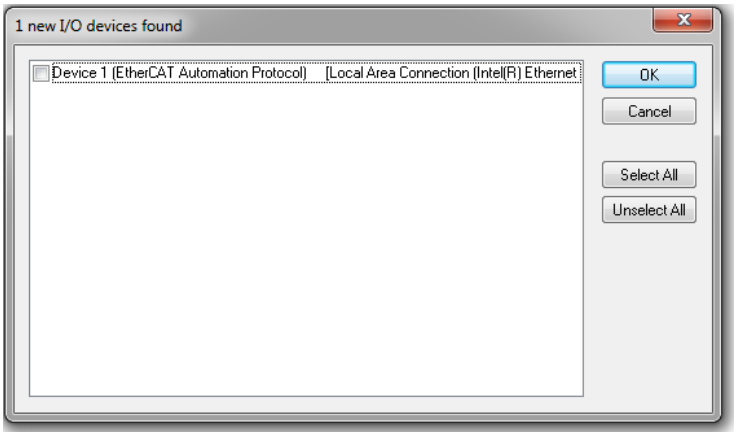

Figure 12: hardware connection with slave devices missing

In this case, it is suggested to check the hardware connection between the EtherCAT devices,

b. Scan the slave devices (Figure 13).

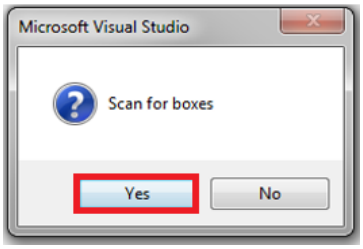

Figure 13: scan slave devices

Note. Pop-ups like in Figure 14 shown during the network scan mean that for one or more detected slaves (identified by Vendor ID, Product Code and Revision Number) no corresponding ESI description was found in the default TwinCAT directory.

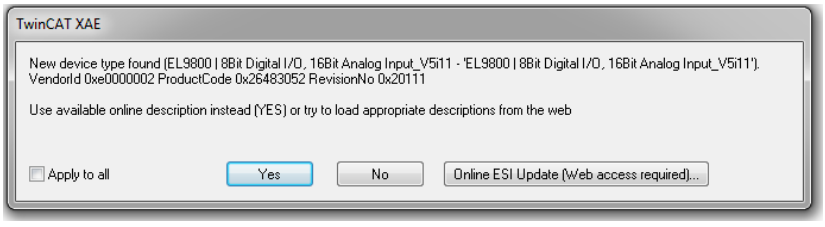

Figure 14: ESI description corresponding to slave identity missing

When answering “Yes” to the previous pop-up, the TwinCAT configurator will load the (default) device description from the EEPROM SII instead of loading it from the ESI file. Nevertheless, it is always suggested to provide the correct ESI file for each device to the Configuration Tool.

3 Testing an EtherCAT Configuration

The TwinCAT 3.1 development environment also allows to simply test the communication on the EtherCAT network.

Here are reported a couple of general remarks which are independent from the specific operating mode:

a. All cycle times in TwinCAT are always expressed as multiple integers of the so-called “Base Time”. The default value of the Base Time is 1 ms, but it can be changed in case the EtherCAT communication needs to be tested with cycle times which are not a multiple integers of 1 ms (Figure 15).

