



Twincat Library: Defining cyclic telegrams

Application Note AN105

Version	Date	Editor	Comment
000	2017-11-21	mvx	Cyclic data as available with firmware 3.2.0

Document AN104_EtherCAT-CyclicTelegrams_EP
Version 000
Source Q:\doc\ApplicationNotes\
Destination T:\doc\ApplicationNotes
Owner mvx

Copyright © 2017
Triamec Motion AG
All rights reserved.

Triamec Motion AG
Industriestrasse 49
6300 Zug / Switzerland

Phone +41 41 747 4040
Email info@triamec.com
Web www.triamec.com

Disclaimer

This document is delivered subject to the following conditions and restrictions:

- This document contains proprietary information belonging to Triamec Motion AG. Such information is supplied solely for the purpose of assisting users of Triamec products.
- The text and graphics included in this manual are for the purpose of illustration and reference only. The specifications on which they are based are subject to change without notice.
- Information in this document is subject to change without notice.

Table of Contents

1 Target and Purpose.....2	3 Configuration of the cyclic data.....3
2 Find the COE address.....2	

1 Target and Purpose

This describes functions for EtherCAT devices. For Tria-Link devices, refer to AN105. The axis module of an EtherCAT device contains basic information of a drive, such as states and actual positions. Further extensions are available for two situations.

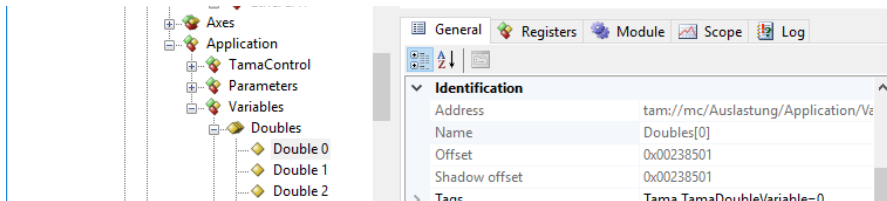
- An extended axis module contains additional information when using TwinCAT touch probes.
- This application note describes further extensions to publish flexible cyclic PDO data from the device to TwinCAT, as available since FW 3.2.0

2 Find the COE address

First we have to find the COE address of the register that should be cyclically exchanged with TwinCAT.

Triamec EtherCAT drives support two register ranges. The first set is published to TwinCAT by the standard COE-Information method. As a consequence, these are shown in the TwinCAT tab “COE-Online”. A second range is not shown directly. This large set of registers is usually accessed using the TAM System Explorer when tuning the axis-motor-drive system. Nevertheless, the registers can be accessed from TwinCAT by the same method as those published by the COE-Information method.

Open the TAM System Explorer and select the register in the tree view to your left. Use the tab “General” and find the entry “Offset”. The 16 bits [23:8] are the index and the lower 8 bits are the subIndex.



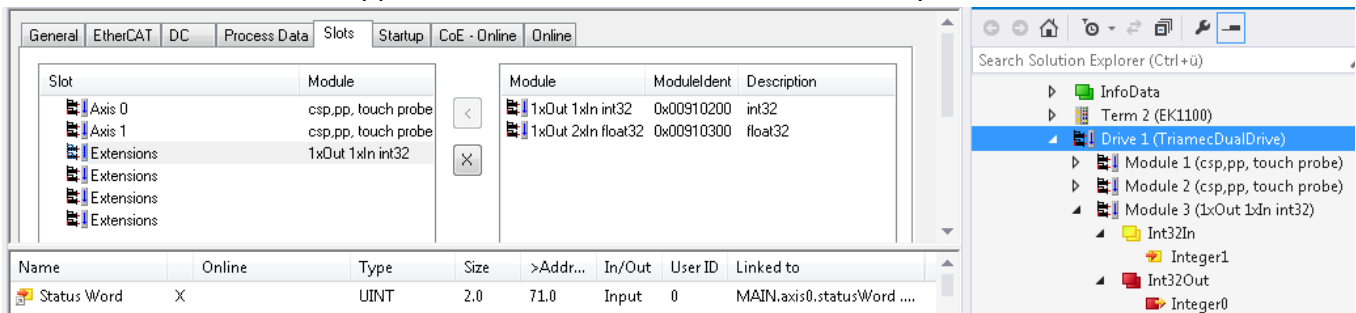
Note: Most registers are 32 Bit wide. Some registers are 64Bit wide (mainly positions). The concept of arrays is

- REAL or INT array indices {0, 1, 2, ...} use subIndices = {1, 2, 3, ...}
- LREAL array indices {0, 1, 2, ...} use subIndices = {1, 3, 5, ...}

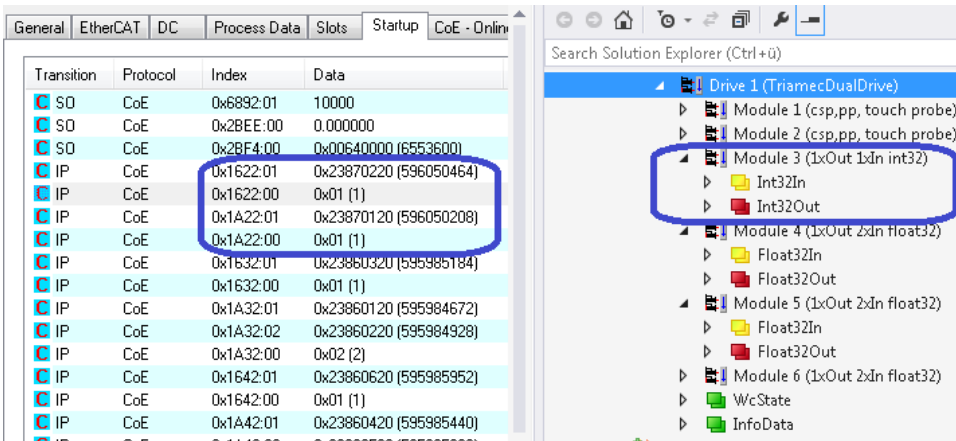
3 Configuration of the cyclic data

We will be using the slot mechanism of TwinCAT for PDO assignment. In the background, this will fill the PdoAssign registers 0x1C12 and 0x1C13. We specified 6 slots in the ESI file. Indices $k=0$ and 1 correspond to axis 0 and 1 and the four extensions correspond to indices $k=2..5$. The following example shows how to use the first extension ($k=2$). The remaining extensions may be used similarly.

- Open *Slots* (see figure below),
- Select the first extension,
- Select one choice from the right list for integer or float variables.
- Use "<" to add this choice to the extension.
- The new variables will appear as an additional module $k+1$ of the cyclic interface of the drive.



The drive registers to be published and received are chosen in the TAB "startup". Highlighted is the module $k=2$, which configures Module 3= $k+1$ in the IO-section to your right.



- 0x16k2:01 configures the register for direction TwinCAT → drive
- 0x1Ak2:01 configures the register for direction drive → TwinCAT

We chose the word 0x23870220. The upper 16bits (here 0x2387) are the COE index. Bits 15:8 (here 0x02) are the COE subIndex. The lowest 8 bits (here 0x20) are the length of the register. Currently, only 32Bit registers are supported.

After activating the project, the cyclic data will be exchanged.