

Twincat Library: Debugging

Application Note

Version	Date	Editor	Comment
002	2012-1127	mvx	Copied from AN150 and converted to new document style

Document AN100_TwinCAT-Debugging_EP
 Version 002
 Source Q:\doc\ApplicationNotes\AN100_TwinCAT-Debugging\
 Destination T:\doc\ApplicationNotes
 Owner mvx

Copyright © 2012	Triamec Motion AG	Phone +41 41 747 4040
Triamec Motion AG	Industriestrasse 49	Email info@triamec.com
All rights reserved.	6300 Zug / 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
----------------------------	------------------

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. The attached code TL_Debug aids debugging by saving asynchronous packet data and cyclic data into text files.

2 PLC code

To use the debugging feature import TL_DEBUG.EXP from this directory and add the following code to Triamec.

Declaration in Global_Variables_Triamec

```
DEBUG      : TL_Debug;
```

Code in MAIN_SLOW

```
DEBUG.stop := Triamec.axes[4].error;  
DEBUG(path := "C:\Machine", Trialink:=Trialink);
```

Code in MAIN_FAST

```
DEBUG.scope.WriteTrace(traceId:=1, trace :=Trialink.Trialink.pll.TimestampPl1);  
DEBUG.scope.WriteTrace(traceId:=2, trace :=in[4].Position);  
DEBUG.scope.WriteTrace(traceId:=3, trace :=AxesPath[4].filter.path_out.V);  
DEBUG.scope.WriteTrace(traceId:=4, trace :=out[4].Position);
```

The scope is triggered with DEBUG.stop with a drive error. To trace path planner errors from CNC/PLC, use the following variable instead of Triamec.axes[4].error

`.gAxis[k].MC_axis_Coupler.TrajectoryWarning`

This gets TRUE if the external pathplanner velocity, acc., or jerk is higher than specified in the drive and stays TRUE until the axis couples in the next time (for example after homing or after acknowledging the error.)

Finally, use readTrace.m to view the result in matlab.

Make sure **path** shows to a valid directory