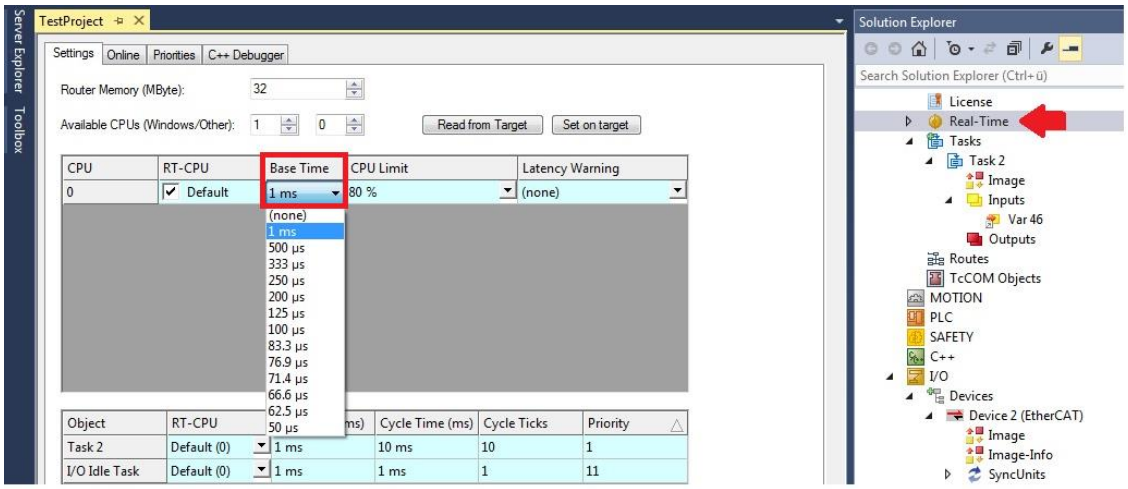

Figure 15: change TwinCAT Base Time

b. Once the communication has been started, the state of the network as well as of single slave devices can be monitored in the “Online” tab of the master device (Figure **16**).

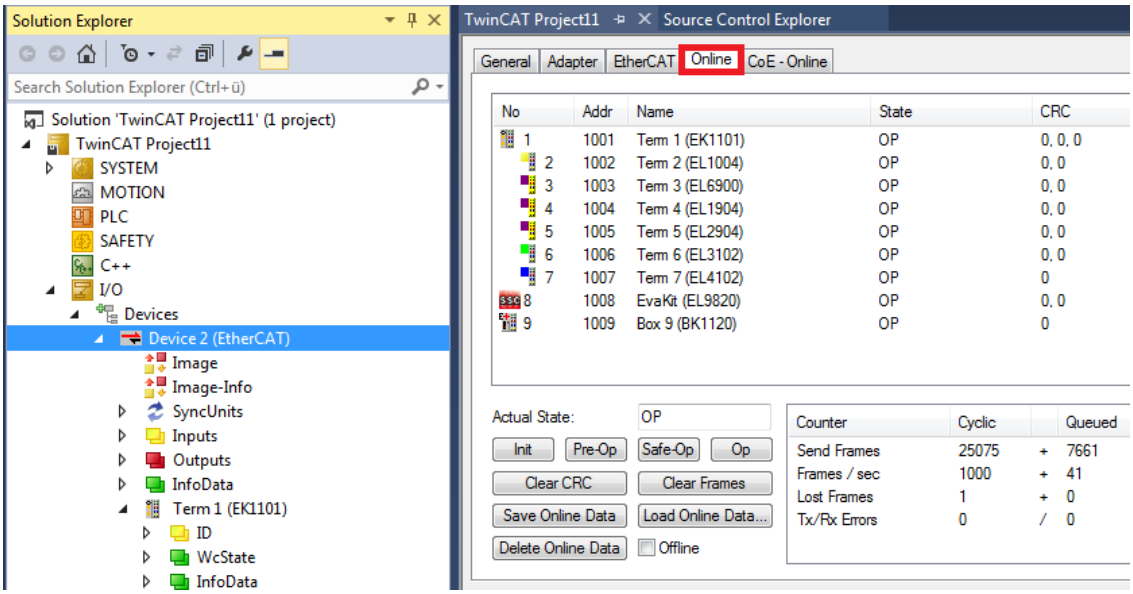

Figure 16: monitor online state of the EtherCAT network

3.1 Free Run Mode

Free Run is a special state of Config Mode, in which the TwinCAT development environment triggers in user mode a cyclic communication with the EtherCAT network. Free Run is sufficient to test most of the EtherCAT communication features, but if the network needs to work with the Distributed Clocks (see DC (Distributed Clocks)): in this case, slaves working in DC-Synchronous mode could be prevented from reaching the Operational state due to communication jitter (testing in Run Mode is suggested for DC slaves, as described in Chapter 3.2).

a. Free Run operation can be started and stopped by means of the toggle command (Figure 17). When Free Run is active, the TwinCAT blue Config Mode icon will blink in red.

