



TwinCAT Library: Cross Publishing Data

Application Note 106

Version	Date	Editor	Comment
002	2014-01-16	mxv	Convert to new document style, minor corrections
003	2023-03-06	sm	Application Note and described concept is discontinued

Document AN106_TwinCAT-CrossPublishing_EP
 Version 003
 Source Q:\doc\ApplicationNotes\
 Destination T:\doc\ApplicationNotes
 Owner mxv

Copyright © 2014 Triamec Motion AG Phone +41 41 747 4040
 Triamec Motion AG Lindenstrasse 16 Email info@triamec.com
 All rights reserved. 6340 Baar / Switzerland 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.....1 | 2 PLC Code.....2

Note This *Application Note* is deprecated. Please refer to *AN 142* for drive-to-drive data exchange.

1 Target and Purpose

The Triamec TwinCAT library comes with basic sample codes for NCI and CNC. This application note describes additional functions available in this library.

This application note describes how data may be transferred cyclically from one drive to another at 10 kHz. Since data does not reach the PC, the PC-CPU load is not affected.

2 PLC Code

The following declaration is made in "MAIN_SLOW"

```
publish          : TL_publishSlave2Slave;
```

add the following code to MAIN_SLOW

```
(* publish the actual position of an axis to an other axis *)
publish.src1     := gAxis[1].MC_axis.register.AxisCmd.PathPlanner.XnewCoupled;
publish.src2     := gAxis[1].MC_axis.register.AxisCmd.PathPlanner.XnewCoupled + 1; (* Float40 *)
publish.src3     := gAxis[1].MC_axis.register.AxisCmd.PathPlanner.VnewCoupled;
publish.src4     := gAxis[1].MC_axis.register.AxisCmd.PathPlanner.AnewCoupled;
publish.src5     := gAxis[1].MC_axis.register.AxisSig.PositionController.ControlOutputCurrentQ;
publish.dest1    := gAxis[2].MC_axis.register.AxisCmd.PathPlanner.XnewCoupled;
publish.dest2    := gAxis[2].MC_axis.register.AxisCmd.PathPlanner.XnewCoupled + 1; (* Float40 *)
publish.dest3    := gAxis[2].MC_axis.register.AxisCmd.PathPlanner.VnewCoupled;
publish.dest4    := gAxis[2].MC_axis.register.AxisCmd.PathPlanner.AnewCoupled;
publish.dest5    := gAxis[2].MC_axis.register.AxisSig.PositionController.ControlOutputCurrentQ;
publish(Trialink:= Trialink, AxisSrc:=gAxis[1], AxisDest:=gAxis[2]);
```