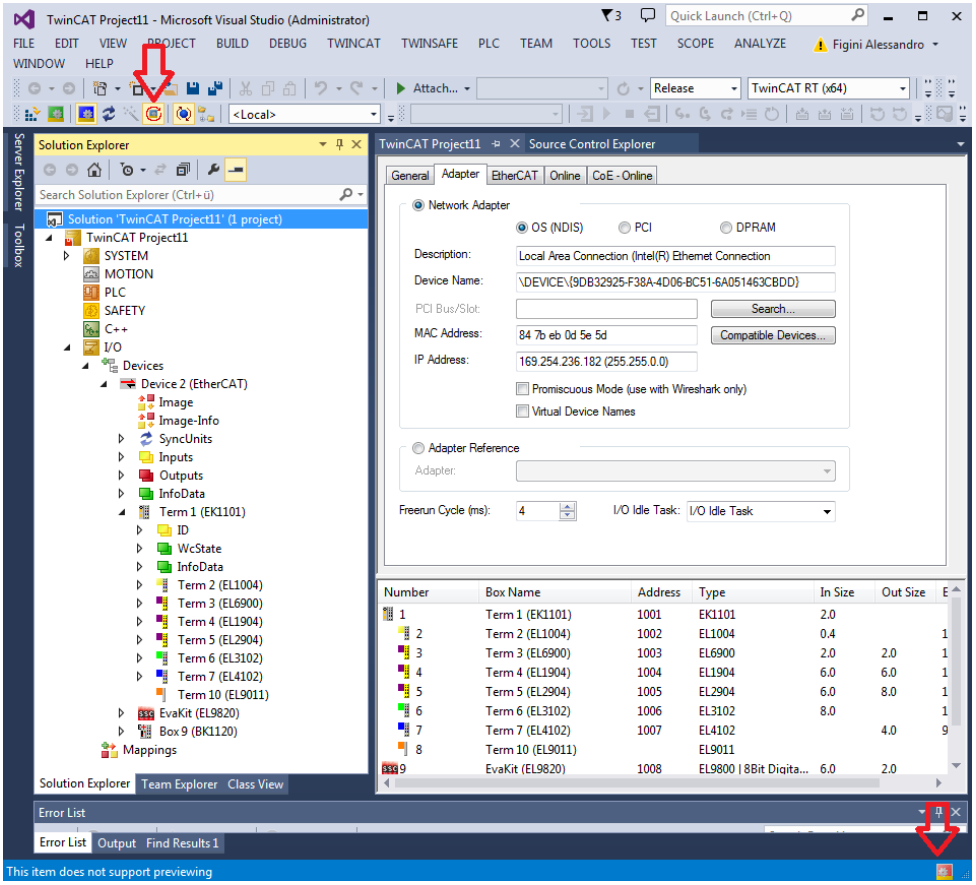

Figure 17: start/stop Free Run

b. If any configuration parameter is changed, and whenever the communication needs to be restarted/refreshed, command “Reload Devices” shall be given in Free Run mode (Figure 18).

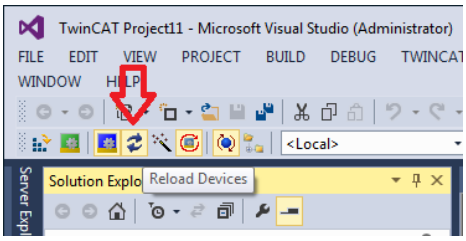

Figure 18: Reload Devices to refresh/restart communication in Free Run mode

c. The default cyclic time in Free Run mode is 4 ms. This setting can be changed in the “Freerun Cycle (ms)” text box of the master Adapter tab (Figure 19). The Free Run cycle time shall be always a multiple integer of the TwinCAT Base Time.

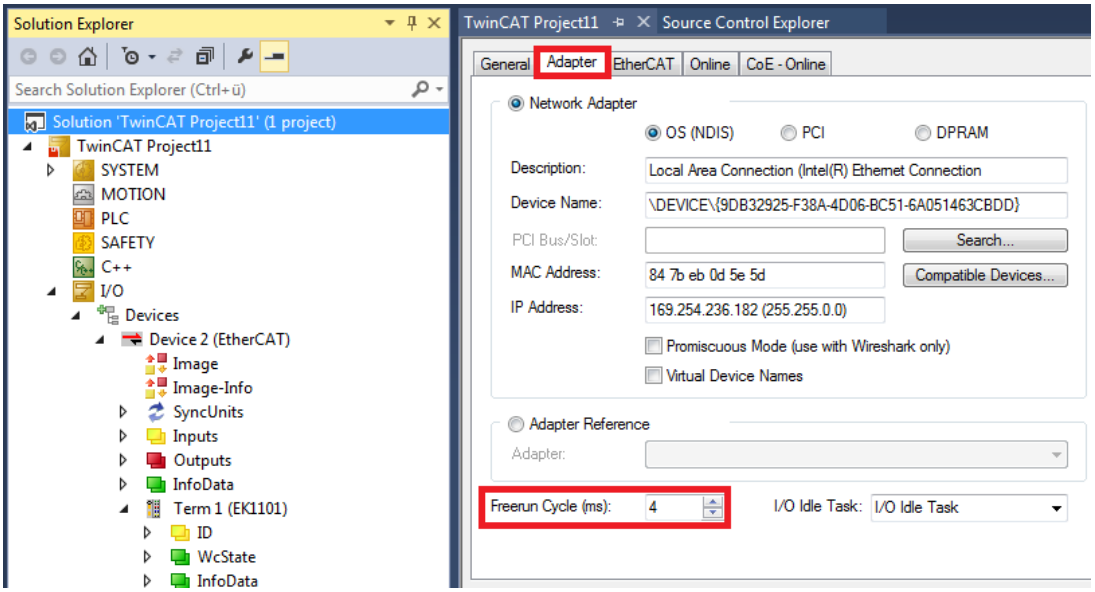

Figure 19: configure cycle time in Free Run mode

3.2 Run Mode

When one or more slave devices need to work in DC-Synchronous mode, and in any case when the communication wants to be tested under hard real-time conditions, Free Run could not be suitable due to higher jitter values in user mode: in this case, it is recommended to operate the network in Run Mode (the EtherCAT communication will take place in kernel mode).

a. Create a new real-time I/O Task (Figure 20).

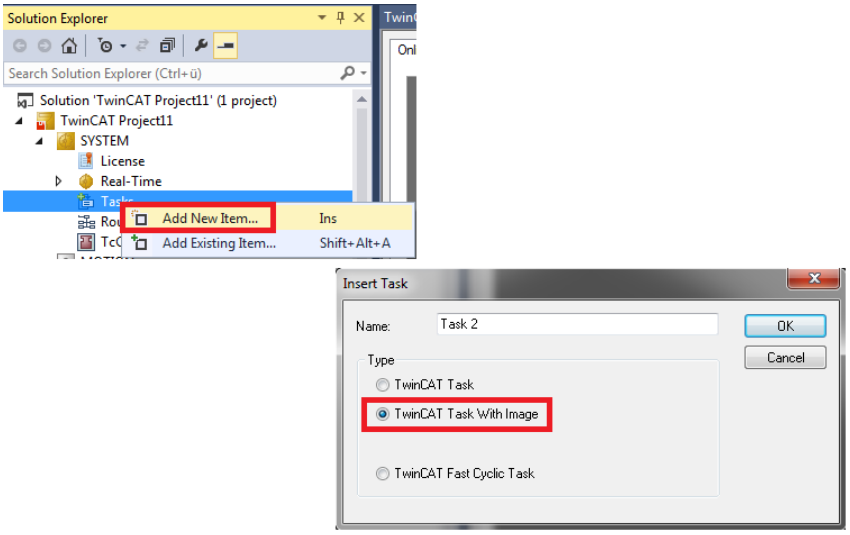

Figure 20: create an I/O task

b. Manually define a cyclic Input or Output variable for the I/O Task with a datatype matching the datatype of a process data variable in the EtherCAT network: the datatype of the EtherCAT variable can be directly checked in the project (Figure 21). The name assigned to the created variable is not relevant.

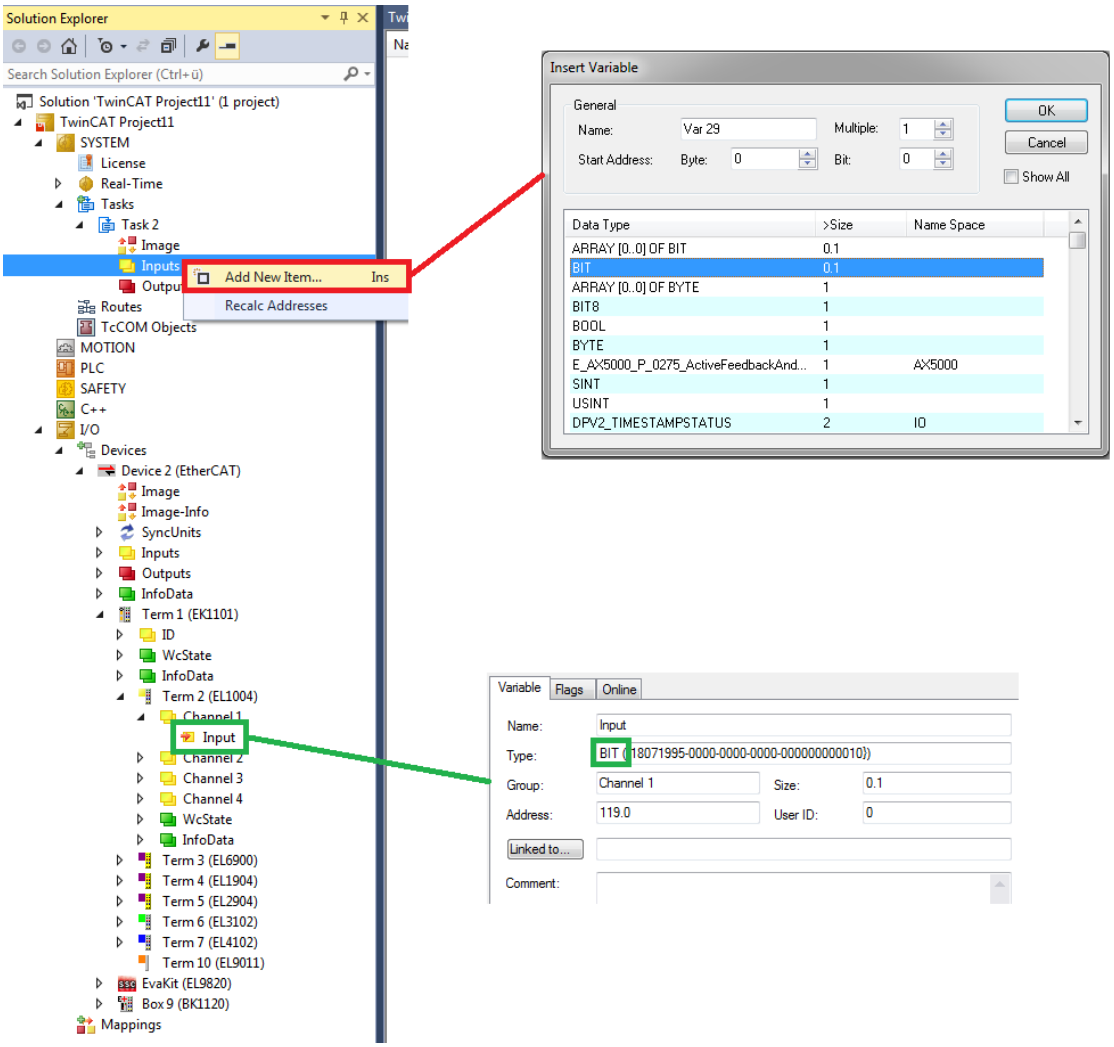

Figure 21: creating a cyclic variable in the I/O Task

c. Create a link between the software variable in the I/O Task and the hardware variable in the EtherCAT process data (Figure 22).

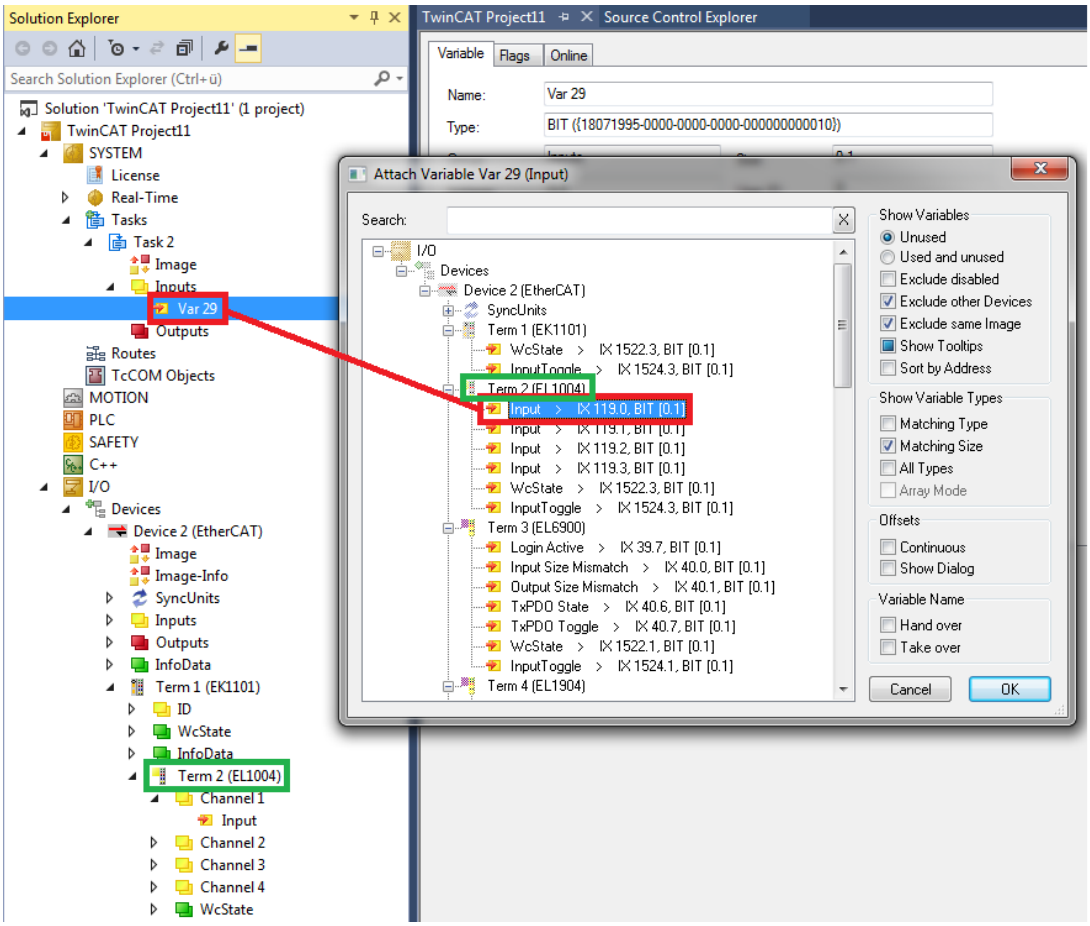

Figure 22: create a link between I/O Task and EtherCAT network

d. Activate configuration and start TwinCAT in Run Mode (Figure **23**). The command “Activate Configuration” shall be applied every time one or more parameters are changed in the configuration and the changes need to be applied (similar from this point of view to “Reload Devices” command in Free Run mode).

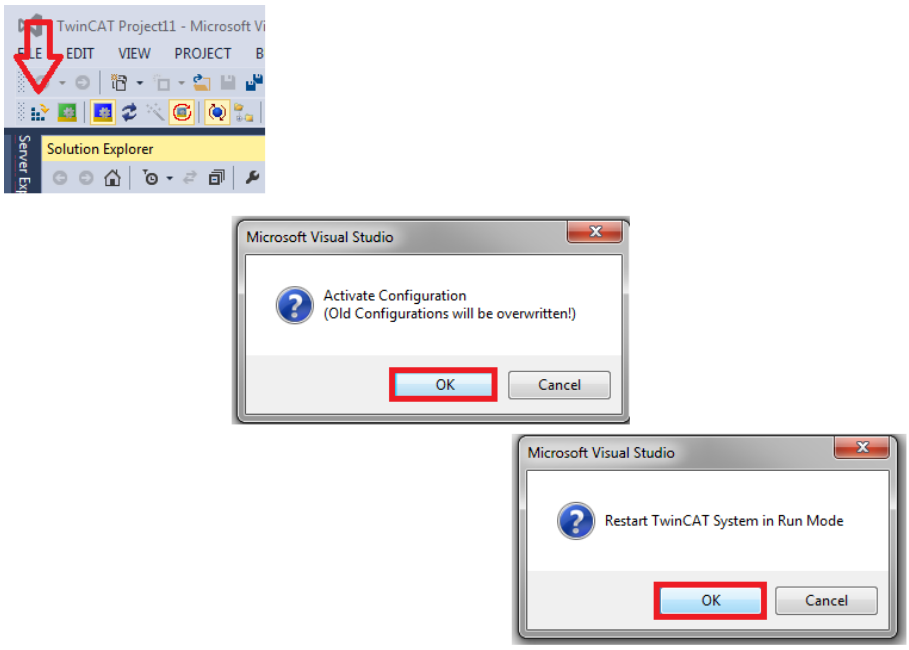

Figure 23: activate configuration and start TwinCAT in Run Mode

e. The cycle time of the EtherCAT communication can be changed by changing the cycle time of the linked I/O Task and reactivating the configuration (Figure 24).The cycle time shall be always a multiple integer of the TwinCAT Base Time.

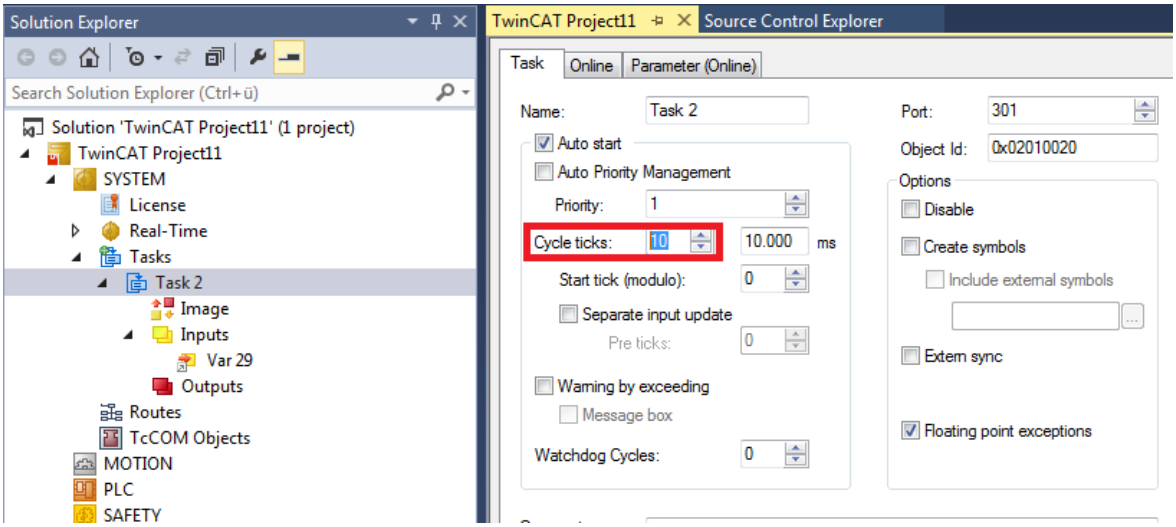

Figure 24: configure cycle time in Run Mode

Note. TwinCAT 3.1 provides a 7-day trial period for Run Mode, which can be extended for an arbitrary number of times. In order to extend the trial license for other 7-days, it will be sufficient to copy the 5-charater code which will show-up when trying to activate the configuration (Figure 25).

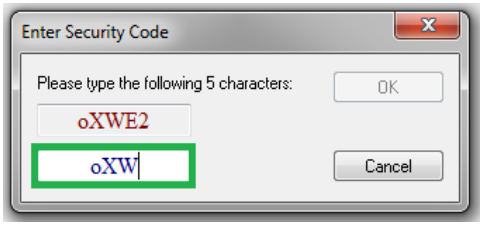

Figure 25: extend 7-days trial license

Note. In DC-Synchronous mode, the master jitter performances are fundamental to guarantee a proper synchronization at a certain cycle time (see also How-To: Configure DC SYNC Shift Times). Under "Advanced Settings…" of the EtherCAT Master it is possible to monitor the worst-case communication jitter (Figure 26): this performance depends mainly on the hardware properties of the master platform chosen. The minimum configurable communication cycle time in Run Mode should not be smaller than 3÷4 times the maximum measured jitter value.

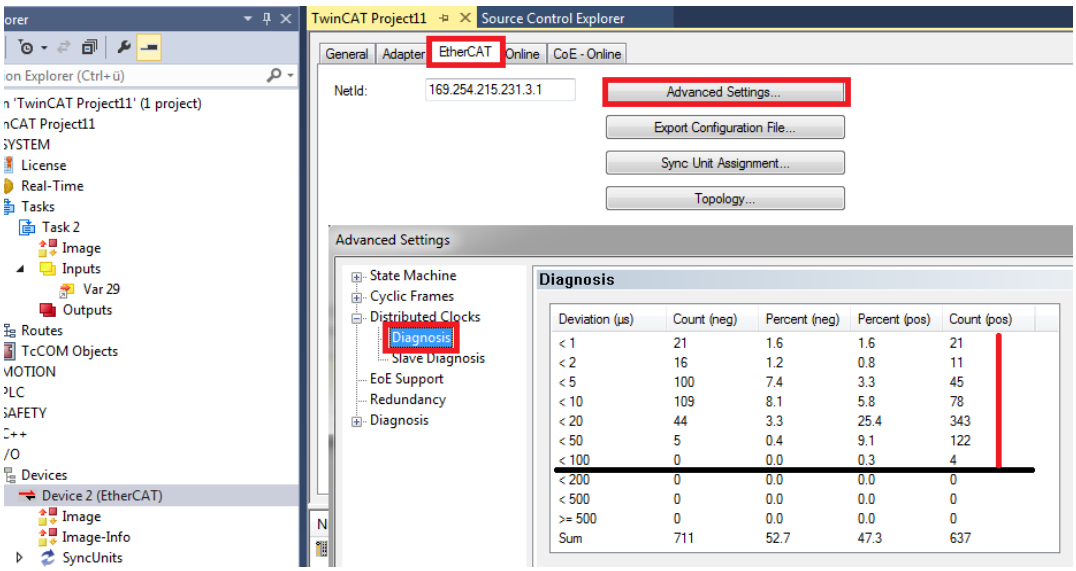

Figure 26: estimation of worst-case master jitter performances

4 References

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ETG Standards

[1] ETG.1000.2: Physical Layer service definition and protocol specification

[2] ETG.1000.3: Data Link Layer service definition

[3] ETG.1000.4: Data Link Layer protocol specification

[4] ETG.1000.5: Application Layer service definition

[5] ETG.1000.6: Application Layer protocol specification

Other References

[6] IEC 61158-x-12 (all parts for type 12): Industrial communication networks – Fieldbus specifications

[7] IEC 61784-2: Industrial communication networks – Profiles – Part 2: Additional fieldbus profiles for real-time networks based on ISO/IEC 8802-3

[8] IEC 61800-7: Adjustable speed electrical power drives systems – Part 7: Generic interface and use of profiles for power drive systems